radware
availability | performance | security

Smart Network. Smart Business

Alteon Application Switch Operating System Application Guide

Software Version 29.0.0.0

Document ID: RDWR-ALOS-V2900_AG1302

February, 2013

Important Notices

The following important notices are presented in English, French, and German.

Important Notices

Notice importante

Copyright Notices

The following copyright notices are presented in English, French, and German.

Copyright Notices

Notice traitant du copyright

Copyrightvermerke

3. Weder der Name der Universität noch die Namen der Beitragenden dürfen ohne ausdrückliche vorherige schriftliche Genehmigung verwendet werden, um von dieser Software abgeleitete Produkte zu empfehlen oder zu bewerben.

Dieses Produkt enthält von Markus Friedl entwickelte Software Dieses Produkt enthält von Theo de Raadt entwickelte Software Dieses Produkt enthält von Niels Provos entwickelte Software Dieses Produkt enthält von Dug Song entwickelte Software

Dieses Produkt enthält von Aaron Campbell entwickelte Software Dieses Produkt enthält von Damien Miller entwickelte Software Dieses Produkt enthält von Kevin Steves entwickelte Software Dieses Produkt enthält von Daniel Kouril entwickelte Software Dieses Produkt enthält von Wesley Griffin entwickelte Software Dieses Produkt enthält von Per Allansson entwickelte Software Dieses Produkt enthält von Nils Nordman entwickelte Software

Dieses Produkt enthält von Simon Wilkinson entwickelte Software

Die Verbreitung und Verwendung in Quell- und binärem Format, mit oder ohne Veränderungen, sind unter folgenden Bedingungen erlaubt:

1. Die Verbreitung von Quellcodes muss den voranstehenden Copyrightvermerk, diese Liste von Bedingungen und den folgenden Haftungsausschluss beibehalten.
2. Die Verbreitung in binärem Format muss den voranstehenden Copyrightvermerk, diese Liste von Bedingungen und den folgenden Haftungsausschluss in der Dokumentation und/oder andere Materialien, die mit verteilt werden, reproduzieren.

SÄMTLICHE VORGENANNTE SOFTWARE WIRD VOM AUTOR IM IST-ZUSTAND (“AS IS”) BEREITGESTELLT. JEGLICHE AUSDRÜCKLICHEN ODER IMPLIZITEN GARANTIEN, EINSCHLIESSLICH, DOCH NICHT BESCHRÄNKT AUF DIE IMPLIZIERTEN GARANTIEN DER MARKTGÄNGIGKEIT UND DER ANWENDBARKEIT FÜR EINEN BESTIMMTEN ZWECK, SIND AUSGESCHLOSSEN.

UNTER KEINEN UMSTÄNDEN HAFTET DER AUTOR FÜR DIREKTE ODER INDIREKTE SCHÄDEN, FÜR BEI VERTRAGSERFÜLLUNG ENTSTANDENE SCHÄDEN, FÜR BESONDERE SCHÄDEN, FÜR SCHADENSERSATZ MIT STRAFCHARAKTER, ODER FÜR FOLGESCHÄDEN EINSCHLIESSLICH, DOCH NICHT BESCHRÄNKT AUF, ERWERB VON ERSATZGÜTERN ODER ERSATZLEISTUNGEN; VERLUST AN NUTZUNG, DATEN ODER GEWINN; ODER GESCHÄFTSUNTERBRECHUNGEN) GLEICH, WIE SIE ENTSTANDEN SIND, UND FÜR JEGLICHE ART VON HAFTUNG, SEI ES VERTRÄGE, GEFÄHRDUNGSHAFTUNG, ODER DELIKTISCHE HAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER ANDERE), DIE IN JEGLICHER FORM FOLGE DER BENUTZUNG DIESER SOFTWARE IST, SELBST WENN AUF DIE MÖGLICHKEIT EINES SOLCHEN SCHADENS HINGEWIESEN WURDE.

Safety Instructions

The following safety instructions are presented in English, French, and German.

Safety Instructions

CAUTION

A readily accessible disconnect device shall be incorporated in the building installation wiring.

Due to the risks of electrical shock, and energy, mechanical, and fire hazards, any procedures that involve opening panels or changing components must be performed by qualified service personnel only.

To reduce the risk of fire and electrical shock, disconnect the device from the power line before removing cover or panels.

The following figure shows the caution label that is attached to Radware platforms with dual power supplies.

Figure 1: Electrical Shock Hazard Label

CAUTION	ATTENTION
This unit has more than one power supply. Disconnect all power supplies before maintenance to avoid electric shock.	Cette unité a plus d'une source d'alimentation électrique. Débranchez toutes les sources d'alimentations électriques avant toute maintenance pour éviter les chocs électriques.

DUAL-POWER-SUPPLY-SYSTEM SAFETY WARNING IN CHINESE

The following figure is the warning for Radware platforms with dual power supplies.

Figure 2: Dual-Power-Supply-System Safety Warning in Chinese

本设备有两个电源供电，未避免电击危险，操作时需要加倍小心。
只有当这两个电源完全断开时才可以安全操作

Translation of Dual-Power-Supply-System Safety Warning in Chinese:

This unit has more than one power supply. Disconnect all power supplies before maintenance to avoid electric shock.

SERVICING

Do not perform any servicing other than that contained in the operating instructions unless you are qualified to do so. There are no serviceable parts inside the unit.

HIGH VOLTAGE

Any adjustment, maintenance, and repair of the opened instrument under voltage must be avoided as much as possible and, when inevitable, must be carried out only by a skilled person who is aware of the hazard involved.
Capacitors inside the instrument may still be charged even if the instrument has been disconnected from its source of supply.

GROUNDING

Before connecting this device to the power line, the protective earth terminal screws of this device must be connected to the protective earth in the building installation.

LASER

This equipment is a Class 1 Laser Product in accordance with IEC60825 - 1: 1993 + A1:1997 + A2:2001 Standard.

FUSES

Make sure that only fuses with the required rated current and of the specified type are used for replacement. The use of repaired fuses and the short-circuiting of fuse holders must be avoided. Whenever it is likely that the protection offered by fuses has been impaired, the instrument must be made inoperative and be secured against any unintended operation.

LINE VOLTAGE

Before connecting this instrument to the power line, make sure the voltage of the power source matches the requirements of the instrument. Refer to the Specifications for information about the correct power rating for the device.

48V DC-powered platforms have an input tolerance of 36-72V DC.

SPECIFICATION CHANGES

Specifications are subject to change without notice.

Note: This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15B of the FCC Rules and EN55022 Class A, EN 55024; EN 61000-3-2; EN 61000-3-3; IEC 61000 4-2 to 4-6, IEC 61000 4-8 and IEC 61000-4-11For CE MARK Compliance. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user is required to correct the interference at his own expense.

VCCI ELECTROMAGNETIC-INTERFERENCE STATEMENTS

Figure 3: Statement for Class A VCCI-certified Equipment

この装置は，クラスA情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。 VCCI－A

Translation of Statement for Class A VCCI-certified Equipment:

This is a Class A product based on the standard of the Voluntary Control Council for Interference by Information Technology Equipment (VCCI). If this equipment is used in a domestic environment, radio disturbance may occur, in which case, the user may be required to take corrective action.

Figure 4: Statement for Class B VCCI-certified Equipment

この装置は，クラスB情報技術装置です。この装置は，家庭環境で使用することを目的としていますが，この装置がラジオやテレビジョン受信機に近接して使用されると，受信障害を引き起こすことがあります。
取扱説明書に従って正しい取り扱いをして下さい。 VCCI－B

Translation of Statement for Class B VCCI-certified Equipment:

This is a Class B product based on the standard of the Voluntary Control Council for Interference by Information Technology Equipment (VCCI). If this is used near a radio or television receiver in a domestic environment, it may cause radio interference.
Install and use the equipment according to the instruction manual.

KCC KOREA

Figure 5: KCC—Korea Communications Commission Certificate of Broadcasting and Communication Equipment

Figure 6: Statement For Class A KCC-certified Equipment in Korean

이 기기는 업무용(A급) 전자파적합기기로서 판
매자 또는 사용자는 이 점을 주의하시기 바라
며, 가정외의 지역에서 사용하는 것을 목적으로
합니다.

Translation of Statement For Class A KCC-certified Equipment in Korean:

This equipment is Industrial (Class A) electromagnetic wave suitability equipment and seller or user should take notice of it, and this equipment is to be used in the places except for home.

SPECIAL NOTICE FOR NORTH AMERICAN USERS

For North American power connection, select a power supply cord that is UL Listed and CSA Certified 3 - conductor, [18 AWG], terminated in a molded on plug cap rated 125 V, [10 A], with a minimum length of 1.5m [six feet] but no longer than 4.5m...For European connection, select a power supply cord that is internationally harmonized and marked “<HAR>”, 3 - conductor, 0,75 mm2 minimum mm2 wire, rated 300 V, with a PVC insulated jacket. The cord must have a molded on plug cap rated 250 V, 3 A.

RESTRICT AREA ACCESS

The DC powered equipment should only be installed in a Restricted Access Area.

INSTALLATION CODES

This device must be installed according to country national electrical codes. For North America, equipment must be installed in accordance with the US National Electrical Code, Articles 110 - 16, 110 -17, and 110 -18 and the Canadian Electrical Code, Section 12.

INTERCONNECTION OF UNITS

Cables for connecting to the unit RS232 and Ethernet Interfaces must be UL certified type DP-1 or DP-2. (Note- when residing in non LPS circuit)

OVERCURRENT PROTECTION

A readily accessible listed branch-circuit over current protective device rated 15 A must be incorporated in the building wiring for each power input.

REPLACEABLE BATTERIES

If equipment is provided with a replaceable battery, and is replaced by an incorrect battery type, then an explosion may occur. This is the case for some Lithium batteries and the following is applicable:

- If the battery is placed in an **Operator Access Area**, there is a marking close to the battery or a statement in both the operating and service instructions.
- If the battery is placed elsewhere in the equipment, there is a marking close to the battery or a statement in the service instructions.

This marking or statement includes the following text warning:

CAUTION

RISK OF EXPLOSION IF BATTERY IS REPLACED BY AN INCORRECT BATTERY TYPE. DISPOSE OF USED BATTERIES ACCORDING TO THE INSTRUCTIONS.

Caution – To Reduce the Risk of Electrical Shock and Fire

1. This equipment is designed to permit connection between the earthed conductor of the DC supply circuit and the earthing conductor equipment. See Installation Instructions.
2. All servicing must be undertaken only by qualified service personnel. There are not user serviceable parts inside the unit.
3. DO NOT plug in, turn on or attempt to operate an obviously damaged unit.
4. Ensure that the chassis ventilation openings in the unit are NOT BLOCKED.
5. Replace a blown fuse ONLY with the same type and rating as is marked on the safety label adjacent to the power inlet, housing the fuse.
6. Do not operate the device in a location where the maximum ambient temperature exceeds 40°C/104°F.
7. Be sure to unplug the power supply cord from the wall socket BEFORE attempting to remove and/or check the main power fuse.
   CLASS 1 LASER PRODUCT AND REFERENCE TO THE MOST RECENT LASER STANDARDS IEC 60 825-1:1993 + A1:1997 + A2:2001 AND EN 60825-1:1994+A1:1996+ A2:2001

AC units for Denmark, Finland, Norway, Sweden (marked on product):

- Denmark - “Unit is class I - unit to be used with an AC cord set suitable with Denmark deviations. The cord includes an earthing conductor. The Unit is to be plugged into a wall socket outlet which is connected to a protective earth. Socket outlets which are not connected to earth are not to be used!”
- Finland - (Marking label and in manual) - “Laite on liitettävä suojamaadoituskoskettimilla varustettuun pistorasiaan”
- Norway (Marking label and in manual) - “Apparatet må tilkoples jordet stikkontakt”
- Unit is intended for connection to IT power systems for Norway only.
- Sweden (Marking label and in manual) - “Apparaten skall anslutas till jordat uttag.”

To connect the power connection:

1. Connect the power cable to the main socket, located on the rear panel of the device.
2. Connect the power cable to the grounded AC outlet.

CAUTION

Risk of electric shock and energy hazard. Disconnecting one power supply disconnects only one power supply module. To isolate the unit completely, disconnect all power supplies.

Instructions de sécurité

AVERTISSEMENT

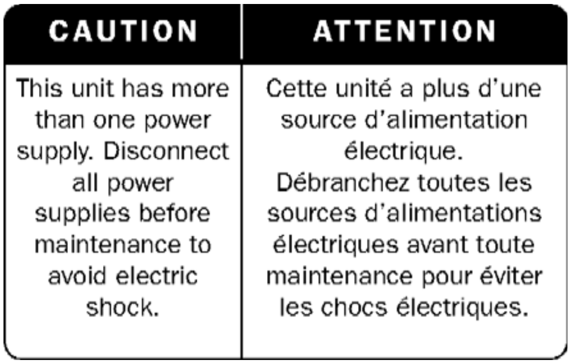
Un dispositif de déconnexion facilement accessible sera incorporé au câblage du bâtiment.

En raison des risques de chocs électriques et des dangers énergétiques, mécaniques et d’incendie, chaque procédure impliquant l’ouverture des panneaux ou le remplacement de composants sera exécutée par du personnel qualifié.

Pour réduire les risques d’incendie et de chocs électriques, déconnectez le dispositif du bloc d’alimentation avant de retirer le couvercle ou les panneaux.

La figure suivante montre l'étiquette d'avertissement apposée sur les plateformes Radware dotées de plus d'une source d'alimentation électrique.

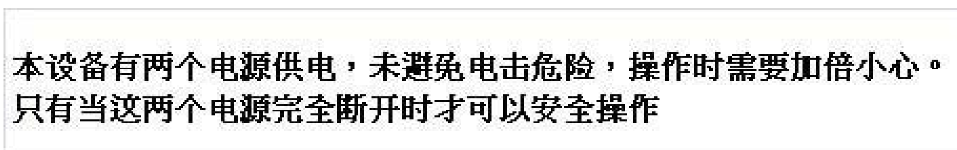
Figure 7: Étiquette d'avertissement de danger de chocs électriques



AVERTISSEMENT DE SÉCURITÉ POUR LES SYSTÈMES DOTÉS DE DEUX SOURCES D'ALIMENTATION ÉLECTRIQUE (EN CHINOIS)

La figure suivante représente l'étiquette d'avertissement pour les plateformes Radware dotées de deux sources d'alimentation électrique.

Figure 8: Avertissement de sécurité pour les systèmes dotes de deux sources d'alimentation électrique (en chinois)



Traduction de la Avertissement de sécurité pour les systèmes dotes de deux sources d'alimentation électrique (en chinois):

Cette unité est dotée de plus d'une source d'alimentation électrique. Déconnectez toutes les sources d'alimentation électrique avant d'entretenir l'appareil ceci pour éviter tout choc électrique.

ENTRETIEN

N'effectuez aucun entretien autre que ceux répertoriés dans le manuel d'instructions, à moins d'être qualifié en la matière. Aucune pièce à l'intérieur de l'unité ne peut être remplacée ou réparée.

HAUTE TENSION

Tout réglage, opération d'entretien et réparation de l'instrument ouvert sous tension doit être évité. Si cela s'avère indispensable, confiez cette opération à une personne qualifiée et consciente des dangers impliqués.

Les condensateurs au sein de l'unité risquent d'être chargés même si l'unité a été déconnectée de la source d'alimentation électrique.

MISE A LA TERRE

Avant de connecter ce dispositif à la ligne électrique, les vis de protection de la borne de terre de cette unité doivent être reliées au système de mise à la terre du bâtiment.

LASER

Cet équipement est un produit laser de classe 1, conforme à la norme IEC60825 - 1: 1993 + A1: 1997 + A2: 2001.

FUSIBLES

Assurez-vous que, seuls les fusibles à courant nominal requis et de type spécifié sont utilisés en remplacement. L'usage de fusibles réparés et le court-circuitage des porte-fusibles doivent être évités. Lorsqu'il est pratiquement certain que la protection offerte par les fusibles a été détériorée, l'instrument doit être désactivé et sécurisé contre toute opération involontaire.

TENSION DE LIGNE

Avant de connecter cet instrument à la ligne électrique, vérifiez que la tension de la source d'alimentation correspond aux exigences de l'instrument. Consultez les spécifications propres à l'alimentation nominale correcte du dispositif.

Les plateformes alimentées en 48 CC ont une tolérance d'entrée comprise entre 36 et 72 V CC.
MODIFICATIONS DES SPÉCIFICATIONS

Les spécifications sont sujettes à changement sans notice préalable.

Remarque: Cet équipement a été testé et déclaré conforme aux limites définies pour un appareil numérique de classe A, conformément au paragraphe 15B de la réglementation FCC et EN55022 Classe A, EN 55024, EN 61000-3-2; EN 61000-3-3; IEC 61000 4-2 to 4-6, IEC 61000 4-8, et IEC 61000-4-11, pour la marque de conformité de la CE. Ces limites sont fixées pour fournir une protection raisonnable contre les interférences nuisibles, lorsque l'équipement est utilisé dans un environnement commercial. Cet équipement génère, utilise et peut émettre des fréquences radio et, s'il n'est pas installé et utilisé conformément au manuel d'instructions, peut entraîner des interférences nuisibles aux communications radio. Le fonctionnement de cet équipement dans une zone résidentielle est susceptible de provoquer des interférences nuisibles, auquel cas l'utilisateur devra corriger le problème à ses propres frais.

DÉCLARATIONS SUR LES INTERFÉRENCES ÉLECTROMAGNÉTIQUES VCCI

Figure 9: Déclaration pour l'équipement de classe A certifié VCCI

この装置は，クラスA情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。 VCCI－A

Traduction de la Déclaration pour l'équipement de classe A certifié VCCI:

Il s'agit d'un produit de classe A, basé sur la norme du Voluntary Control Council for Interference by Information Technology Equipment (VCCI). Si cet équipement est utilisé dans un environnement domestique, des perturbations radioélectriques sont susceptibles d'apparaître. Si tel est le cas, l'utilisateur sera tenu de prendre des mesures correctives.

Figure 10: Déclaration pour l'équipement de classe B certifié VCCI

この装置は，クラスB情報技術装置です。この装置は，家庭環境で使用することを目的としていますが，この装置がラジオやテレビジョン受信機に近接して使用されると，受信障害を引き起こすことがあります。

取扱説明書に従って正しい取り扱いをして下さい。 VCCI－B

Traduction de la Déclaration pour l'équipement de classe B certifié VCCI:

Il s'agit d'un produit de classe B, basé sur la norme du Voluntary Control Council for Interference by Information Technology Equipment (VCCI). S'il est utilisé à proximité d'un poste de radio ou d'une télévision dans un environnement domestique, il peut entraîner des interférences radio.

Installez et utilisez l'équipement selon le manuel d'instructions.

KCC Corée

Figure 11: KCC—Certificat de la commission des communications de Corée pour les equipements de radiodiffusion et communication.

Figure 12: Déclaration pour l'équipement de classe A certifié KCC en langue coréenne

이 기기는 업무용(A급) 전자파적합기기로서 판매자 또는 사용자는 이 점을 주의하시기 바라며, 가정외의 지역에서 사용하는 것을 목적으로 합니다.

Translation de la Déclaration pour l'équipement de classe A certifié KCC en langue coréenne:

Cet équipement est un matériel (classe A) en adéquation aux ondes électromagnétiques et le vendeur ou l'utilisateur doit prendre cela en compte. Ce matériel est donc fait pour être utilisé ailleurs qu' á la maison.

NOTICE SPÉCIALE POUR LES UTILISATEURS NORD-AMÉRICAINS

Pour un raccordement électrique en Amérique du Nord, sélectionnez un cordon d'alimentation homologué UL et certifié CSA 3 - conducteur, [18 AWG], muni d'une prise moulée à son extrémité, de 125 V, [10 A], d'une longueur minimale de 1,5 m [six pieds] et maximale de 4,5m...Pour la connexion européenne, choisissez un cordon d'alimentation mondialement homologué et marqué "<HAR>", 3 - conducteur, câble de 0,75 mm2 minimum, de 300 V, avec une gaine en PVC isolée. La prise à l'extrémité du cordon, sera dotée d'un sceau moulé indiquant: 250 V, 3 A.

ZONE A ACCÈS RESTREINT

L'équipement alimenté en CC ne pourra être installé que dans une zone à accès restreint. CODES D'INSTALLATION

Ce dispositif doit être installé en conformité avec les codes électriques nationaux. En Amérique du Nord, l'équipement sera installé en conformité avec le code électrique national américain, articles 110-16, 110 -17, et 110 -18 et le code électrique canadien, Section 12. INTERCONNEXION DES UNÎTES.

Les câbles de connexion à l'unité RS232 et aux interfaces Ethernet seront certifiés UL, type DP-1 ou DP-2. (Remarque- s'ils ne résident pas dans un circuit LPS) PROTECTION CONTRE LES SURCHARGES.

Un circuit de dérivation, facilement accessible, sur le dispositif de protection du courant de 15 A doit être intégré au câblage du bâtiment pour chaque puissance consommée.

BATTERIES REMPLAÇABLES

Si l'équipement est fourni avec une batterie, et qu'elle est remplacée par un type de batterie incorrect, elle est susceptible d'exploser. C'est le cas pour certaines batteries au lithium, les éléments suivants sont donc applicables:

- Si la batterie est placée dans une zone d'accès opérateur, une marque est indiquée sur la batterie ou une remarque est insérée, aussi bien dans les instructions d'exploitation que d'entretien.
- Si la batterie est placée ailleurs dans l'équipement, une marque est indiquée sur la batterie ou une remarque est insérée dans les instructions d'entretien.

Cette marque ou remarque inclut l'avertissement textuel suivant:

AVERTISSEMENT

RISQUE D'EXPLOSION SI LA BATTERIE EST REMPLACÉE PAR UN MODÈLE INCORRECT. METTRE AU REBUT LES BATTERIES CONFORMÉMENT AUX INSTRUCTIONS.

Attention - Pour réduire les risques de chocs électriques et d'incendie

1. Cet équipement est conçu pour permettre la connexion entre le conducteur de mise à la terre du circuit électrique CC et l'équipement de mise à la terre. Voir les instructions d'installation.
2. Tout entretien sera entrepris par du personnel qualifié. Aucune pièce à l'intérieur de l'unité ne peut être remplacée ou réparée.
3. NE branchez pas, n'allumez pas ou n'essayez pas d'utiliser une unité manifestement endommagée.
4. Vérifiez que l'orifice de ventilation du châssis dans l'unité n'est PAS OBSTRUE.
5. Remplacez le fusible endommagé par un modèle similaire de même puissance, tel qu'indiqué sur l'étiquette de sécurité adjacente à l'arrivée électrique hébergeant le fusible.
6. Ne faites pas fonctionner l'appareil dans un endroit, où la température ambiante dépasse la valeur maximale autorisée. 40°C/104°F.
7. Débranchez le cordon électrique de la prise murale AVANT d'essayer de retirer et/ou de vérifier le fusible d'alimentation principal.

PRODUIT LASER DE CLASSE 1 ET RÉFÉRENCE AUX NORMES LASER LES PLUS RÉCENTES: IEC 60

825-1: 1993 + A1: 1997 + A2: 2001 ET EN 60825-1: 1994+A1: 1996+ A2: 2001

Unités à CA pour le Danemark, la Finlande, la Norvège, la Suède (indiqué sur le produit):

- Danemark - Unité de classe 1 - qui doit être utilisée avec un cordon CA compatible avec les déviations du Danemark. Le cordon inclut un conducteur de mise à la terre. L'unité sera branchée à une prise murale, mise à la terre. Les prises non-mises à la terre ne seront pas utilisées!
- Finlande (Étiquette et inscription dans le manuel) - Laite on liitettävä suojamaadoituskoskettimilla varustettuun pistorasiaan
- Norvège (Étiquette et inscription dans le manuel) - Apparatet må tilkoples jordet stikkontakt
- L'unité peut être connectée à un système électrique IT (en Norvège uniquement).
- Suède (Étiquette et inscription dans le manuel) - Apparaten skall anslutas till jordat uttag.

Pour brancher à l'alimentation électrique:

1. Branchez le câble d'alimentation à la prise principale, située sur le panneau arrière de l'unité.
2. Connectez le câble d'alimentation à la prise CA mise à la terre.

AVERTISSEMENT

Risque de choc électrique et danger énergétique. La déconnexion d'une source d'alimentation électrique ne débranche qu'un seul module électrique. Pour isoler complètement l'unité, débranchez toutes les sources d'alimentation électrique.

ATTENTION

Risque de choc et de danger électriques. Le débranchement d'une seule alimentation stabilisée ne débranche qu'un module "Alimentation Stabilisée". Pour Isoler complètement le module en cause, il faut débrancher toutes les alimentations stabilisées.

Attention: Pour Réduire Les Risques d'Électrocution et d'Incendie

1. Toutes les opérations d'entretien seront effectuées UNIQUEMENT par du personnel d'entretien qualifié. Aucun composant ne peut être entretenu ou remplacée par l'utilisateur.
2. NE PAS connecter, mettre sous tension ou essayer d'utiliser une unité visiblement défectueuse.
3. Assurez-vous que les ouvertures de ventilation du châssis NE SONT PAS OBSTRUÉES.

4. Remplacez un fusible qui a sauté SEULEMENT par un fusible du même type et de même capacité, comme indiqué sur l'étiquette de sécurité proche de l'entrée de l'alimentation qui contient le fusible.
5. NE PAS UTILISER l'équipement dans des locaux dont la température maximale dépasse 40 degrés Centigrades.
6. Assurez vous que le cordon d'alimentation a été déconnecté AVANT d'essayer de l'enlever et/ou vérifier le fusible de l'alimentation générale.

Sicherheitsanweisungen

VORSICHT

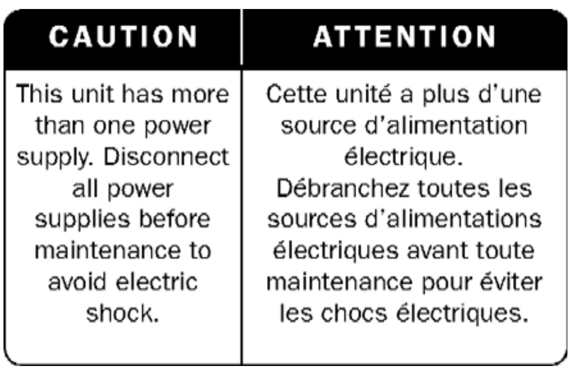
Die Elektroinstallation des Gebäudes muss ein unverzüglich zugängliches Stromunterbrechungsgerät integrieren.

Aufgrund des Stromschlagrisikos und der Energie-, mechanische und Feuergefahr dürfen Vorgänge, in deren Verlauf Abdeckungen entfernt oder Elemente ausgetauscht werden, ausschließlich von qualifiziertem Servicepersonal durchgeführt werden.

Zur Reduzierung der Feuer- und Stromschlaggefahr muss das Gerät vor der Entfernung der Abdeckung oder der Paneele von der Stromversorgung getrennt werden.

Folgende Abbildung zeigt das VORSICHT-Etikett, das auf die Radware-Plattformen mit Doppelspeisung angebracht ist.

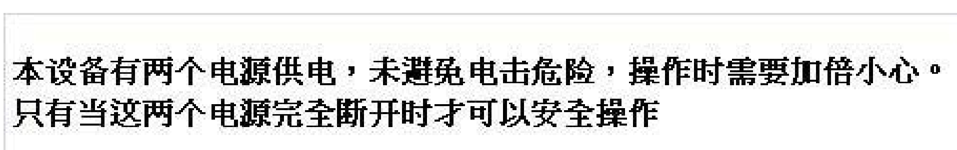
Figure 13: Warnetikett Stromschlaggefahr



SICHERHEITSHINWEIS IN CHINESISCHER SPRACHE FÜR SYSTEME MIT DOPPELSPEISUNG

Die folgende Abbildung ist die Warnung für Radware-Plattformen mit Doppelspeisung.

Figure 14: Sicherheitshinweis in chinesischer Sprache für Systeme mit Doppelspeisung



Übersetzung von Sicherheitshinweis in chinesischer Sprache für Systeme mit Doppelspeisung:

Die Einheit verfügt über mehr als eine Stromversorgungsquelle. Ziehen Sie zur Verhinderung von Stromschlag vor Wartungsarbeiten sämtliche Stromversorgungsleitungen ab.

WARTUNG

Führen Sie keinerlei Wartungsarbeiten aus, die nicht in der Betriebsanleitung angeführt sind, es sei denn, Sie sind dafür qualifiziert. Es gibt innerhalb des Gerätes keine wartungsfähigen Teile.

HOCHSPANNUNG

Jegliche Einstellungs-, Instandhaltungs- und Reparaturarbeiten am geöffneten Gerät unter Spannung müssen so weit wie möglich vermieden werden. Sind sie nicht vermeidbar, dürfen sie ausschließlich von qualifizierten Personen ausgeführt werden, die sich der Gefahr bewusst sind.

Innerhalb des Gerätes befindliche Kondensatoren können auch dann noch Ladung enthalten, wenn das Gerät von der Stromversorgung abgeschnitten wurde.

ERDUNG

Bevor das Gerät an die Stromversorgung angeschlossen wird, müssen die Schrauben der Erdungsleitung des Gerätes an die Erdung der Gebäudeverkabelung angeschlossen werden.

LASER

Dieses Gerät ist ein Laser-Produkt der Klasse 1 in Übereinstimmung mit IEC60825 - 1: 1993 + A1:1997 + A2:2001 Standard.

SICHERUNGEN

Vergewissern Sie sich, dass nur Sicherungen mit der erforderlichen Stromstärke und der angeführten Art verwendet werden. Die Verwendung reparierter Sicherungen sowie die Kurzschließung von Sicherungsfassungen muss vermieden werden. In Fällen, in denen wahrscheinlich ist, dass der von den Sicherungen gebotene Schutz beeinträchtigt ist, muss das Gerät abgeschaltet und gegen unbeabsichtigten Betrieb gesichert werden.

LEITUNGSSPANNUNG

Vor Anschluss dieses Gerätes an die Stromversorgung ist zu gewährleisten, dass die Spannung der Stromquelle den Anforderungen des Gerätes entspricht. Beachten Sie die technischen Angaben bezüglich der korrekten elektrischen Werte des Gerätes.

Plattformen mit 48 V DC verfügen über eine Eingangstoleranz von 36-72 V DC. ÄNDERUNGEN DER TECHNISCHEN ANGABEN

Änderungen der technischen Spezifikationen bleiben vorbehalten.

Hinweis: Dieses Gerät wurde geprüft und entspricht den Beschränkungen von digitalen Geräten der Klasse 1 gemäß Teil 15B FCC-Vorschriften und EN55022 Klasse A, EN55024; EN 61000-3-2; EN; IEC 61000 4-2 to 4-6, IEC 61000 4-8 und IEC 61000-4- 11 für Konformität mit der CE-Bezeichnung. Diese Beschränkungen dienen dem angemessenen Schutz vor schädlichen Interferenzen bei Betrieb des Gerätes in kommerziellem Umfeld. Dieses Gerät erzeugt, verwendet und strahlt elektromagnetische Hochfrequenzstrahlung aus. Wird es nicht entsprechend den Anweisungen im Handbuch montiert und benutzt, könnte es mit dem Funkverkehr interferieren und ihn beeinträchtigen. Der Betrieb dieses Gerätes in Wohnbereichen wird höchstwahrscheinlich zu schädlichen Interferenzen führen. In einem solchen Fall wäre der Benutzer verpflichtet, diese Interferenzen auf eigene Kosten zu korrigieren.

ERKLÄRUNG DER VCCI ZU ELEKTROMAGNETISCHER INTERFERENZ

Figure 15: Erklärung zu VCCI-zertifizierten Geräten der Klasse A

この装置は，クラスA情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。　VCCI－A

Übersetzung von Erklärung zu VCCI-zertifizierten Geräten der Klasse A:

Dies ist ein Produkt der Klasse A gemäß den Normen des Voluntary Control Council for Interference by Information Technology Equipment (VCCI). Wird dieses Gerät in einem Wohnbereich benutzt, können elektromagnetische Störungen auftreten. In einem solchen Fall wäre der Benutzer verpflichtet, korrigierend einzugreifen.

Figure 16: Erklärung zu VCCI-zertifizierten Geräten der Klasse B

この装置は，クラスB情報技術装置です。この装置は，家庭環境で使用することを目的としていますが，この装置がラジオやテレビジョン受信機に近接して使用されると，受信障害を引き起こすことがあります。
取扱説明書に従って正しい取り扱いをして下さい。 VCCI－B

Übersetzung von Erklärung zu VCCI-zertifizierten Geräten der Klasse B:

Dies ist ein Produkt der Klasse B gemäß den Normen des Voluntary Control Council for Interference by Information Technology Equipment (VCCI). Wird dieses Gerät in einem Wohnbereich benutzt, können elektromagnetische Störungen auftreten.

Montieren und benutzen Sie das Gerät laut Anweisungen im Benutzerhandbuch.

KCC KOREA

Figure 17: KCC—Korea Communications Commission Zertifikat für Rundfunk-und Nachrichtentechnik

Figure 18: Erklärung zu KCC-zertifizierten Geräten der Klasse A

이 기기는 업무용(A급) 전자파적합기기로서 판매자 또는 사용자는 이 점을 주의하시기 바라며, 가정외의 지역에서 사용하는 것을 목적으로 합니다.

Übersetzung von Erklärung zu KCC-zertifizierten Geräten der Klasse A:

Verkäufer oder Nutzer sollten davon Kenntnis nehmen, daß dieses Gerät der Klasse A für industriell elektromagnetische Wellen geeignete Geräten angehört und dass diese Geräte nicht für den heimischen Gebrauch bestimmt sind.

BESONDERER HINWEIS FÜR BENUTZER IN NORDAMERIKA

Wählen Sie für den Netzstromanschluss in Nordamerika ein Stromkabel, das in der UL aufgeführt und CSA-zertifiziert ist 3 Leiter, [18 AWG], endend in einem gegossenen Stecker, für 125 V, [10 A], mit einer Mindestlänge von 1,5 m [sechs Fuß], doch nicht länger als 4,5 m. Für europäische Anschlüsse verwenden Sie ein international harmonisiertes, mit “<HAR>” markiertes Stromkabel, mit 3 Leitern von mindestens 0,75 mm2, für 300 V, mit PVC-Umkleidung. Das Kabel muss in einem gegossenen Stecker für 250 V, 3 A enden.

BEREICH MIT EINGESCHRÄNKTEM ZUGANG

Das mit Gleichstrom betriebene Gerät darf nur in einem Bereich mit eingeschränktem Zugang montiert werden.

INSTALLATIONSCODES

Dieses Gerät muss gemäß der landesspezifischen elektrischen Codes montiert werden. In Nordamerika müssen Geräte entsprechend dem US National Electrical Code, Artikel 110 - 16, 110 - 17 und 110 - 18, sowie dem Canadian Electrical Code, Abschnitt 12, montiert werden.

VERKOPPLUNG VON GERÄTEN Kabel für die Verbindung des Gerätes mit RS232- und Ethernet- müssen UL-zertifiziert und vom Typ DP-1 oder DP-2 sein. (Anmerkung: bei Aufenthalt in einem nicht-LPS-Stromkreis)

ÜBERSTROMSCHUTZ

Ein gut zugänglicher aufgeführter Überstromschutz mit Abzweigstromkreis und 15 A Stärke muss für jede Stromeingabe in der Gebäudeverkabelung integriert sein.

AUSTAUSCHBARE BATTERIEN

Wird ein Gerät mit einer austauschbaren Batterie geliefert und für diese Batterie durch einen falschen Batterietyp ersetzt, könnte dies zu einer Explosion führen. Dies trifft zu für manche Arten von Lithiumsbatterien zu, und das folgende gilt es zu beachten:

- Wird die Batterie in einem Bereich für Bediener eingesetzt, findet sich in der Nähe der Batterie eine Markierung oder Erklärung sowohl im Betriebshandbuch als auch in der Wartungsanleitung.
- Ist die Batterie an einer anderen Stelle im Gerät eingesetzt, findet sich in der Nähe der Batterie eine Markierung oder einer Erklärung in der Wartungsanleitung.

Diese Markierung oder Erklärung enthält den folgenden Warntext: VORSICHT

EXPLOSIONSGEFAHR, FALLS BATTERIE DURCH EINEN FALSCHEN BATTERIETYP ERSETZT WIRD. GEBRAUCHTE BATTERIEN DEN ANWEISUNGEN ENTSPRECHEND ENTSORGEN.

- Denmark - “Unit is class I - mit Wechselstromkabel benutzen, dass für die Abweichungen in Dänemark eingestellt ist. Das Kabel ist mit einem Erdungsdraht versehen. Das Kabel wird in eine geerdete Wandsteckdose angeschlossen. Keine Steckdosen ohne Erdungsleitung verwenden!”
- Finland - (Markierungsetikett und im Handbuch) - Laite on liitettävä suojamaadoituskoskettimilla varustettuun pistorasiaan
- Norway - (Markierungsetikett und im Handbuch) - Apparatet må tilkoples jordet stikkontakt Ausschließlich für Anschluss an IT-Netzstromsysteme in Norwegen vorgesehen
- Sweden - (Markierungsetikett und im Handbuch) - Apparaten skall anslutas till jordat uttag.

Anschluss des Stromkabels:

1. Schließen Sie das Stromkabel an den Hauptanschluss auf der Rückseite des Gerätes an.
2. Schließen Sie das Stromkabel an den geerdeten Wechselstromanschluss an.

VORSICHT

Stromschlag- und Energiegefahr Die Trennung einer Stromquelle trennt nur ein Stromversorgungsmodul von der Stromversorgung. Um das Gerät komplett zu isolieren, muss es von der gesamten Stromversorgung getrennt werden.

Vorsicht - Zur Reduzierung der Stromschlag- und Feuergefahr

1. Dieses Gerät ist dazu ausgelegt, die Verbindung zwischen der geerdeten Leitung des Gleichstromkreises und dem Erdungsleiter des Gerätes zu ermöglichen. Siehe Montageanleitung.
2. Wartungsarbeiten jeglicher Art dürfen nur von qualifiziertem Servicepersonal ausgeführt werden. Es gibt innerhalb des Gerätes keine vom Benutzer zu wartenden Teile.
3. Versuchen Sie nicht, ein offensichtlich beschädigtes Gerät an den Stromkreis anzuschließen, einzuschalten oder zu betreiben.
4. Vergewissern Sie sich, dass sie Lüftungsöffnungen im Gehäuse des Gerätes NICHT BLOCKIERT SIND.
5. Ersetzen Sie eine durchgebrannte Sicherung ausschließlich mit dem selben Typ und von der selben Stärke, die auf dem Sicherheitsetikett angeführt sind, das sich neben dem Stromkabelanschluss, am Sicherungsgehäuse.
6. Betreiben Sie das Gerät nicht an einem Standort, an dem die Höchsttemperatur der Umgebung 40°C überschreitet.
7. Vergewissern Sie sich, das Stromkabel aus dem Wandstecker zu ziehen, BEVOR Sie die Hauptsicherung entfernen und/oder prüfen.

Altitude and Climate Warning

Note: This warning only applies to The People's Republic of China.

1. 对于在非热带气候条件下运行的设备而言， Tma：为制造商规范允许的最大环境温度，或者为 25°C，采用两者中的较大者。
2. 关于在海拔不超过 2000m 或者在非热带气候地区使用的设备，附加警告要求如下：

关于在海拔不超过 2000m 的地区使用的设备，必须在随时可见的位置处粘贴包含如下内容或者类似用语的警告标记、或者附件 DD 中的符号。

“ 只可在海拔不超过 2000m 的位置使用。”

关于在非热带气候地区使用的设备，必须在随时可见的位置处粘贴包含如下内容的警告标记：

“ 只可在非热带气候地区使用。”

附件 DD：有关新安全警告标记的说明。

DD.1 海拔警告标记

标记含义：设备的评估仅基于 2000m 以下的海拔高度，因此设备只适用于该运行条件。如果在海拔超过 2000m 的位置使用设备，可能会存在某些安全隐患。

DD.2 气候警告标记

标记含义：设备的评估仅基于温带气候条件，因此设备只适用于该运行条件。如果在热带气候地区使用设备，可能会存在某些安全隐患。

Document Conventions

The following describes the conventions and symbols that this guide uses:

Item	Description	Description (French)	Beschreibung (German)
Example	An example scenario	Un scénario d'exemple	Ein Beispielszenarium
Caution:	Possible damage to equipment, software, or data	Endommagement possible de l'équipement, des données ou du logiciel	Mögliche Schäden an Gerät, Software oder Daten
Note:	Additional information	Informations complémentaires	Zusätzliche Informationen
To	A statement and instructions	Références et instructions	Eine Erklärung und Anweisungen
Tip:	A suggestion or workaround	Une suggestion ou solution	Ein Vorschlag oder eine Umgehung
Warning:	Possible physical harm to the operator	Blessure possible de l'opérateur	Verletzungsgefahr des Bedieners

Table of Contents

Chapter 1 – Preface

This guide describes how to configure and use the Alteon Application Switch Operating System (AlteonOS) software on the Alteon Application Switches. Throughout this guide, in most cases the AlteonOS and the Alteon platform are referred to as Alteon. For documentation on installation and initial configuration of Alteon, see the *Radware Alteon Installation and Maintenance Guide*.

Who Should Use This Guide

This guide is intended for network installers and system administrators engaged in configuring and maintaining a network. The administrator should be familiar with Ethernet concepts, IP addressing, the Spanning Tree Protocol, and SNMP configuration parameters.

What You Will Find in This Guide

This guide helps you to plan, implement, and administer Alteon. Where possible, each section provides feature overviews, usage examples, and configuration instructions.

Part 1—Basic Features

- Accessing Alteon describes how to access Alteon to configure, view information, and run statistics using the CLI, the Browser-Based Interface (BBI), SNMP, and the management port.
- Securing Alteon describes how to protect the system from attacks, unauthorized access, and discusses different methods to manage Alteon for remote administrators using specific IP addresses, RADIUS authentication, Secure Shell (SSH), and Secure Copy (SCP).
- VLANs describes how to configure Virtual Local Area Networks (VLANs) for creating separate network segments, including how to use VLAN tagging for Alteons that use multiple VLANs.
- Port Trunking describes how to group multiple physical ports together to aggregate the bandwidth between large-scale network devices.
- Port Teaming describes how to configure port teaming.
- Spanning Tree Protocol discusses how spanning trees configure the network to use the most efficient path when multiple paths exist.

Part 2—IP Routing

- Basic IP Routing describes how to configure IP routing using IP subnets and DHCP Relay.
- Routing Information Protocol describes the implementation of standard RIP for exchanging TCP/IP route information with other routers.
- Border Gateway Protocol describes Border Gateway Protocol (BGP) concepts and BGP features.
- Open Shortest Path First (OSPF) describes OSPF concepts, how OSPF is implemented, and examples of how to configure OSPF support.

Part 3—Application Load Balancing Fundamentals

- Server Load Balancing describes how to balance network traffic among a pool of available servers for more efficient, robust, and scalable network services.

- Load Balancing Special Services describes how to extend Server Load Balancing (SLB) configurations to load balance services including source IP addresses, FTP, RTSP, DNS, WAP, IDS, and Session Initiation Protocol (SIP).
- WAN Link Load Balancing describes how to balance user session traffic among a pool of available WAN Links.
- Offloading SSL Encryption and Authentication describes SSL offloading capabilities, which perform encryption, decryption, and verification of Secure Sockets Layer (SSL) transmissions between clients and servers, relieving the back-end servers of these tasks.
- Filtering and Traffic Manipulation describes how to configure and optimize network traffic filters for security and Network Address Translation (NAT).
- ADC-VX Management describes how to use ADC-VX in an Alteon environment. A vADC is a virtualized instance of the AlteonOS that behaves in the same manner as a traditional standalone Alteon ADC, with the exception that while it is bound to a specific hardware resource, the amount of resources allocated to the vADC may vary based on the user's or application's resource needs.
- Application Redirection describes how to use filters for redirecting traffic to such network streamlining devices as caches.
- Health Checking describes how to recognize the availability of the various network resources used with the various load balancing and application redirection features.
- High Availability describes how to use the Virtual Router Redundancy Protocol (VRRP) to ensure that network resources remain available if one Alteon is removed for service.

Part 4—Advanced Load Balancing

- Persistence describes how to ensure that all connections from a specific client session reach the same server. Persistence can be based on cookies or SSL session ID.
- Advanced Denial of Service Protection describes the protection features that can be used to prevent a wide range of network attacks.
- Firewall Load Balancing describes how to combine features to provide a scalable solution for load balancing multiple firewalls.
- Virtual Private Network Load Balancing describes load balancing secure point-to-point links.
- Global Server Load Balancing describes configuring server load balancing across multiple geographic sites.
- Bandwidth Management describes how to allocate specific portions of the available bandwidth for specific users or applications.
- XML Configuration API describes how to use and configure the XML Configuration API.
- AppShape++ Scripting describes the AppShape++ framework for customizing application delivery using user-written scripts.

Appendices

- Layer 7 String Handling describes how to perform load balancing and application redirection based on Layer 7 packet content information (such as URL, HTTP Header, browser type, and cookies).
- Content-Intelligent Server Load Balancing Not Using Layer 7 Content Switching Rules describes the sole content-intelligent server load balancing methodology prior to version 28.1.
- IPv6 describes how to configure the IP version 6 features.

Related Documentation

- *Alteon Application Switch Operating System Release Notes*
- *Radware Alteon Maintenance and Installation Guide*
- *Alteon Application Switch Operating System Command Reference*
- *Alteon Application Switch Operating System Browser-Based Interface (BBI) Quick Guide*
- *Alteon Application Switch Operating System Troubleshooting Guide*

Chapter 2 – Accessing Alteon

The AlteonOS lets you access, configure, and view information and statistics about Alteon. The following topics are discussed in this chapter:

Using the CLI

The Command Line Interface (CLI) is a built-in, text-based menu system for access via a local terminal or remote Telnet or Secure Shell (SSH) session. The CLI is the most direct method for collecting information and configuring Alteon. The following is the CLI *Main Menu* with Administrator privileges:

```
[Main Menu]
     info   - Information Menu
     stats  - Statistics Menu
     cfg    - Configuration Menu
     oper   - Operations Command Menu
     boot   - Boot Options Menu
     maint  - Maintenance Menu
     diff   - Show pending config changes [global command]
     apply  - Apply pending config changes [global command]
     save   - Save updated config to FLASH [global command]
     revert - Revert pending or applied changes [global command]
     exit   - Exit [global command, always available]
```

You can access the CLI in the following ways:

- **Using a serial connection via the console port**—You can access and configure Alteon by using a computer running terminal emulation software.
- **Using the management port**—The management port is a Gigabit Ethernet port that is used exclusively for managing Alteon.

  For more information on the management port, see Using the Management Port, page 52.
- **Using a Telnet connection over the network**—A Telnet connection offers the convenience of accessing Alteon from any workstation connected to the network. Telnet access provides the same options for user and administrator access as those available through the console port.

To establish a Telnet connection with Alteon

From the CLI of your workstation, enter **telnet**, followed by the Alteon IP address:

```
telnet <Alteon_IP_address>
```

- **Using an SSH connection to securely log into another computer over a network**—The SSH (Secure Shell) protocol enables you to securely log into another computer over a network to execute commands remotely. As a secure alternative to using Telnet to manage the Alteon configuration, SSH ensures that all data sent over the network is encrypted and secure. For more information, see Secure Shell and Secure Copy, page 70.

For more information on CLI menus and commands, see the *Alteon Application Switch Operating System Command Reference.*

Using SNMP

Alteon provides Simple Network Management Protocol (SNMP) v1.0 and SNMP v3.0 support for access through any network management software, such as APSolute Vision or HP-OpenView.

SNMP v1.0

To access the SNMP agent, the read and write community strings on the SNMP manager should be configured to match those on Alteon. The default read community string on Alteon is set to **public**, and the default write community string is set to **private**.

Caution: Leaving the default community strings enabled on Alteon presents a security risk. You can change the community strings as follows:

- **Read community string**—/cfg/sys/ssnmp/rcomm <string>
- **Write community string**—/cfg/sys/ssnmp/wcomm <string>

The SNMP manager should reach the management interface (management port) or any one of the Alteon IP interfaces.

SNMP v3.0

SNMPv3 is an enhanced version of SNMP, approved by the Internet Engineering Steering Group in March, 2002. SNMP v3.0 contains additional security and authentication features that provide data origin authentication, data integrity checks, timeliness indicators, and encryption to protect against threats such as masquerade, modification of information, message stream modification, and disclosure.

SNMPv3 ensures that the client can use SNMP v3 to query the MIBs, mainly for security purposes.

To access the SNMP v3.0 menu, enter the following command in the CLI:

```
>> # /cfg/sys/ssnmp/snmpv3
```

For more information on SNMP MIBs and the commands used to configure SNMP on Alteon, see the *Alteon Application Switch Operating System Command Reference*.

Default Configuration

Alteon has two default users: **adminmd5**, which uses MD5 authentication, and **adminsha**, which uses SHA authentication.

Both these users have access to all the MIBs supported by Alteon. By default, the passwords for these users are **adminmd5** and **adminsha**, respectively.

To configure an SNMP username

```
>> # /cfg/sys/ssnmp/snmpv3/usm <x>
```

User Configuration

Configure users to use the authentication and privacy options. Alteon supports two authentication algorithms: MD5 and SHA.

To configure authentication and privacy options

This example procedure configures a user with the name **test**, authentication type **MD5**, authentication password **test**, privacy option **DES**, and with privacy password **test**.

1. Enter the following CLI commands:

```
>> # /cfg/sys/ssnmp/snmpv3/usm 5
>> SNMPv3 usmUser 5 # name "test"
>> SNMPv3 usmUser 5 # auth md5
>> SNMPv3 usmUser 5 # authpw test
>> SNMPv3 usmUser 5 # priv des
>> SNMPv3 usmUser 5 # privpw test
```

2. After configuring a user, specify the access level for this user along with the views to which the user is allowed access. This is specified in the access table.

```
>> # /cfg/sys/ssnmp/snmpv3/access 5
>> SNMPv3 vacmAccess 5 # name "testgrp"
>> SNMPv3 vacmAccess 5 # level authPriv
>> SNMPv3 vacmAccess 5 # rview "iso"
>> SNMPv3 vacmAccess 5 # wview "iso"
>> SNMPv3 vacmAccess 5 # nview "iso"
```

3. Link the user to a particular access group.

```
>> # /cfg/sys/ssnmp/snmpv3/group 5
>> SNMPv3 vacmSecurityToGroup 5 # uname test
>> SNMPv3 vacmSecurityToGroup 5 # gname testgrp
```

If you want to allow the user to access only certain MIBs, see View-Based Configurations, page 46.

View-Based Configurations

To configure an SNMP user equivalent to the user **CLI access level**

```
/cfg/sys/ssnmp/snmpv3/usm 4
name "usr"
/cfg/sys/ssnmp/snmpv3/access 3
name "usrgrp"
rview "usr"
wview "usr"
nview "usr"
/cfg/sys/ssnmp/snmpv3/group 4
uname usr
gname usrgrp
/cfg/sys/ssnmp/snmpv3/view 6
name "usr"
tree "1.3.6.1.4.1.1872.2.5.1.2"
/cfg/sys/ssnmp/snmpv3/view 7
name "usr"
tree "1.3.6.1.4.1.1872.2.5.1.3"
/cfg/sys/ssnmp/snmpv3/view 8
name "usr"
tree "1.3.6.1.4.1.1872.2.5.2.2"
/cfg/sys/ssnmp/snmpv3/view 9
name "usr"
tree "1.3.6.1.4.1.1872.2.5.2.3"
/cfg/sys/ssnmp/snmpv3/view 10
name "usr"
tree "1.3.6.1.4.1.1872.2.5.3.2"
/cfg/sys/ssnmp/snmpv3/view 11
name "usr"
tree "1.3.6.1.4.1.1872.2.5.3.3"
/cfg/sys/ssnmp/snmpv3/view 12
name "usr"
tree "1.3.6.1.4.1.1872.2.5.4.2"
/cfg/sys/ssnmp/snmpv3/view 13
name "usr"
tree "1.3.6.1.4.1.1872.2.5.4.3"
/cfg/sys/ssnmp/snmpv3/view 14
name "usr"
tree "1.3.6.1.4.1.1872.2.5.5.2"
/cfg/sys/ssnmp/snmpv3/view 15
name "usr"
tree "1.3.6.1.4.1.1872.2.5.5.3"
/cfg/sys/ssnmp/snmpv3/view 16
name "usr"
tree "1.3.6.1.4.1.1872.2.5.6.2"
```

To configure an SNMP user equivalent to the oper **CLI access level**

```
/cfg/sys/ssnmp/snmpv3/usm 5
name "slboper"
/cfg/sys/ssnmp/snmpv3/access 4
name "slbopergrp"
rview "slboper"
wview "slboper"
nview "slboper"
/cfg/sys/ssnmp/snmpv3/group 4
uname slboper
gname slbopergrp
/cfg/sys/ssnmp/snmpv3/view 20
name "slboper"
tree "1.3.6.1.4.1.1872.2.5.1.2" /cfg/sys/ssnmp/snmpv3/view 21
name "slboper"
tree "1.3.6.1.4.1.1872.2.5.1.3"
/cfg/sys/ssnmp/snmpv3/view 22
name "slboper"
tree "1.3.6.1.4.1.1872.2.5.2.2"
/cfg/sys/ssnmp/snmpv3/view 23
name "slboper"
tree "1.3.6.1.4.1.1872.2.5.2.3"
/cfg/sys/ssnmp/snmpv3/view 24
name "slboper"
tree "1.3.6.1.4.1.1872.2.5.3.2"
/cfg/sys/ssnmp/snmpv3/view 25
name "slboper"
tree "1.3.6.1.4.1.1872.2.5.3.3"
/cfg/sys/ssnmp/snmpv3/view 26
name "slboper"
tree "1.3.6.1.4.1.1872.2.5.4"
/cfg/sys/ssnmp/snmpv3/view 27
name "slboper"
tree "1.3.6.1.4.1.1872.2.5.4.1"
type excluded
/cfg/sys/ssnmp/snmpv3/view 28
name "slboper"
tree "1.3.6.1.4
.1.1872.2.5.5.2"
/cfg/sys/ssnmp/snmpv3/view 29
name "slboper"
tree "1.3.6.1.4.1.1872.2.5.5.3"
/cfg/sys/ssnmp/snmpv3/view 30
name "slboper"
tree "1.3.6.1.4.1.1872.2.5.6.2"
```

Configuring SNMP Trap Hosts

This section describes how to configure the following SNMP trap hosts:

- SNMPv1 Trap Host, page 48
- SNMPv2 Trap Host, page 49
- SNMPv3 Trap Host, page 49

SNMPv1 Trap Host

To configure an SNMPv1 trap host

1. Configure a user with no authentication and password.

```
>> # /cfg/sys/ssnmp/snmpv3/usm 10 name "v1trap"
```

2. Configure an access group and group table entries for the user. Use the **nview** command to specify which traps can be received by the user. In the following example, the user receives the traps sent by Alteon:

```
>> # /cfg/sys/ssnmp/snmpv3/access 10
>> SNMPv3 vacmAccess 10 # name "v1trap"
>> SNMPv3 vacmAccess 10 # model snmpv1
>> SNMPv3 vacmAccess 10 # nview "iso"

>> # /cfg/sys/ssnmp/snmpv3/group 10
>> SNMPv3 vacmSecurityToGroup 10 # model snmpv1
>> SNMPv3 vacmSecurityToGroup 10 # uname v1trap
>> SNMPv3 vacmSecurityToGroup 10 # gname v1trap
```

3. Configure an entry in the notify table.

```
>> # /cfg/sys/ssnmp/snmpv3/notify 10
>> SNMPv3 vacmSecurityToGroup 10 # name v1trap
>> SNMPv3 vacmSecurityToGroup 10 # tag v1trap
```

4. Specify the IP address and other trap parameters in the **targetAddr** and **targetParam** tables. Use the **uname** command to specify the user name used with this targetParam table.

```
>> # /cfg/sys/ssnmp/snmpv3/taddr 10                (Access the TargetAddrTable menu)

>> SNMPv3 snmpTargetAddrTable 10 # name v1trap
>> SNMPv3 snmpTargetAddrTable 10 # addr 50.80.23.245
>> SNMPv3 snmpTargetAddrTable 10 # taglist v1trap
>> SNMPv3 snmpTargetAddrTable 10 # pname v1param

>> # /cfg/sys/ssnmp/snmpv3/tparam 10               (Access the TargetParamsTable menu)

>> SNMPv3 snmpTargetParamsTable 10 # name v1param
>> SNMPv3 snmpTargetParamsTable 10 # mpmodel snmpv1
>> SNMPv3 snmpTargetParamsTable 10 # uname v1trap
>> SNMPv3 snmpTargetParamsTable 10 # model snmpv1
```

5. Specify the community string used in the traps using the community table.

```
>> # /cfg/sys/ssnmp/snmpv3/comm 10                 (Select the community table)
>> SNMPv3 snmpCommunityTable 10 # index v1trap
>> SNMPv3 snmpCommunityTable 10 # name public
>> SNMPv3 snmpCommunityTable 10 # uname v1trap
```

SNMPv2 Trap Host

The SNMPv2 trap host configuration is similar to the SNMPv1 trap host configuration. Wherever you specify the model, specify **snmpv2** instead of **snmpv1**.

```
/cfg/sys/ssnmp/snmpv3/usm 10
name "v2trap"
/cfg/sys/ssnmp/snmpv3/access 10
        name "v2trap"
        model snmpv2
        nview "iso"
/cfg/sys/ssnmp/snmpv3/group 10
        model snmpv2
        uname v2trap
        gname v2trap
/cfg/sys/ssnmp/snmpv3/taddr 10
        name v2trap
        addr 50.81.25.66
        taglist v2trap
        pname v2param
/cfg/sys/ssnmp/snmpv3/tparam 10
        name v2param
        mpmodel snmpv2c
        uname v2trap
        model snmpv2
/cfg/sys/ssnmp/snmpv3/notify 10
        name v2trap
tag v2trap/cfg/sys/ssnmp/snmpv3/comm 10
        index v2trap
        name public
        uname v2trap
```

SNMPv3 Trap Host

To configure a user for SNMPv3 traps

You can choose to send the traps with both privacy and authentication, with authentication only, or with neither.

1. Configure an SNMPv3 trap host the access table as follows:

```
>> # /cfg/sys/ssnmp/snmpv3/access <x> /level
Enter new access level [noAuthNoPriv|authNoPriv|authPriv]:
access-level>

>> # /cfg/sys/ssnmp/snmpv3/tparam <snmpTargetParams number: (1-16)>
```

2. Configure the user in the user table from the *SNMPv3 usmUser 1* menu:

```
>> /cfg/sys/ssnmp/snmpv3/usm <usmUser number: (1-16)>
```

Note: It is not necessary to configure the community table for SNMPv3 traps because the community string is not used by SNMPv3.

Example

The following example illustrates how to configure an SNMPv3 user **v3trap** with authentication only:

```
/cfg/sys/ssnmp/snmpv3/usm 11
        name "v3trap"
        auth md5
        authpw v3trap
/cfg/sys/ssnmp/snmpv3/access 11
        name "v3trap"
        level authNoPriv
        nview "iso"
/cfg/sys/ssnmp/snmpv3/group 11
        uname v3trap
        gname v3trap
/cfg/sys/ssnmp/snmpv3/taddr 11
        name v3trap
        addr 50.81.25.66
        taglist v3trap
        pname v3param
/cfg/sys/ssnmp/snmpv3/tparam 11
        name v3param
        uname v3trap
        level authNoPriv
/cfg/sys/ssnmp/snmpv3/notify 11
        name v3trap
        tag v3trap
```

Using the Browser-Based Interface

The Browser-Based Interface (BBI) is a Web-based management interface for interactive Alteon access through your Web browser.

Configuring BBI Access via HTTP

To enable BBI access on Alteon via HTTP

```
/cfg/sys/access/http ena
```

To change the HTTP web server port from the default port 80

```
/cfg/sys/access/wport <x>
```

To access your Alteon via the Browser-Based Interface

1. Open a Web browser window.
2. Type the Alteon hostname or the IP address.

Configuring BBI Access via HTTPS

You can access the BBI via a secure HTTPS connection over management and data ports.

To enable BBI access on Alteon via HTTPS

```
/cfg/sys/access/https/https ena
```

To change the HTTPS Web server port number from the default port 443

```
/cfg/sys/access/https/port <x>
```

Generating a Certificate for BBI Access via HTTPS

Accessing the BBI via HTTPS requires that you generate a certificate for use during the key exchange. The system creates a default certificate the first time you enable HTTPS, but you can create a new certificate defining the information you want to be used in the various fields using the following command:

```
>>/cfg/sys/access/https/generate
This operation will generate a self-signed server certificate.
Enter key size [512|1024|2048|4096] [1024]:
Enter server certificate hash algorithm [md5|sha1|sha256|sha384|sha512]
[sha1]:
Enter certificate Common Name (e.g. your site's name):
Use certificate default values? [y/n]:
Enter certificate Country Name (2-letter code) []: us
Enter certificate State or Province Name (full name) []: newyork
Enter certificate locality name (e.g. city) []: newyork
Enter certificate Organization Name (e.g. company) []: example
Enter certificate Organizational Unit Name (e.g. accounting) []: exam
Enter certificate Email (e.g. admin@company.com) []: example@example.com
Enter certificate validation period in days (1-3650) [365]:
........
Self signed server certificate, certificate signing request and key added.
```

You can save the certificate to flash for use if you reboot Alteon by using the **apply** and **save** commands.

When a client (for example, a Web browser) connects to Alteon, the client is asked to accept the certificate and verify that the fields are what are expected. Once you grant BBI access to the client, the BBI can be used as described in the *Alteon Application Switch Browser-Based Interface Quick Guide*.

Using the Management Port

The management port is a Gigabit Ethernet port on Alteon that is used exclusively for managing Alteon. While you can manage Alteon from any network port, the management port conserves a data port that could otherwise be used for processing requests. You can use the management port to access Alteon using Telnet (CLI), SSH, or HTTP (BBI).

The management port does not participate in the switching and routing protocols that run on the data ports, but it can be used to perform management functions such as:

- Accessing the NTP server
- Sending out SNMP traps
- Sending out syslog messages
- Accessing the RADIUS server
- Accessing the TACACS+ server
- Accessing the DNS server
- Performing TFTP or FTP functions (ptimg, gtimg, ptcfg, gtcfg, ptdmp)
- Accessing the SMTP server

- Running the **ping**, **telnet**, and **traceroute** commands

Note: BOOTP is not supported over the management port.

For more information on using the commands to perform these functions, see the *Alteon Application Switch Operating System Command Reference*.

Setting Up the Management Port

This section describes how to set up the management port.

Notes

- To configure MNG 1 as a management port for dedicated out-of-band management on devices *other than* the Alteon Application Switch 4408 platform, use the command **/cfg/sys/mmgmt ena** to enable the management port. For more information, see the section on configuring management ports in the *Radware Alteon Installation and Maintenance Guide*.
- To configure port 6 / MNG 1 as a management port for dedicated out-of-band management on the Alteon Application Switch 4408 platform, first enable the physical port with the command **/boot/mgmt ena**, then use the command **/cfg/sys/mmgmt ena** to enable the management port. For more information, see the section on configuring management ports in the *Radware Alteon Installation and Maintenance Guide*.

To set up the management port

1. Configure a default gateway address. Both IPv4 and IPv6 addresses can be configured on the management port, each one with its own gateway.

`>> Main# /cfg/sys/mmgmt/gw 10.10.10.1`	(Configure an IPv4 default gateway)
`>> Main# /cfg/sys/mmgmt/gw6 2001::1111 (configure an IPv6 default gateway)`	(Configure an IPv6 default gateway)

2. Configure a static IP address. Both IPv4 and IPv6 addresses can be configured on the management port.

`>> Management Port# addr 10.10.10.5`	(Configure a static IPv4 address)
`>> Management Port# mask 255.255.255.0`	(Configure an IPv4 network mask)
`>> Management Port# addr6 2001::2213(`	(Configure a static IPv6 address)
`>> Management Port# prefix6 64`	(Configure IPv6 prefix length)

3. Enable the management port. When you enable the management port, you can use it to access Alteon via Telnet, SSH, or BBI, provided you enabled the commands on Alteon. These commands can occur simultaneously on both the management port and the data ports.

`>> Management Port# ena`	(Enable the management port)

Note: There are a maximum of four concurrent Telnet sessions over the management and data ports combined.

4. Configure the default port type for each management function.

   Select the management port or the default data port for each management function. For example, select the management port for NTP, RADIUS, and syslog functions only. SMTP, TFTP, and SNMP traps are configured to use the default data ports.

`>> Management Port# ntp mgmt`	(Select the management port for NTP)
`>> Management Port# radius mgmt`	(Select the management port for RADIUS)
`>> Management Port# syslog mgmt`	(Select the management port for syslog)

Note: The default for TFTP can be overridden by using the **–data** or **–mgmt** option after a **gtimg**, **ptimg**, **gtcfg**, **ptcfg**, or **ptdmp** command.

5. Apply, verify your configuration, and save the changes.

`>> Management Port # apply`	(Make your changes active)
`>> Management Port # cur`	(View current settings)
`>> Management Port # save`	(Save for restore after reboot)

Limiting Management Access

In a standalone appliance, you can disable access to a management service from a data port using one of the commands as described in Table 1:

Table 1: Commands to Limit Standalone Management Access

Command	Description
/cfg/sys/access/port/add <port number>	Enable port for management access.
/cfg/sys/access/port/rem <port number>	Disable port from management access.
/cfg/sys/access/port/arem	Disable all ports from management access.
/cfg/sys/access/port/cur	Current listing of data ports with management access.

ADC-VX supports virtual ADC (vADC) management access through VLANs. Unlike standalone appliances, a vADC does not necessarily own the entire physical port and can share it with other applications or services. To accommodate such a design, the data port management access for vADCs is supported by VLAN IDs and not by physical ports.

Table 2 lists the commands that can be used to limit management services from VLANs:

Table 2: Commands to Limit vADC Management Access

Command	Description
/cfg/sys/access/vlan/add <vlan number>	Enable VLAN for management access.
/cfg/sys/access/vlan/aadd <vlan number>	Enable all VLANs for management access.
/cfg/sys/access/vlan/rem <vlan number>	Disable VLAN from management access.
/cfg/sys/access/vlan/arem	Disable all VLANs from management access.
/cfg/sys/access/vlan/cur	Current listing of VLANs with management access.

File Transfers

Alteon supports the File Transfer Protocol (FTP) as an alternative to the Trivial File Transfer Protocol (TFTP). FTP is supported over data and management ports for the upload and download of the following file types:

- Configuration files
- Technical Support (TS) dumps
- Panic dumps

An FTP hostname, filename, username, and password are requested when using FTP.

Time Configuration

This section describes the Alteon time configuration options.

Time Zone Configuration

Upon set up, you should configure Alteon with the appropriate time zone configuration. This enables Alteon to provide proper time offsets and to adjust for Daylight Savings Time.

Example Set the Time Zone

Set the time zone to Atlantic Time for an Alteon that is physically located in Atlantic Canada.

1. Access time zone configuration.

```
>> Main# /cfg/sys/timezone
```

2. Select the general geographic zone in which Alteon is located.

```
Please identify a location so that time zone rules can be
set correctly.
Please select a continent or ocean.
 1) Africa
 2) Americas
 3) Antarctica
 4) Arctic Ocean
 5) Asia
 6) Atlantic Ocean
 7) Australia
 8) Europe
 9) Indian Ocean
10) Pacific Ocean
11) None - disable timezone setting
Enter the number of your choice:  2
```

Note: The time zone setting can be disabled in this menu by selecting **11**.

3. Select the country inside the selected geographic zone.

```
Please select a country.
1) Anguilla               18) Ecuador               35) Paraguay
2) Antigua & Barbuda      19) El Salvador           36) Peru
3) Argentina              20) French Guiana         37) Puerto Rico
4) Aruba                  21) Greenland             38) St Kitts & Nevis
5) Bahamas                22) Grenada               39) St Lucia
6) Barbados               23) Guadeloupe            40) St Pierre & Miquelon
7) Belize                 24) Guatemala             41) St Vincent
8) Bolivia                25) Guyana                42) Suriname
9) Brazil                 26) Haiti                 43) Trinidad & Tobago
10) Canada                27) Honduras              44) Turks & Caicos Is1
11) Cayman Islands        28) Jamaica               45) United States
12) Chile                 29) Martinique            46) Uruguay
13) Colombia              30) Mexico                47) Venezuela
14) Costa Rica            31) Montserrat            48) Virgin Islands (UK)
15) Cuba                  32) Netherlands Antilles  49) Virgin Islands(US)
16) Dominica              33) Nicaragua
17) Dominican Republic    34) Panama
Enter the number of your choice:  10
```

4. Select the time zone appropriate to the specific geographic location of Alteon.

```
Please select one of the following time zone regions.
1) Newfoundland Island
2) Atlantic Time - Nova Scotia (most places), NB, W Labrador, E Que-bec & PEI
3) Atlantic Time - E Labrador
4) Eastern Time - Ontario & Quebec - most locations
5) Eastern Time - Thunder Bay, Ontario
6) Eastern Standard Time - Pangnirtung, Nunavut
7) Eastern Standard Time - east Nunavut
8) Eastern Standard Time - central Nunavut
9) Central Time - Manitoba & west Ontario
10) Central Time - Rainy River & Fort Frances, Ontario
11) Central Time - west Nunavut
12) Central Standard Time - Saskatchewan - most locations
13) Central Standard Time - Saskatchewan - midwest
14) Mountain Time - Alberta, east British Columbia & west Saskatchewan
15) Mountain Time - central Northwest Territories
16) Mountain Time - west Northwest Territories
17) Mountain Standard Time - Dawson Creek & Fort Saint John, British
Columbia
18) Pacific Time - west British Columbia
19) Pacific Time - south Yukon
20) Pacific Time - north Yukon
Enter the number of your choice:  2
```

5. **Apply** and **save** the configuration change.

Network Time Protocol

The Network Time Protocol (NTP) provides the accurate time by synchronizing with a time server on either an internal or external network. Using NTP ensures that Alteon always has the accurate time for the various functions that integrate and use time.

To view the current NTP settings

```
>> Main# /cfg/sys/ntp/cur
Current NTP state: disabled
Current primary NTP server: 0.0.0.0
Current resync interval: 1440 minutes
Current GMT timezone offset: -8:00
```

Example Configure NTP for an Alteon

1. Access the NTP menu. You can configure an IPv4 or IPv6 address for the NTP server.

```
>> Main# /cfg/sys/ntp
```

2. Set the IP address of the primary NTP server. This is the NTP server that Alteon would regularly synchronize with to adjust its time.

```
>> NTP Server# prisrv
Current NTP server address: 0.0.0.0
Enter new NTP server address:  192.168.249.13
```

3. Set the IP address of the secondary NTP server. This is the NTP server that Alteon would synchronize with in instances where the primary server is not available. You can configure an IPv4 or IPv6 address for the NTP server.

```
>> NTP Server# secsrv
Current NTP server address: 0.0.0.0
Enter new NTP server address:  192.168.249.45
```

4. Set the re-synchronization interval. The re-synchronization interval is the amount of time Alteon waits between queries to the NTP server.

```
>> NTP Server# intrval
Current resync interval (minutes): 1440
Enter new resync interval (minutes) [1-44640]: 2000
```

5. Optionally, set the NTP time zone offset. The NTP time zone offset from Greenwich Mean Time defaults to the setting configured when the Alteon time zone was set. If this has not been done, or you want to override the current value, do the following:

```
>> NTP Server# tzone
Current GMT timezone offset:    -8:00
Enter new GMT timezone offset in hours [-12:00, +12:00]:  +4:00
```

6. Enable NTP functionality.

```
>> NTP Server# onCurrent status: OFF
New status:    ON
```

Note: To disable NTP functionality, use the **off** command.

Chapter 3 – Securing Alteon

Secure management is necessary for environments in which significant management functions are performed across the Internet.

The following topics are addressed in this chapter:

Protecting Alteon-Owned Addresses from Attacks

Denial of Service (DoS) attacks can be targeted not only at real servers, but at any IP address that is owned by an Alteon. A DoS attack can potentially overwhelm Alteon resources. You can use the system-wide *rlimit* (rate limiting) command to prevent DoS attacks over Address Resolution Protocol (ARP), ICMP, TCP, and UDP traffic by setting the maximum rate at which packets can enter Alteon. After the configured limit has been reached, packets are dropped. The maximum rate (packets per second) can be configured differently for each of the supported protocols.

How Different Protocols Attack Alteon

Without the system-wide rate limiting commands enabled, the following protocol packets destined for an Alteon-owned management interface could potentially overwhelm its management processor's CPU capacity:

- ARP requests to the management interface IP address.
- ICMP pings to the management interface IP address.
- TCP SYN packets sent the management interface IP address, including Telnet sessions, HTTP requests via the Browser-Based Interface, and BGP peer connections to Alteon. TCP Rate Limiting should also be configured to limit TCP packets destined to an Alteon virtual server IP (VIP) address. For more information, see TCP Rate Limiting, page 613.
- UDP packets sent to an Alteon interface address, including Routing Information Protocol (RIP) and Simple Network Management Protocol (SNMP) packets.

Configuring Denial of Service Protection

To configure Denial of Service (DoS) protection

1. Set the rate limit for the desired protocol.

```
>> /cfg/sys/access/rlimit
Enter protocol [arp|icmp|tcp|udp]:     arp
Current max rate:      0
Enter new max rate:    1000                  (Set the rate to 1000 packets per
                                             second)
```

2. Repeat step 1 to configure rate limits on any other of the supported protocols.
3. **Apply** and **save** the configuration.

Viewing Dropped Packets

Use the **/stats/sp/maint** command to view the number of dropped packets for each protocol which are configured for system-wide rate limiting. The information is available on a per-Alteon processor (SP) basis.

Note: This is available only in the vADC Administrator environment.

```
>> Main# /stats/sp/maint
Enter SP number: (1-4) 2
------------------------------------------------------------------Maintenanc
e statistics for SP 2:
Receive Letter success from MP:      6487510
Receive Letter success from SP 1:          0
Receive Letter success from SP 3:          0
Receive Letter success from SP 4:          0
Receive Letter errors from MP:             0
Receive Letter errors from SP 1:           0
Receive Letter errors from SP 3:           0
Receive Letter errors from SP 4:           0
Send Letter success to MP:          13808935
Send Letter success to SP 1:               0
Send Letter success to SP 3:               0
Send Letter success to SP 4:               8
Send Letter failures to MP:               13
Send Letter failures to SP 1:              0
Send Letter failures to SP 3:              0
Send Letter failures to SP 4:              0
learnErrNoddw:          0 resolveErrNoddw:       0
ageMPNoddw:             0 deleteMiss:            0
pfdbFreeEmpty:          0
arpDiscards:            0 icmpDiscards:          0
tcpDiscards:            0 udpDiscards:           0
```

```
(continued)

Sp - Application Services Engine Statistics
-----------------------------------------------------
 Client frames sent  : Success:    0
 Client frames sent  : Failed:     0
 Server frames sent  : Success:    0
 Server frames sent  : Failed:     0
 Packets received:                 0
 Packets dropped:                  0
 Invalid frames received:          0
 Invalid Session index:            0
 Memory allocation failures:       0
 Letter sent to sp success:        0
 Letter sent to sp failed:         0
 Packet buffers allocated:         0
 Packet buffers freed:             0
 Packet allocation failures:       0
 sameWire:               0   flood:                 0
 learn_SA:              84   match_SA:      955663336
 match_DA:               0   move_SA:               0
 resolve_DA_req:         0   resolve_DA_resp:       0
 aged_entries:         440   old_entries:          70
 age_zero:             370   deleted_entries:      70
 delete mismatches:                  0
 VRRP MAC delete attempts:           0
 age mismatches:                     0
 fill mismatches:                    0
```

Setting Source IP Address Ranges for Management

To limit access to Alteon without having to configure filters for each Alteon port, you can set a source IP address or range that allows you to connect to Alteon IP interface through Telnet, SSH, SNMP, or the Browser-Based Interface (BBI). This also helps to prevent spoofing or attacks on the TCP/IP stack.

When an IP packet reaches Alteon, Alteon checks the source IP address against the range of addresses defined by the management network and mask. If the source IP address of the host or hosts are within this range, Alteon allows the packets to attempt to log in. Any packet addressed to an Alteon IP interface with a source IP address outside this range is discarded.

You can configure both IPv4 and IPv6 IP ranges with up to 128 management IP addresses and mask/prefix pairs.

Example

Definition of a range of allowed source IP addresses between 192.192.192.1 to 192.192.192.127:

```
>> Main# /cfg/sys/access/mgmt add
Enter Management Network Address:192.192.192.0
Enter Management Network Mask: 255.255.255.128
```

In this example, the following source IP addresses are granted or not granted access to Alteon:

- A host with a source IP address of 192.192.192.21 falls within the defined range and is granted access to Alteon.
- A host with a source IP address of 192.192.192.192 falls outside the defined range and is not granted access.

To ensure that the source IP address is valid, you would need to shift the host to an IP address within the valid range specified by the address and mask, or modify the address to be 192.192.192.128 and the mask to be 255.255.255.128. This would put the 192.192.192.192 host within the valid range allowed by the address and mask (192.192.192.128-255).

RADIUS Authentication and Authorization

Alteon supports the Remote Authentication Dial-in User Service (RADIUS) method to authenticate and authorize remote administrators for managing Alteon. This method is based on a client/server model. The Remote Access Server (RAS) (Alteon) is a client to the back-end database server. A remote user (the remote administrator) interacts only with the RAS, not the back-end server and database.

RADIUS authentication consists of the following components:

- A protocol with a frame format that uses UDP over IP (based on *RFC 2138* and *RFC 2866*)
- A centralized server that stores all the user authorization information
- A client, in this case, Alteon

RADIUS Authentication Features

Alteon supports the following RADIUS authentication features:

- Supports RADIUS client in Alteon, based on the protocol definitions in *RFC 2138* and *RFC 2866*.
- Allows RADIUS secret passwords up to 32 bytes and less than 16 octets.
- Supports a **secondary authentication server** so that when the primary authentication server is unreachable, Alteon can send client authentication requests to the secondary authentication server. Use the **`/cfg/sys/radius/cur`** command to show the currently active RADIUS authentication server.
- Supports the following user-configurable RADIUS server retry and time-out values:
  - Time-out value: 1 to 10 seconds
  - Retries: 1 to 3

  Alteon times out if it does not receive a response from the RADIUS server within 1 to 3 retries. Alteon also retries connecting to the RADIUS server before it declares the server down.
- Supports a user-configurable RADIUS application port.

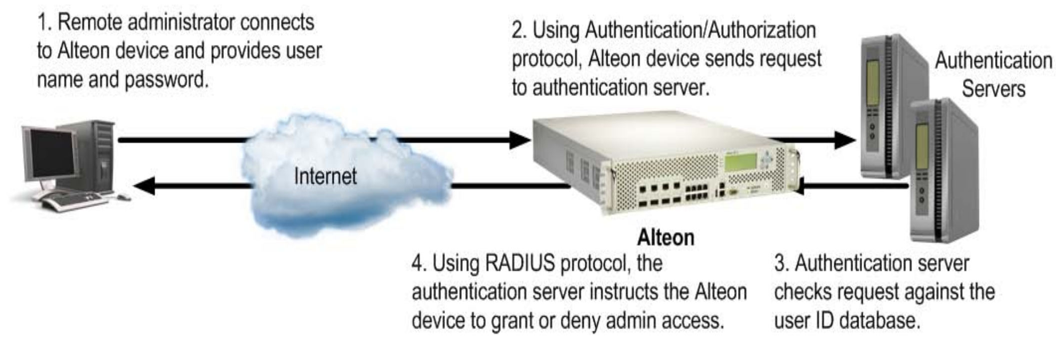
  The default is 1812/UDP, based on *RFC 2138*.
- Allows the network administrator to define privileges for one or more specific users to access Alteon at the RADIUS user database.
- Supports SecurID if the RADIUS server can do an ACE/Server client proxy. The password is the PIN number, plus the token code of the SecurID card.

How RADIUS Authentication Works

Figure 1 - RADIUS Authentication Process, page 63 illustrates the RADIUS Authentication process.

In the figure, Alteon acts as the RADIUS client, and communicates to the RADIUS server to authenticate and authorize a remote administrator using the protocol definitions specified in *RFC 2138* and *RFC 2866*. Transactions between the client and the RADIUS server are authenticated using a shared key that is not sent over the network. In addition, the remote administrator passwords are sent encrypted between the RADIUS client (Alteon) and the back-end RADIUS server.

Figure 1: RADIUS Authentication Process



Configuring RADIUS Authentication in Alteon

The following is an example RADIUS authentication configuration.

1. Turn RADIUS authentication on, then configure the primary and secondary RADIUS servers. You can configure IPv4 or IPv6 addresses for the RADIUS servers.

```
>> Main# /cfg/sys/radius                    (Select the RADIUS Server menu)
>> RADIUS Server# on                        (Turn RADIUS on)
Current status: OFF
New status:     ON
>> RADIUS Server# prisrv 10.10.1.1          (Enter the primary server IP)
Current primary RADIUS server:     0.0.0.0
New pending primary RADIUS server: 10.10.1.1
>> RADIUS Server# secsrv 10.10.1.2          (Enter the secondary server IP)
Current secondary RADIUS server:     0.0.0.0
New pending secondary RADIUS server:
10.10.1.2
```

2. Configure the RADIUS secret.

```
>> RADIUS Server# secret
Enter new RADIUS secret: <1-32 character secret>
```

Caution: If you configure the RADIUS secret using any method other than a direct console connection, the secret may be transmitted over the network as clear text.3.Optionally, you can change the default TCP port number used to listen to RADIUS.

The well-known port for RADIUS is 1812.

```
>> RADIUS Server# port
Current RADIUS port: 1812
Enter new RADIUS port [1500-3000]: <port number>
```

4. Configure the number of retry attempts for contacting the RADIUS server, and the timeout period.

`>> RADIUS Server# retries` `Current RADIUS server retries: 3`	
`Enter new RADIUS server retries [1-3]:`	(Server retries)
`>> RADIUS Server# time` `Current RADIUS server timeout: 3`	
`Enter new RADIUS server timeout [1-10]: 10`	(Enter the timeout period in minutes)

5. **Apply** and **save** the configuration.

User Accounts

The user accounts listed in Table 3 describe the user levels

- that can be defined in the RADIUS server dictionary file. For more information, see RADIUS Attributes for User Privileges, page 65.
- for defining the class of service for the End User Access Control feature. For more information, see End User Access Control, page 76.

Table 3: Alteon User Accounts and Access Levels

User Account	Description and Tasks Performed	Password
User	The User has no direct responsibility for Alteon management. The User can view all Alteon status information and statistics but cannot make any configuration changes to Alteon.	user
SLB Viewer	The SLB Viewer can view Alteon information, Server Load Balancing (SLB) statistics and information but cannot make any configuration changes to Alteon.	slbview
SLB Operator	The SLB Operator manages content servers and other Internet services and their loads. In addition to viewing all Alteon information and statistics, the SLB Operator can enable or disable servers using the SLB operation menu. Available to the vADC administrator only.	slboper
Layer 1 Operator	The Layer 1 Operator access allows the user to display information on Layer 1 parameters, such as LACP link information.	l1oper
Layer 2 Operator	The Layer 2 Operator access allows the user to display information related to Layer 2, such as routing and ARP.	l2oper
Layer 3 Operator	The Layer 3 Operator access allows the user to display information related to Layer 3. Available to the vADC administrator only.	l3oper

Table 3: Alteon User Accounts and Access Levels

User Account	Description and Tasks Performed	Password
Layer 4 Operator	The Layer 4 Operator manages traffic on the lines leading to the shared Internet services. This user currently has the same access level as the SLB operator. This level is reserved for future use to provide access to operational commands for operators managing traffic on the line leading to the shared Internet services. Available to the vADC administrator only.	l4oper
Operator	The Operator manages all functions of Alteon. In addition to SLB Operator functions, the Operator can reset ports.	oper
SLB Administrator	The SLB Administrator configures and manages content servers and other Internet services and their loads. In addition to SLB Operator functions, the SLB Administrator can configure parameters on the SLB menus, with the exception of configuring filters or bandwidth management. Available to the vADC administrator only.	slbadmin
Layer 3 Administrator	The Layer 3 Administrator manages Layer 3 features. Available to the vADC administrator only.	l3admin
Layer 4 Administrator	The Layer 4 Administrator configures and manages traffic on the lines leading to the shared Internet services. In addition to SLB Administrator functions, the Layer 4 Administrator can configure all parameters on the SLB menus, including filters and bandwidth management. Available to the vADC administrator only.	l4admin
Administrator	The superuser Administrator has complete access to all menus, information, and configuration commands, including the ability to change both the user and administrator passwords.	admin
Certificate Administrator	The Certificate Administrator has full access to the *Certificate Repository* menu (`/cfg/slb/ssl/certs`), including the ability to view, import, export, create, update, and decrypt the SSLdump capture. In addition, the Certificate Administrator has standard User privileges (he can view statuses and statistics). Unlike other user accounts, there is no default user called "crtadmin" and there is no default password. A Certificate Administrator user can only log in after the Administrator defines a user with certificate administrator privileges.	No default password

RADIUS Attributes for User Privileges

When a user logs in, Alteon authenticates the user's access level by sending the RADIUS access request (the client authentication request) to the RADIUS authentication server. If the remote user is successfully authenticated by the authentication server, Alteon verifies the privileges of the remote user and authorizes the appropriate access.

Backdoor Access

When both the primary and secondary authentication servers are not reachable, the administrator has the option to allow backdoor access on a per user basis. This access is disabled by default and must be activated for each individual user the administrator wishes to grant it to.

Note: If a user cannot establish a connection to the RADIUS server, failover to the local backdoor users are not permitted. This is done to avoid a DoS attack on RADIUS or Alteon allowing access.

Examples

A The following command enables backdoor access for user 9:

```
>> Main# /cfg/sys/access/user/uid 9/backdoor e
```

B The following command disables access for user 9:

```
>> Main# /cfg/sys/access/user/uid 9/backdoor d
```

Defining User Privileges in the RADIUS Dictionary

All user privileges, other than those assigned to the administrator, have to be defined in the RADIUS dictionary. RADIUS attribute 6, which is built into all RADIUS servers, defines the administrator. The filename of the dictionary is RADIUS vendor-dependent.

The following RADIUS attributes are defined for Alteon user privileges levels:

Table 4: Alteon-Proprietary Attributes for RADIUS

Username/Access	User Service Type	Value
l1oper	Vendor-supplied	259
l2oper	Vendor-supplied	258
l3oper	Vendor-supplied	257
l3admin	Vendor-supplied	256
user	Vendor-supplied	255
slboper	Vendor-supplied	254
l4oper	Vendor-supplied	253
oper	Vendor-supplied	252
slbadmin	Vendor-supplied	251
l4admin	Vendor-supplied	250
crtadmin	Vendor-supplied	249
slbadmin + crtmng	Vendor-supplied	248
l4admin + crtmng	Vendor-supplied	247
slbview	Vendor-supplied	246
admin	Vendor-supplied	6 (pre-defined)

TACACS+ Authentication

Alteon supports authentication and authorization with networks using the Cisco Systems® TACACS+ protocol. Alteon functions as the Network Access Server by interacting with the remote client and initiating authentication and authorization sessions with the TACACS+ access server. The remote user is defined as someone requiring management access to Alteon either through a data or management port.

TACACS+ offers the following advantages over RADIUS:

- TACACS+ uses TCP-based, connection-oriented transport, while RADIUS is UDP-based. TCP offers a connection-oriented transport, while UDP offers best-effort delivery. RADIUS requires additional programmable variables such as re-transmit attempts and timeouts to compensate for best-effort transport, but it lacks the level of built-in support that a TCP transport offers.
- TACACS+ offers full packet encryption, while RADIUS offers password-only encryption in authentication requests.
- TACACS+ separates authentication, authorization, and accounting.
- TACACS+ offers privilege level mapping. By enabling **cmap**, the privilege level can be increased from default 0-9 to 0-22.
- Alteon sends command log messages to the TACACS+ server when **clog** is enabled.

How TACACS+ Authentication Works

TACACS+ works much in the same way as RADIUS authentication, as described on How RADIUS Authentication Works, page 63:

1. The remote administrator connects to Alteon and provides the user name and password.
2. Using the authentication or authorization protocol, Alteon sends the request to the authentication server.
3. The authentication server checks the request against the user ID database.
4. Using the TACACS+ protocol, the authentication server instructs Alteon to grant or deny administrative access.

TACACS+ uses the AAA architecture, which separates authentication, authorization, and accounting. This allows separate authentication solutions that can still use TACACS+ for authorization and accounting. For example, with TACACS+, it is possible to use Kerberos authentication and TACACS+ authorization and accounting. After Alteon authenticates a user on a Kerberos server, it requests authorization information from a TACACS+ server without requiring re-authentication. Alteon informs the TACACS+ server that it has successfully authenticated the user on a Kerberos server, and the server then provides authorization information.

During a session, if additional authorization checking is needed, Alteon checks with a TACACS+ server to determine if the user is granted permission to use a particular command.

TACACS+ Authentication Features

Authentication is the action of determining the identity of a user, and is generally done when the user first attempts to log into Alteon or gain access to its services. Alteon supports ASCII inbound logins.

The following are not supported:

- PAP, CHAP, and ARAP login methods
- TACACS+ change password requests
- One-time password authentication

Authorization

Authorization is the action of determining a user's privileges on Alteon, and usually takes place after authentication.

The mapping between TACACS+ authorization levels and Alteon management access levels is described in Accounting, page 69.

Table 5 displays TACACS+ levels with disabled privilege level mapping (`/cfg/sys/tacacs/cmap/dis`):

Table 5: Alteon-Proprietary with Disabled Privilege Level Mapping for TACACS+

Alteon User Access Level	TACACS+ level
user	0
slboper	1
l4oper	2
oper	3
slbadmin	4
l4admin	5
admin	6
slbview	7
crtadmin	7
slbadmin + crtmng	8
l4admin + crtmng	9
l1oper	10
l2oper	11
l3oper	12
l3admin	13

Table 6 displays TACACS+ levels with enabled privilege level mapping (`/cfg/sys/tacacs/cmap/ena`):

Table 6: Alteon-Proprietary with Enabled Privilege Level Mapping for TACACS+

Alteon User Access Level	TACACS+ level
user	0, 1
slboper	2, 3
l4oper	4, 5
oper	6, 7, 8
slbadmin	9, 10, 11
l4admin	12, 13
admin	14, 15
slbview	16, 17
crtadmin	16, 17
slbadmin + crtmng	18, 19, 20
l4admin + crtmng	21, 22
l1oper	23

Table 6: Alteon-Proprietary with Enabled Privilege Level Mapping for TACACS+

Alteon User Access Level	TACACS+ level
l2oper	24
l3oper	25
l3admin	26

Accounting

Accounting is the act of recording a user's activities on Alteon for the purposes of billing and/or security. It follows the authentication and authorization actions. If the authentication and authorization actions are not performed through TACACS+, no TACACS+ accounting messages are sent out.

Whenever a command successfully executes, TACACS+ creates an accounting message and sends it to the TACACS+ server.

The attributes provided for the TACACS+ accounting are:

- protocol (console, Telnet, SSH, HTTP)
- start time (in seconds)
- stop time (in seconds)
- elapsed time (in seconds)
- disc cause (a string)

Note: Other than these attributes, the **cmd** and **cmd-arg** accounting attributes are also supported for command logging.

Configuring TACACS+ Authentication

To configure TACACS+ authentication

1. Turn TACACS+ authentication on, then configure the primary and secondary TACACS+ servers. You can configure IPv4 or IPv6 addresses for TACACS servers.

```
>> Main# /cfg/sys/tacacs                    (Select the TACACS+ Server menu)
>> TACACS+ Server# on                       (Turn TACACS+ on)
Current status: OFF
New status:     ON
>> TACACS+ Server# prisrv 10.10.1.1         (Enter the primary server IP)
Current primary TACACS+ server:     0.0.0.0
New pending primary TACACS+ server: 10.10.1.1
>> TACACS+ Server# secsrv 10.10.1.2         (Enter the secondary server IP)
Current secondary TACACS+ server:     0.0.0.0
New pending secondary TACACS+ server: 10.10.1.2
```

2. Configure the TACACS+ secret.

```
>> TACACS+ Server# secret
Enter new TACACS+ secret: <1-32 character secret>
```

Caution: If you configure the TACACS+ secret using any method other than a direct console connection, the secret may be transmitted over the network as clear text.

3. Optionally, you can change the default TCP port number used to listen to TACACS+.

   The well-known port for TACACS+ is 49.

```
>> TACACS+ Server# port Current TACACS+ port: 49
Enter new TACACS+ port [1-65000]: <port number>
```

4. Configure the number of retry attempts for contacting the TACACS+ server, and the timeout period.

```
>>TACACS+ Server# retries
Current TACACS+ server retries: 3
Enter new TACACS+ server retries [1-3]:         (Server retries)
>> TACACS+ Server# time
Current TACACS+ server timeout: 4
Enter new TACACS+ server timeout [1-15]: 10     (Enter the timeout period in minutes)
```

5. **Apply** and **save** the configuration.

Secure Shell and Secure Copy

The Telnet method for managing Alteon does not provide a secure connection. Secure Shell (SSH) and Secure Copy (SCP), however, use secure tunnels so that messages between a remote administrator and Alteon is encrypted and secured.

SSH is a protocol that enables remote administrators to log securely into another computer over a network to execute management commands.

SCP is typically used to copy files securely from one computer to another. SCP uses SSH for encryption of data on the network. Alteon uses SCP to download and upload the Alteon configuration via secure channels.

The Alteon implementation of SSH supports both versions 1.5 and 2.0, and supports SSH clients version 1.5 to 2.*x*. The following SSH clients have been tested:

- SSH 1.2.23 and SSH 1.2.27 for Linux (freeware)
- SecureCRT 3.0.2 and SecureCRT 3.0.3 for Windows NT (Van Dyke Technologies, Inc.)
- F-Secure SSH 1.1 for Windows (Data Fellows)
- Putty SSH
- Cygwin OpenSSH
- Mac X OpenSSH
- Solaris 8 OpenSSH
- AxeSSH SSHPro

- SSH Communications Vandyke SSH A
- F-Secure

Note: There can be a maximum number of four simultaneous Telnet, SSH, SCP connections at one time. The **/cfg/sys/radius/telnet** command also applies to SSH/SCP connections.

Configuring SSH and SCP Features

You can configure SSH and SCP parameters via the console port only, using the CLI. However, SCP **putcfg** and TFTP **getcfg** can also change the SSH and SCP configurations. When you enable SSH, SCP is also enabled. The Alteon SSH daemon uses TCP port 22 only and is not configurable.

Before you can use SSH commands, you must turn on SSH and SCP.

To enable or disable SSH

1. To enable SSH:

```
>> Main# /cfg/sys/access/sshd/on
Current status: OFF
New status: ON
```

2. To disable SSH:

```
>> Main# /cfg/sys/access/sshd/off
Current status: ON
New status: OFF
```

To enable or disable SCP putcg_apply **and** putcg_apply_save

1. To enable SCP **putcfg_apply** and **putfg_apply_save**:

```
>> # /cfg/sys/access/sshd/ena             (Enable SCP apply and save)
SSH Server# apply                         (Apply the changes to start generating RSA host
                                          and server keys)
RSA host key generation starts
..............................................................................
.....................................
RSA host key generation completes (lasts 212549 ms)
RSA host key is being saved to Flash ROM, please don't reboot
the box immediately.RSA server key generation starts
............................................................
RSA server key generation completes (lasts 75503 ms)
RSA server key is being saved to Flash ROM, please don't reboot
the box immediately.
------------------------------------------------------------------
Apply complete; don't forget to "save" updated configuration.
```

2. To disable SCP **putcg_apply** and **putcg_apply_save**:

```
>> Main# /cfg/sys/access/sshd/dis
```

Configuring the SCP Administrator Password

To configure the SCP Administrator (scpadmin**) password**

1. Connect to Alteon via the RS-232 management console. For security reasons, the scpadmin password may only be configured when connected directly to the console.
2. Enter the following commands:

   Note: The factory default setting for the SCP administrator password is **admin**.

```
>> /cfg/sys/access/sshd/scpadm
Changing SCP-only Administrator password; validation required...
Enter current administrator password: <password>
Enter new SCP-only administrator password: <new password>
Re-enter new SCP-only administrator password: <new password>
New SCP-only administrator password accepted.
```

SCP Services

To perform SCP commands, you need the SCP admin password with administrator privileges (this password must be different from the admin password).

The following SCP commands are supported in this service. These commands are entered using the CLI on the client that is running the SCP application:

- **getcfg**—Used to download the configuration to the remote host via SCP.
- **putcfg**—Used to upload the configuration from a remote host to Alteon. The **diff** command is executed at the end of **putcfg** to notify the remote client of the difference between the new and the current configurations.
- **putcfg_apply**—Runs the **apply** command after the putcfg is done.
- **putcfg_apply_save**—Saves the new configuration to the flash after putcfg_apply is done.

Note: The **putcfg_apply** and **putcfg_apply_save** commands are provided because additional apply and save commands are usually required after a **putcfg**, and an SCP session is not run in an interactive mode.

Using SSH and SCP Client Commands

This section includes the syntax and examples for some client commands. The examples use 192.168.249.13 as the IP address of a sample Alteon.

Logging into Alteon

The following is the syntax for logging into Alteon:

```
ssh <Alteon IP address> or ssh -l <login-name> <Alteon IP address>
```

Example Logging into Alteon

```
>> # ssh 192.168.249.13
>> # ssh -l <login-name> 192.168.249.13            (Log into Alteon)
```

Downloading the Configuration Using SCP

The following is the syntax for downloading the configuration using SCP:

```
>> # scp <Alteon IP address> :getcfg <local filename>
```

Example Downloading Alteon Configuration Using SCP

```
>> # scp 192.168.249.13:getcfg appldevice.cfg
```

Uploading the Configuration to Alteon

The following is the syntax for uploading the configuration to Alteon:

```
scp <local filename> <Alteon IP address> :putcfg
```

Example Uploading the Configuration to Alteon

```
>> # scp appldevice.cfg 192.168.249.13:putcfg
```

The **apply** and **save** commands are still needed after the last command (**`scp appldevice.cfg 192.168.249.13:putcfg`**). Alternately, you can use the following commands:

```
>># scp appldevice.cfg 192.168.249.13:putcfg_apply
>># scp appldevice.cfg 192.168.249.13:putcfg_apply_save
```

Notes

- The **diff** command is executed at the end of **putcfg** to notify the remote client of the difference between the new and the current configurations.
- **putcfg_apply** runs the **apply** command after the **putcfg** command.
- **putcfg_apply_save** saves the new configuration to the flash after the **putcfg_apply** command.

SSH and SCP Encryption of Management Messages

Table 7 shows the encryption and authentication methods that are supported for SSH and SCP:

Table 7: SSH and SCP Encryption of Management Messages

Encryption/Authentication	Method
Server host authentication	The client RSA authenticates Alteon at the beginning of every connection.
Key exchange	RSA
Encryption	3DES-CBC, DES
User authentication	Local password authentication, RADIUS, **SecurID** via RADIUS, for SSH only. It does not apply to SCP.

Generating RSA Host and Server Keys for SSH Access

To support the SSH server feature, two sets of RSA keys (host and server keys) are required. The host key is 1024 bits and is used to identify Alteon. The server key is 768 bits and is used to make it impossible to decipher a captured session by breaking into Alteon at a later time.

When you first enable and apply the SSH server, Alteon generates the RSA host and server keys and is stored in the flash memory.

To configure RSA host and server keys

1. Connect to Alteon via the console port (the commands for this procedure are not available via Telnet connection).
2. Enter the following commands to generate the keys manually:

```
>> # /cfg/sys/access/sshd/hkeygen          (Generates the host key)
>> # /cfg/sys/access/sshd/skeygen          (Generates the server key)
```

These two commands take effect immediately without the need of an apply command.

When Alteon reboots, it retrieves the host and server keys from the flash memory. If these two keys are not available in the flash memory and if the SSH server feature is enabled, Alteon generates them during the system reboot. This process may take several minutes to complete.

To set the interval of RSA server key auto-generation

Alteon can also regenerate the RSA server key, using the following command:

```
>> # /cfg/sys/access/sshd/intrval <number of hours (0-24)>
```

Note: This command is available when connected through either the console port, Telnet, or SSH.

The number of hours must be between 0 and 24. 0 indicates that RSA server key auto-generation is disabled. When greater than 0, Alteon auto-generates the RSA server key every specified interval. However, RSA server key generation is skipped if Alteon is busy with other key or cipher generation when the timer expires.

Note: Alteon performs only one key/cipher generation session at a time. As a result, an SSH/SCP client cannot log in if Alteon is performing key generation at the same time, or if another client has just logged in. Also, key generation fails if an SSH/SCP client is logging in at the same time.

SSH/SCP Integration with RADIUS Authentication

SSH/SCP is integrated with RADIUS authentication. After you enable the RADIUS server, Alteon redirects all subsequent SSH authentication requests to the specified RADIUS servers for authentication. This redirection is transparent to the SSH clients.

SSH/SCP Integration With SecurID

SSH/SCP can also work with SecurID, a token card-based authentication method. Using SecurID requires the interactive mode during login, which is not provided by the SSH connection.

Note: There is no SNMP or BBI support for SecurID because the SecurID server, ACE, is a one-time password authentication and requires an interactive session.

Using SecurID with SSH

Using SecurID with SSH includes the following tasks:

1. To log in using SSH, use a special username, "ace", to bypass the SSH authentication.
2. After an SSH connection is established, you are prompted to enter the username and password, after which the SecurID authentication is performed.
3. Provide your username and the token in your SecurID card as a regular Telnet user.

Using SecurID with SCP

Using SecurID with SCP can be performed in one of the following ways:

- **Using a RADIUS server to store an administrator password**—You can configure a regular administrator with a fixed password in the RADIUS server if it can be supported. A regular administrator with a fixed password in the RADIUS server can perform both SSH and SCP with no additional authentication required.

- **Using an SCP-only administrator password**—Use the command **`/cfg/sys/access/sshd/scpadm`** to bypass the checking of SecurID.

  Note: The **`/cfg/sys/access/sshd/scpadmin`** command is only available when connected through the console port for the Global Administrator, and Telnet for the vADC Administrator.

  An SCP-only administrator's password is typically used when SecurID is used. For example, it can be used in an automation program (in which the tokens of SecurID are not available) to back up (download) the configurations each day.

  Note: The SCP-only administrator password must be different from the regular administrator password. If the two passwords are the same, the administrator using that password is not allowed to log in as an SSH user because Alteon recognizes him as the SCP-only administrator, and only allows the administrator access to SCP commands.

  Alternately, you can configure a regular administrator with a fixed password in the RADIUS server if it can be supported. A regular administrator with a fixed password in the RADIUS server can perform both SSH and SCP with no additional authentication required.

End User Access Control

Alteon allows an administrator to define end user accounts that permit end users to operationally act on their own real servers via the CLI commands. Once end user accounts are configured and enabled, Alteon requires username and password authentication.

For example, an administrator can assign a user to manage real servers 1 and 2 only. The user can then log into Alteon and perform operational commands (effective only until the next reboot), to enable or disable the real servers, or change passwords on the real servers.

Considerations for Configuring End User Accounts

- Only one user ID can be assigned to a real server resource to enable or disable a real server. Consequently, a single end user may be assigned the maximum number of real servers that can be configured, to the exclusion of any other users.
- A maximum of 10 user IDs are supported.
- The administrator must ensure that all real and backup servers or groups belonging to a virtual service are owned by the same end-user ID. Alteon does not validate configurations. The criterion for displaying virtual service information for end users is based on the validation of ownership of the first real server in the group for a given virtual server port.
- Alteon has end-user support for console and Telnet access. As a result, only very limited access is granted to the primary administrator under the BBI/SSH1 mode of access.
- RADIUS authentication and user passwords cannot be used concurrently to access Alteon.
- Passwords can be up to 128 characters for TACACS, RADIUS, Telnet, SSH, console, and Web access.

User Access Control Menu

The *End User Access Control* menu is located in the *System Access* menu:

```
>> # /cfg/sys/access/user
```

Setting up User IDs

To set up a user ID

You can configure up to 10 user IDs. For example:

```
/cfg/sys/access/user/uid 1
```

Defining User Names and Passwords

To define user names and passwords

The following is an example for defining a user name and password:

```
>> User ID 1 # name jane
Current user name:
New user name:     jane
```

Changing Passwords

To change passwords

The following is an example for changing passwords:

```
>> User ID 1 # pswd
Changing user password; validation required:
Enter current admin password: <current administrator password>
Enter new user password: <new user password>
Re-enter new user password: <new user password>
New user password accepted.
```

Defining User Access Level

By default, the end user is assigned to the user access level (also known as class of service, or CoS). The CoS for all user accounts has global access to all resources except for User CoS, which has access to view resources that only the user owns. For more information, see Table 3 - Alteon User Accounts and Access Levels, page 64.

To change the user's level

Enter the class of service **cos** command, and select one of the following options:

```
>> User ID 1 # cos <user|slboper|l4oper|oper|slbadmin|l4admin|admin>
```

Assigning One or More Real Servers to the End User

A single end user may be assigned up to 1023 real servers. Once assigned, the real server cannot be assigned to any other user.

```
>> User ID 1 # add
Enter real server number: (1-1023) 23
```

Validating User Configuration

The following is an example of a currently defined user configuration:

```
User ID 2 # cur
  name jane    , dis, cos user    , password valid, offline
  real servers:
   23: 0.0.0.0, disabled, name , weight 1,
timeout 20 mins, max-
con 200000
   24: 0.0.0.0, disabled, name , weight 1,
timeout 20 mins, max-
con 200000
```

Listing Current Users

The **cur** command displays defined user accounts and whether each user is currently logged into Alteon:

```
# /cfg/sys/access/user/cur

Usernames:
  user     - Enabled
  slbview  - Disabled
  slboper  - Disabled
  l4oper   - Disabled
  oper     - Disabled
  l3admin  - Disabled
  slbadmin - Disabled
  l4admin  - Disabled
  admin    - Always Enabled

Current User ID table:
 1: name jane    , ena, cos user    , password valid, online
    real servers:
    1: 10.10.10.211, disabled, name , weight 1, timeout 10 mins,
maxcon 200000
     2: 10.10.10.212, enabled, name , weight 1, timeout 10 mins,
maxcon 200000
 2: name john    , ena, cos user    , password valid, online
    real servers:
     3: 10.10.10.213, enabled, name , weight 1, timeout 10 mins,
maxcon 200000
```

Enabling or Disabling a User

You must enable an end-user account before Alteon recognizes and permits login under the account. Once enabled, Alteon requires any user to enter both a username and password.

```
>> # /cfg/sys/access/user/uid <#> /ena
>> # /cfg/sys/access/user/uid <#> /dis
```

Logging into an End User Account

After you have configured and enabled an end-user account, the user can log into Alteon with a username and password combination. The CoS established for the end user account determines the level of access.

Disabling a User Account

The User account is enabled by default on Alteon and ADC-VX hypervisors. To disable a user account, set the user password to empty.

To disable a user account

The following is an example for disabling user accounts:

```
>> # /cfg/sys/access/user/usrpw
Changing USER password; validation required:
Enter current admin password:
Enter new user password:
Re-enter new user password:

"user" disabled with empty password. New user password accepted.
```

Deny Routes

A deny route, or black hole route, can be configured to deny Layer 3 routable packets to destinations covered by a static route. A deny route is created by setting the gateway address in a static route to 0. If the longest prefix match route (which is obtained via route lookup) is a deny route, the packet is dropped.

A deny route may be configured when an administrator discovers a specific user or network under attack. This feature is similar to a deny filter, except that it works only on routable Layer 3 traffic. It does not deny Layer 2 traffic.

Configuring a Deny Route

In this example, IP addresses in the network 62.62.0.0 are under attack from an unknown source. You temporarily configure Alteon with a deny route so that any traffic destined to this network is dropped. In the meantime, the attack pattern and source can be detected.

To deny traffic to the destination network 62.62.0.0

`>> # /cfg/l3/route`	(Select the IP Static Route menu)
`>> IP Static Route# add`	(Add a static route)
`Enter destination IP address:   62.62.0.0`	(Of this IP network address)
`Enter destination subnet mask: 255.255.0.0 (And this mask address)` `Enter gateway IP address (for martian/deny route use 0):0`	
	(Enter 0 to create a deny route)
`Enter interface number: (1-256)`	(A deny route will ignore an Inter face number, so don't enter one here.)

Caution: Do not configure a deny route that covers the destination/mask pair of an existing IP interface's IP address/mask pair. For example, if you have an IP interface of 50.0.0.1/255.0.0.0, and a deny route of 50.0.0.0/255.0.0, then traffic to the interface as well as the subnet is denied, which is *not* the desired result.

Viewing a Deny Route

The following is an example view, or dump, of a deny route.

To view a deny route

Enter the `/info/l3/dump` command. A deny route appears in the routing table in bold.

Status code: * - best

Destination	Mask	Gateway	Type	Tag	Metr	If
* 0.0.0.0	0.0.0.0	47.80.16.1	indirect	static	47	
* 52.80.16.0	255.255.254.0	47.80.16.59	direct	fixed	47	
* 52.80.16.59	255.255.255.25	47.80.16.59	local	addr	47	
*** 62.62.0.0**	**255.255.0.0**	**0.0.0.0**	**deny**	**static**	**47**	

Chapter 4 – VLANs

This chapter describes network design and topology considerations for using Virtual Local Area Networks (VLANs). VLANs are commonly used to split groups of network users into manageable broadcast domains to create logical segmentation of workgroups, and to enforce security policies among logical segments.

The following topics are addressed in this chapter:

- VLAN ID Numbers, page 81—This section discusses VLANs with VLAN ID numbers.
- VLAN Tagging, page 81—This section discusses VLAN tagging.
- VLANs and the IP Interfaces, page 82—This section briefly describes how management functions can only be accomplished from stations on VLANs that include an IP interface to Alteon.
- VLAN Topologies and Design Issues, page 82—This section discusses how you can logically connect users and segments to a host that supports many logical segments or subnets by using the flexibility of the multiple VLAN system.
- VLANs and Default Gateways, page 85—This section discusses associating gateways to VLANs.

Notes

- Basic VLANs can be configured during initial configuration. For more information, see *Using the Setup Utility* in the *Alteon Application Switch Operating System Command Reference*.
- More comprehensive VLAN configuration can be done from the CLI. For more information, see *VLAN Configuration*, as well as *Port Configuration*, in the *Alteon Application Switch Operating System Command Reference*.

VLAN ID Numbers

Alteon supports up to 2048 VLANs per Alteon. Even though the maximum number of VLANs supported at any given time is 2048, each can be identified with any number between 1 and 4090.

VLANs are defined on a per-port basis. Each port on Alteon can belong to one or more VLANs, and each VLAN can have any number of ports in its membership. Any port that belongs to multiple VLANs, however, must have VLAN tagging enabled (see VLAN Tagging, page 81).

Each port has a configurable default VLAN number, known as its **PVID**. The factory default value for all PVIDs is 1. This places all ports on the same VLAN initially, although each PVID is configurable to any VLAN number between 1 and 4090.

Any untagged frames (those with no VLAN specified) are classified with the sending port's PVID.

VLAN Tagging

Alteon supports 802.1Q VLAN tagging, providing standards-based VLAN support for Ethernet systems.

Tagging places the VLAN identifier in the frame header, allowing multiple VLANs per port. When you configure multiple VLANs on a port, you must also enable tagging on that port.

Because tagging fundamentally changes the format of frames transmitted on a tagged port, you must carefully plan the design of a network to prevent transmission of tagged frames to devices that do not support 802.1Q VLAN tags.

VLANs and the IP Interfaces

You can access Alteon for remote configuration, trap messages, and other management functions only from stations on VLANs that include an IP interface to Alteon. For more information, see the *IP Interface Menu* section in the *Alteon Application Switch Operating System Command Reference*. Likewise, you can cut off access to management functions to any VLAN by excluding IP interfaces from the VLAN membership.

Note: Carefully consider how you create VLANs so that communication with Alteon remains possible.

For example, if all IP interfaces are left on VLAN 1 (the default), and all ports are configured for VLANs other than VLAN 1, then management features are effectively cut off. If an IP interface is added to one of the other VLANs, the stations in that VLAN will all have access to management features.

VLAN Topologies and Design Issues

By default, Alteon has a single VLAN configured on every port. This configuration groups all ports into the same broadcast domain. The VLAN has an 802.1Q VLAN PVID of 1. VLAN tagging is turned off, because by default only a single VLAN is configured per port.

Since VLANs are most commonly used to create individual broadcast domains and/or separate IP subnets, host systems should be present on more than one VLAN simultaneously. Alteon and VLAN-tagging server adapters support multiple VLANS on a per-port or per-interface basis, allowing very flexible configurations.

You can configure multiple VLANs on a single VLAN-tagging server adapter, with each VLAN being configured through a logical interface and logical IP address on the host system. Each VLAN configured on the server adapter must also be configured on the port to which it is connected. If multiple VLANs are configured on the port, tagging must be turned on.

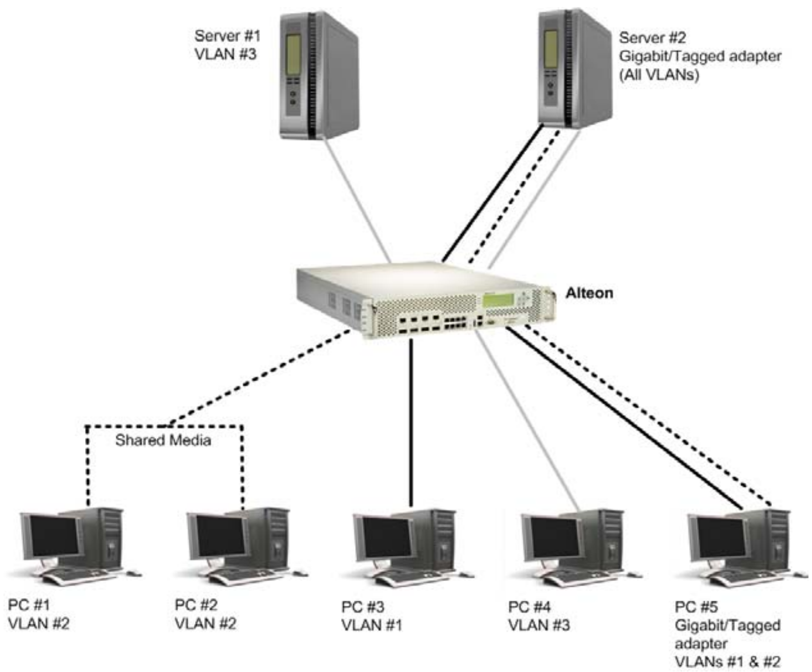
Using this flexible multiple VLAN system, you can logically connect users and segments to a host with a single VLAN-tagging adapter that supports many logical segments or subnets.

If a 802.1Q tagged frame is sent to a port that has VLAN-tagging disabled, then the frames are dropped at the ingress port.

Examples

A Multiple VLANS with Tagging Adapters

Figure 2: Multiple VLANs with Tagging Adapters Example



The components of this example VLAN configuration are described in Table 8:

Table 8: Explanation of Example of Multiple VLANs with Tagging Adapters

Component	Description
Alteon	This Alteon is configured for three VLANs that represent three different IP subnets. Two servers and five clients are attached to Alteon.
Server #1	This server is part of VLAN 3 and is present in only one IP subnet. The port that the VLAN is attached to is configured only for VLAN 3, so VLAN tagging is off.
Server #2	This high-use server needs to be accessed from all VLANs and IP subnets. The server has a VLAN-tagging adapter installed with VLAN tagging turned on. The adapter is attached to one of Alteon's Gigabit Ethernet ports that is configured for VLANs 1, 2, and 3. Tagging is turned on. Because of the VLAN tagging capabilities of both the adapter and Alteon, the server is able to communicate on all three IP subnets in this network. Broadcast separation between all three VLANs and subnets, however, is maintained.
PCs #1 and #2	These PCs are attached to a shared media hub that is then connected to Alteon. They belong to VLAN 2 and are logically in the same IP subnet as Server 2 and PC 5. Tagging is not enabled on their ports.
PC #3	A member of VLAN 1, this PC can minimize its broadcast domain to Server 2 and PC 5.

Table 8: Explanation of Example of Multiple VLANs with Tagging Adapters

Component	Description
PC #4	A member of VLAN 3, this PC can minimize its broadcast domain to Server 1 and Server 2.
PC #5	A member of both VLAN 1 and VLAN 2, this PC has VLAN-tagging Gigabit Ethernet adapter installed. It can minimize its broadcast domain to Server #2 via VLAN 1, and to PC #1 and PC #2 via VLAN 2. The port to which it is connected is configured for both VLAN 1 and VLAN 2 and has tagging enabled.

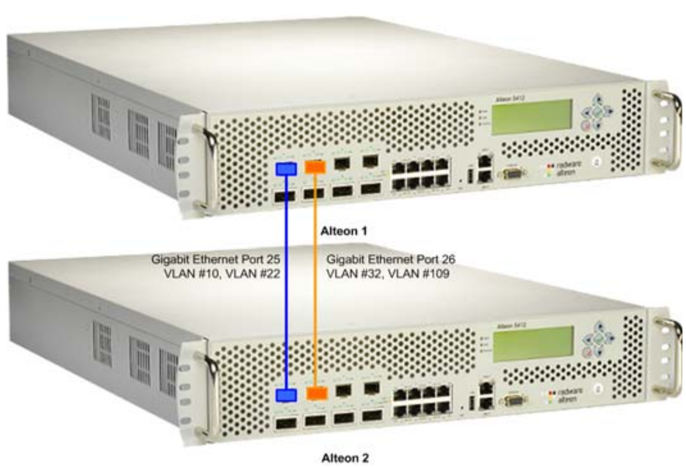
Note: VLAN tagging is required only on ports that are connected to other Alteons or on ports that connect to tag-capable end-stations, such as servers with VLAN- tagging adapters.

B Parallel Links with VLANs

This example shows how it is possible through the use of VLANs to create configurations where there are multiple links between two Alteons, without creating broadcast loops.

In Figure 3 - Parallel Links with VLANs Example, page 84, two Alteons are connected with two different Gigabit Ethernet links. Without VLANs, this configuration would create a broadcast loop. To prevent broadcast loops, port 25 is on VLAN 10 and port 26 is on VLAN 109. Both Alteon-to-Alteon links are on different VLANs and therefore are separated into their own broadcast domains.

Figure 3: Parallel Links with VLANs Example



Note: In this example, the Gig ports are on different VLANs and the Spanning Tree Protocol (STP) is disabled. For information on STP, see Spanning Tree Protocol, page 99.

VLANs and Default Gateways

Alteon lets you assign different gateways for each VLAN. You can effectively map multiple customers to specific gateways on a single Alteon. The benefits of segregating customers to different default gateways are:

- Resource optimization
- Enhanced customer segmentation
- Improved service differentiation

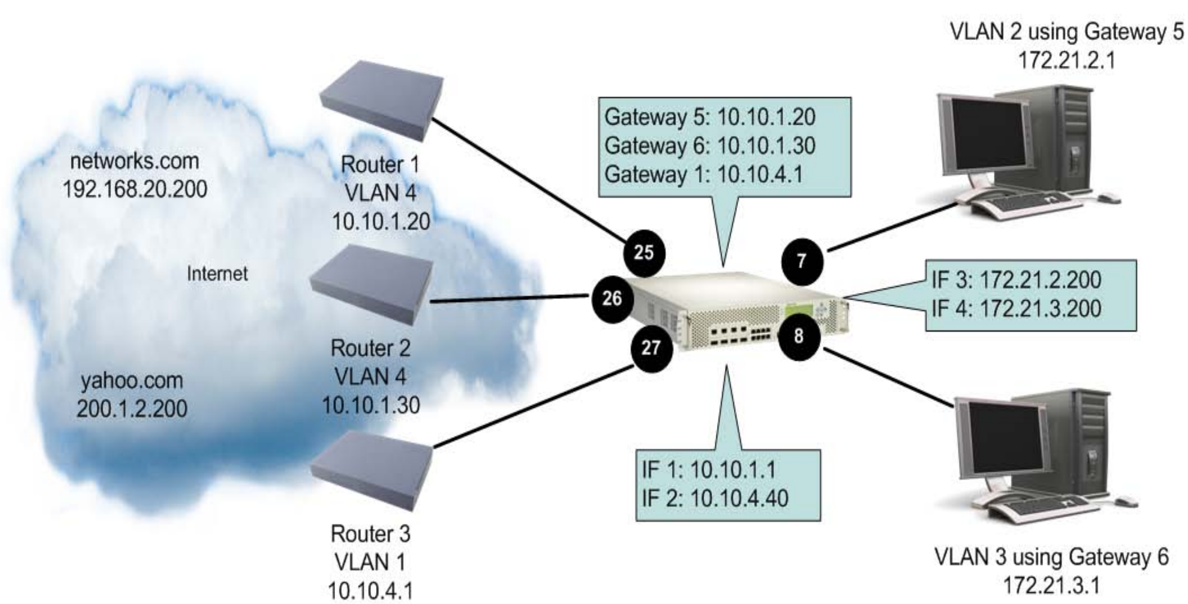
Segregating VLAN Traffic

Deploy this feature in an environment where you want to segregate VLAN traffic to a configured default gateway.

Example Segregation of VLAN Traffic

Figure 4 - Example Segregation of VLAN Traffic Configuration, page 85 illustrates a configuration where VLANs 2 and 3 have different routing requirements. VLAN 2 is required to route traffic through default gateway 5 and VLAN 3 is required to route traffic through default gateway 6.

Figure 4: Example Segregation of VLAN Traffic Configuration



You can configure up to 255 gateways with one gateway per VLAN with values starting from 5 through 259. If the gateways per VLAN fail, then traffic is directed to default gateways 1 through 4. Default gateways 1 through 4 are used for load balancing session requests and as backup when a specific gateway that has been assigned to a VLAN is down.

If gateways 5 or 6 fail, then traffic is directed to default gateway 1, which is configured with IP address 10.10.4.1. If default gateways 1 through 4 are not configured, then packets from VLAN 2 and VLAN 3 are discarded.

The route cache table records each session request by mapping the destination IP address with the MAC address of the default gateway. The command **/info/l3/arp/dump** displays the entries in the route cache similar to Table 9. The destination IP addresses (see the last two rows of the table) are associated with the MAC addresses of the gateways.

Table 9: Route Cache Table in the Example

Destination IP address	Flags	MAC Address	VLAN	Port	Shared	Referenced SPs
10.10.1.1	P	00:60:cf:46:48:60	4		ENA	1-4
10.10.1.20		00:60:cf:44:cd:a0	4	25 (Gig)	ENA	empty
10.10.1.30		00:60:cf:42:3b:40	4	26 (Gig)	ENA	empty
10.10.4.1		00:60:cf:42:77:e0	1	27 (Gig)	DIS	empty
10.10.4.40	P	00:60:cf:46:48:60	1		DIS	1-4
172.21.2.27		00:50:da:17:c8:05	2	7	DIS	1
172.21.2.200	P	00:60:cf:46:48:60	2		DIS	1-4
172.21.3.14		00:c0:4f:09:3e:56	3	8	DIS	2
172.21.2.200	P	00:60:cf:46:48:60	3		DIS	1-4
192.168.20.200	R	00:60:cf:44:cd:a0	4	1	DIS	7
200.1.2.200	R	00:60:cf:42:3b:40	4	2	DIS	8

Traffic from VLAN 2 uses Gateway 5 to access destination IP address 192.168.20.200. If traffic from VLAN 3 requests the same destination address, then traffic is routed via Gateway 5 instead of Gateway 6, because 192.168.20.200 in the route cache is mapped to Gateway 5. If the requested route is not in the route cache, then Alteon reads the routing table. If the requested route is not in the routing table, then Alteon looks at the configured default Gateway.

Example VLAN-Based Gateway

VLAN-based gateways do not apply to client-based traffic. Rather, defining a VLAN-based gateway configures Alteon to use a predetermined gateway for the real server response.

The following configuration has three VLANs:

```
VLAN    Name                     Status   BWC    Learn   Ports
----    ------------------------  ------  ---    ------------
1       Default VLAN             ena      256     1      3 5 7-23 25-28
2       VLAN 2                   ena      256     2      4
3       VLAN 3                   ena      256     6      24
```

The real servers reside on VLAN 1. By specifying a VLAN-based gateway, Alteon controls which external link these real servers will use to respond to client requests. The external link used is not dependent on whether the client traffic was sourced from VLAN 2 or VLAN 3.

Configuring the Local Network

To completely segregate VLAN traffic to its own default gateway, you can configure the local network addresses of the VLAN. As shown in Example Segregation of VLAN Traffic, page 85, this ensures that all traffic from VLAN 2 is forwarded to Gateway 5 and all traffic from VLAN 3 is forwarded to Gateway 6.

Typically, Alteon routes traffic based on the routes in the routing table. The routing table contains an entry of the configured local network with the default gateway. The route cache will not contain the route entry. This configuration provides a more secure environment, but affects performance if the routing table is close to its maximum capacity.

Configuring Gateways Per VLAN

The following is an example gateway configuration for a VLAN.

Example Gateway Configuration for a VLAN

1. Assign an IP address for each router and client workstation.
2. Assign an IP interface for each subnet attached to Alteon.

`>> /cfg/l3/if 1`	(Select IP interface 1 for gateway 5 and 6 subnet)
`>> IP Interface 1#  addr 10.10.1.1`	(Assign IP address for interface 1)
`>> IP Interface 1#  mask 255.255.255.0`	(Assign mask for IF 1)
`>> IP Interface 1#  vlan 4`	(Assign VLAN 4 to IF 1)
`>> IP Interface 1#  /cfg/l3/if 2`	(Select IP interface 2 for gateway 1)
`>> IP Interface 2#  addr 10.10.4.40`	(Assign IP address for interface 2)
`>> IP Interface 2#  mask 255.255.255.0`	(Assign mask for IF 2)
`>> IP Interface 2#  vlan 1`	(Assign VLAN 1 to IF 2)
`>> IP Interface 2#  /cfg/l3/if 3`	(Select IP interface 3 for VLAN 2 subnet)
`>> IP Interface 3#  addr 172.21.2.200`	(Assign IP address for interface 3)
`>> IP Interface 3#  mask 255.255.255.0`	(Assign mask for IF 3)
`>> IP Interface 3#  vlan 2`	(Assign VLAN 2 to IF 3)
`>> IP Interface 3#  /cfg/l3/if 4`	(Select IP interface 4 for VLAN 3) subnet)
`>> IP Interface 4#  addr 172.21.3.200`	(Assign IP address for interface 4)
`>> IP Interface 4#  mask 255.255.255.0`	(Assign mask for IF 4)
`>> IP Interface 4#  vlan 3`	(Assign VLAN 3 to IF 4)

3. Configure the default gateways. Configure gateways 5 and 6 for VLANs 2 and 3, respectively. Configure default gateway 1 for load-balancing session requests and as backup when gateways 5 and 6 fail.

`>> /cfg/l3/gw 5`	(Select gateway 5)
`>> Default gateway 5# addr 10.10.1.20`	(Assign IP address for gateway 5)
`>> Default gateway 5# /cfg/l3/gw 6`	(Select default gateway 6)
`>> Default gateway 6# addr 10.10.1.30`	(Assign IP address for gateway 6)
`>> Default gateway 6# /cfg/l3/gw 1`	(Select default gateway 1)
`>> Default gateway 1# addr 10.10.4.1`	(Assign IP address for gateway 1)

Note: The IP address for default gateways 1 to 4 must be unique. IP addresses for default gateways 5 to 259 can be set to the same IP address as the other gateways (including default gateway 1 to 4). For example, you can configure two default gateways with the same IP address for two different VLANs.

4. Add the VLANs to the gateways and enable them.

`>> /cfg/l3/gw 5`	(Select gateway 5)
`>> Default gateway 5# vlan 2`	(Add VLAN 2 for default gateway 5)
`>> Default gateway 5# ena`	(Enable gateway 5)
`>> Default gateway 5# /cfg/l3/gw 6`	(Select gateway 6)
`>> Default gateway 6# vlan 3`	(Add VLAN 3 for default gateway 6)
`>> Default gateway 6# ena`	(Enable gateway 6)
`>> Default gateway 6# /cfg/l3/gw 1`	(Select default gateway 1)
`>> Default gateway 1# ena`	(Enable gateway 1 for all VLAN s)

5. Apply and verify your configuration.

`>> Default gateway 1# /cfg/l3/cur`	(View current Layer 3 settings)

6. Optionally, configure the local networks using address and mask pairs to ensure that the VLANs use the configured default gateways.

`>> Default gateway 1# /cfg/l3/frwd/local`	(Select the local network menu)
`>> IP Forwarding# add 10.10.0.0 255.255.0.0`	(Specify the network for routers 1, 2, and 3)
`>> IP Forwarding# add 172.21.2.0 255.255.255.0`	(Specify the network for VLAN 2)
`>> IP Forwarding# add 172.21.3.0 255.255.255.0`	(Specify the network for VLAN 3)

7. Apply and save your new configuration changes.

```
>> IP Forwarding# apply
>> IP Forwarding# save
```

Chapter 5 – Port Trunking

Trunk groups can provide super-bandwidth, multi-link connections between Alteons or other trunk-capable devices. A trunk group is a group of ports that act together, combining their bandwidth to create a single, larger virtual link. This chapter provides configuration background and examples for trunking multiple ports together either in a static (manually configured) trunk group, or dynamic trunk group using Link Aggregation Control Protocol (LACP).

The following topics are addressed in this chapter:

Overview

When using port trunk groups between two Alteons, as shown in Figure 5 - Example Port Trunk Group Between Alteons, page 89, you can create a virtual link between Alteons operating up to 4 gigabits per second, depending on how many physical ports are combined. Alteon supports up to 12 static trunk groups per Alteon, each with two to eight ports per group.

Figure 5: Example Port Trunk Group Between Alteons

Trunk groups are also useful for connecting an Alteon to third-party devices that support link aggregation, such as Cisco routers and switches with EtherChannel® technology (not ISL trunking technology) and Sun's Quad Fast Ethernet Adapter. Trunk group technology is compatible with these devices when they are configured manually.

Statistical Load Distribution

Network traffic is statistically load balanced between the ports in a trunk group. Alteon uses both the Layer 2 MAC address and Layer 3 IP address information present in each transmitted frame for determining load distribution.

The addition of Layer 3 IP address examination is an important advance for traffic distribution in trunk groups. In some port trunking systems, only Layer 2 MAC addresses are considered in the distribution algorithm. Each packet's particular combination of source and destination MAC addresses results in selecting one line in the trunk group for data transmission. If there are enough Layer 2 devices feeding the trunk lines, then traffic distribution becomes relatively even. In some topologies, however, only a limited number of Layer 2 devices (such as a handful of routers and servers) feed the trunk lines. When this occurs, the limited number of MAC address combinations encountered results in lopsided traffic distribution, which can reduce the effective combined bandwidth of the trunked ports.

By adding Layer 3 IP address information to the distribution algorithm, a far wider variety of address combinations are seen. Even with just a few routers feeding the trunk, the normal source and destination IP address combinations (even within a single LAN) can be widely varied. This results in a wider statistical load distribution and maximizes the use of the combined bandwidth available to trunked ports.

The Trunk Hash Algorithm

In order to distribute the load across all active ports in a trunk group, the following algorithm is used to determine which port within the trunk group to use for frame forwarding, where *x* is the number of active ports within the trunk group:

```
hash_idx = (A xor B)
port = (lower 6 bits of hash_idx) mod x
```

The values of parameters A and B are defined below for the different types of forwarding and frames. These two parameters are XORed together to give the hash index. The modulus (mod) *x* of the lower 6 bits of the hash index is then taken to give the port of the trunk group.

Note: The same algorithm is used across all Alteons.

- For Layer 2 forwarding of non-IP frames:

  A = lower 16 bits of destination MAC address

  B = lower 32 bits of source MAC address
- For Layer 2 forwarding of IP frames:

  A = lower 16 bits of source IP address

  B = lower 32 bits of source MAC address
- For Layer 3 forwarding (enabled in WSM platform and Cheetah 20.1):

  A = lower 32 bits of destination IP

  B = lower 16 bits of source MAC
- For Layer 4 trunking (traffic towards the real servers in SLB and WCR):

  A = lower 32 bits of source IP

  B = lower 16 bits of destination MAC

Note: Layer 4 trunk hashing is currently supported only in Alteon 21.0 and higher.

Built-In Fault Tolerance

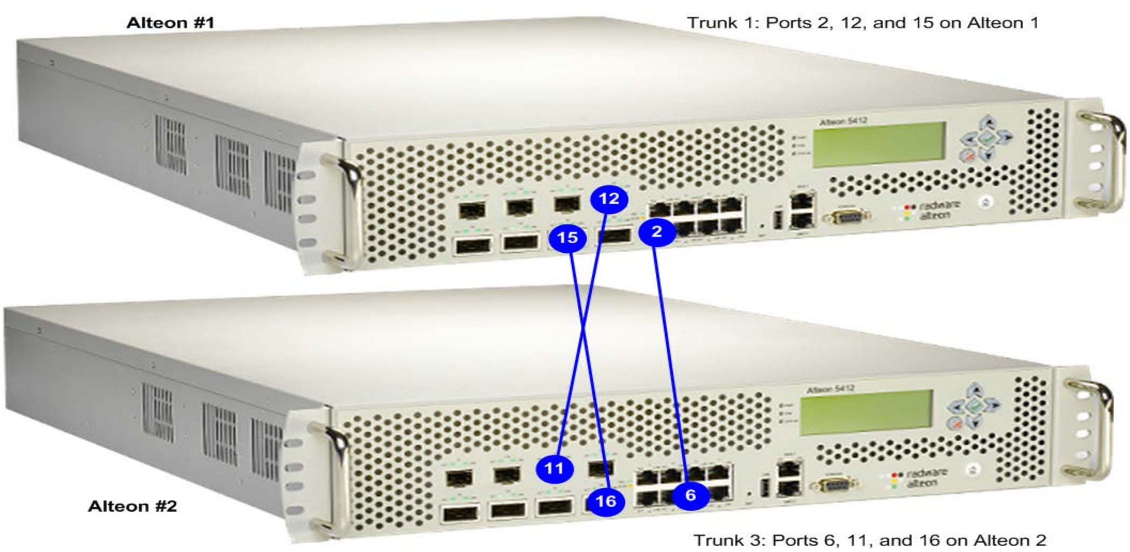
Since each trunk group comprises multiple physical links, the trunk group is inherently fault tolerant. As long as one connection between the Alteons is available, the trunk remains active.

Statistical load balancing is maintained whenever a port in a trunk group is lost or returned to service.

Example Static Port Trunking

In the following example, three ports are trunked between two Alteons:

Figure 6: Static Port Trunking Example



Prior to configuring each Alteon, you must connect to the appropriate CLI as the administrator.

Note: For details about accessing and using any of the menu commands described in this example, see the *Alteon Application Switch Operating System Command Reference*.

In this example, two Alteons are used. If a third-party device supporting link aggregation is used (such as Cisco routers and switches with EtherChannel technology or Sun's Quad Fast Ethernet Adapter), trunk groups on the third-party device should be configured manually. Connection problems could arise when using automatic trunk group negotiation on the third-party device.

Caution: To prevent spanning tree instability, do not change the spanning tree parameters on individual ports belonging to any trunk group.

1. Connect the ports that are involved in the trunk group.
2. On Alteon 1, define a trunk group.

`>> # /cfg/l2/trunk 1`	(Select trunk group 1)
`>> Trunk group 1# add 2`	(Add port 2 to trunk group 1)
`>> Trunk group 1# add 12`	(Add port 12 to trunk group 1)
`>> Trunk group 1# add 15`	(Add port 15 to trunk group 1)
`>> Trunk group 1# ena`	(Enable trunk group 1)

3. Apply and verify the configuration.

`>> Trunk group 1# apply`	(Make your changes active)
`>> Trunk group 1# cur`	(View current trunking configuration)

4. Examine the resulting information. If any settings are incorrect, make appropriate changes.
5. Save your new configuration changes.

```
>> Trunk group 1# save
```

6. Repeat the process on Alteon 2.

`>> # /cfg/l2/trunk 3`	(Select trunk group 3)
`>> Trunk group 3# add 6`	(Add port 6 to trunk group 3)
`>> Trunk group 3# add 11`	(Add port 11 to trunk group 3)
`>> Trunk group 3# add 16`	(Add port 16 to trunk group 3)
`>> Trunk group 3# ena`	(Enable trunk group 3)
`>> Trunk group 3# apply`	(Make your changes active)
`>> Trunk group 3# cur`	(View current trunking configuration)
`>> Trunk group 3# save`	(Save for restore after reboot)

Trunk group 1 (on Alteon 1) is now connected to trunk group 3 (on Alteon 2).

7. Examine the trunking information on each Alteon.

```
>> /info/l2/trunk
```

Information about each port in each configured trunk group is displayed. Make sure that trunk groups consist of the expected ports and that each port is in the expected state.

Notes

- Any physical port can belong to only one trunk group.
- Up to eight ports can belong to the same trunk group.
- Best performance is achieved when all ports in any given trunk group are configured for the same speed.
- Trunking from non-Alteon devices must comply with Cisco EtherChannel technology.

Link Aggregation Control Protocol Trunking

The Link Aggregation Control Protocol (LACP) is an IEEE 802.3ad standard for grouping several physical ports into one logical port (known as a trunk group or a link aggregation group) with any device that supports the standard. If a link in a LACP trunk group fails, traffic is reassigned dynamically to any of the remaining links of the LACP trunk group. Link aggregation is a method of grouping physical link segments of the same media type and speed in full duplex, and treating them as if they were part of a single, logical link segment. Refer to IEEE 802.3ad-2002 for a full description of the standard.

When using LACP, any trunk groups you may have already configured according to the manual procedure described in Static Port Trunking, page 91 are "static trunks." Any trunk groups using LACP are "dynamic trunks." With LACP, the maximum number of trunk groups has increased to 40. Static trunks continue to be limited to trunk IDs 1 through 12, and LACP trunks use IDs 13 through 40.

The Alteon implementation of LACP lets you group a maximum of eight physical ports into one logical port (LACP trunk group). Standby ports in LACP are created only when there are more than eight LACP ports configured in a trunk. Alteon assigns any non-trunked LACP-configured ports as standby ports for the LACP trunk. If any of the eight primary LACP ports fails, Alteon dynamically replaces it with the standby port.

Alteon can form trunk groups with any device which supports the IEEE 802.3ad standard.

Each LACP port has a parameter called **admin key**. An LACP trunk group is formed with the ports with the same admin key. The value of admin key can be any integer between 1 and 65535.

Example Actor Versus Partner LACP Configuration

Table 10: Actor Versus Partner LACP Configuration

Actor Device	Partner Device 1
Port 1 (admin key = 100)	Port 1 (admin key = 50)
Port 2 (admin key = 100)	Port 2 (admin key = 50)
Port 3 (admin key = 100)	Port 3 (admin key = 50)
Port 4 (admin key = 100)	Port 4 (admin key = 50)

In this example, actor device ports 1 through 4 can aggregate to form an LACP trunk group with the partner device ports 1 through 4. Note that the port admin key value has local significance only. The admin key value for the partner device ports can be any integer value but it should be same for all ports 1 through 4. In this example, it is 50.

LACP Modes

Each port can have one of the following LACP modes:

- **off (default)**—The user can configure this port into a regular static trunk group.
- **active**—The port is capable of forming an LACP trunk. This port sends LACP data unit (LACPDU) packets to partner system ports.
- **passive**—The port is capable of forming an LACP trunk. This port only responds to the LACPDU packets sent from an LACP **active** port.

Each active LACP port transmits LACPDUs, while each passive LACP port listens for LACPDUs. During LACP negotiation, the admin key value is exchanged. The LACP trunk group is enabled as long as the information matches at both ends of the link. If the admin key value changes for a port at either end of the link, that port's association with the LACP trunk group is lost.

When the system is initialized, all ports by default are in LACP off mode and are assigned unique admin keys. To make a group of ports eligible for aggregation, you assign all of them the same admin key. You must set the port's LACP mode to active to activate LACP negotiation. You can set another port's LACP mode to passive, to reduce the amount of LACPDU traffic, at the initial trunk-forming stage.

Note: LACP implementation does not support the Churn machine, an option used for detecting the port is operable within a bounded time period between the actor and the partner. Only the marker responder is implemented, and there is no marker protocol generator. Refer to 802.3ad-2002 for details.

Configuring LACP

Use the following procedure to configure LACP for port 1 through port 4 for the actor device to participate in link aggregation. Perform a similar configuration on the partner device with admin key 50.

1. Set the LACP mode on port 1.

`>> # /cfg/l2/lacp/port 1/mode`	(Select port 1 for LACP mode of operation)
`>> LACP port 1# active`	(Set port 1 to LACP active)
`Current Port 1 LACP mode setting: off` `New Port 1 LACP mode setting: active`	

2. Define the admin key on port 1. Only ports with the same admin key can form a LACP trunk group.

`>> # /cfg/l2/lacp/port 1/adminkey 100`	(Set port 1 adminkey to 100)
`Current LACP port adminkey:     1` `New pending LACP port adminkey: 100`	

3. Set the LACP mode on ports 2 to 4.

`>> # /cfg/l2/lacp/port 2/mode active`	(Select port 2 mode of operation)
`>> # /cfg/l2/lacp/port 3/mode active`	(Select port 3 mode of operation)
`>> # /cfg/l2/lacp/port 4/mode active`	(Select port 4 mode of operation)

4. Define the admin key on ports 2 to 4.

`>> # /cfg/l2/lacp/port 2/adminkey 100`	(Select port 2 adminkey to 100)
`>> # /cfg/l2/lacp/port 3/adminkey 100`	(Select port 3 adminkey to 100)
`>> # /cfg/l2/lacp/port 4/adminkey 100`	(Select port 4 adminkey to 100)

5. Apply and verify the configuration.

`>> LACP port 4# apply`	(Make your changes active)
`>> LACP port 4# cur`	(View current trunking configuration)

6. Save your new configuration changes.

`>> LACP port 4# save`	(Save for restore after reboot)

Chapter 6 – Port Teaming

Port teaming is a feature deployed in scenarios where the Virtual Router Redundancy Protocol (VRRP) is not used to detect link failures. For more information on VRRP, see High Availability, page 507. If an uplink connection fails, then Alteon notifies uplink routers and switches of the failure instead of waiting for the routers and switches to time out.

This feature is also used to team ports or trunks so that when one port or trunk in the team is down, all others in the team are operationally disabled. Alteon supports a maximum of 8 port teams.

To create a simple two-port team

1. Create a new port team.

```
>> Main# /cfg/l2/team 1
```

2. Add ports to the new team.

```
>> Port Team 1# addport 1
>> Port Team 1# addport 2
```

3. Enable port team.

```
>> Port Team 1# ena
```

To create a simple two-trunk team

1. Create a new port team.

```
>> Main# /cfg/l2/team 2
```

2. Add trunks to the new team.

```
>> Port Team 2# addtrunk 1
>> Port Team 2# addtrunk 2
```

3. Enable port team.

```
>> Port Team 2# ena
```

In both of these examples, the teams are placed in **passive** mode with either the ports or trunks operational. The team is in passive mode when all ports or trunks are operational, and the team is waiting for any one of the ports or trunks to become disabled. When one of the ports or trunks is disabled, the team goes to **active** mode and the other ports or trunks in the team are operationally disabled. The port or trunk that triggered this becomes the master port or trunk.

When the master port or trunk becomes operational once more, the other ports or trunks in the team are operationally enabled. When all the ports or trunks are operational, the team goes back to passive mode.

In some cases when the ports and trunks are operationally enabled, some of the other ports or trunks in the team are not operational either because of a link going down, or because they were operationally disabled or were set as disabled. If this happens, the team goes into **off** mode. In this mode, the team waits until all ports or trunks are operational before going back to passive mode to repeat the cycle.

Chapter 7 – Spanning Tree Protocol

When multiple paths exist on a network, the Spanning Tree Protocol (STP) configures the network so that Alteon uses only the most efficient path.

The following topics are addressed in this chapter:

Overview

STP detects and eliminates logical loops in a bridged or switched network. STP forces redundant data paths into a standby (blocked) state. When multiple paths exist, STP configures the network so that an Alteon uses only the most efficient path. If that path fails, STP sets up another active path on the network to sustain network operations.

The relationship between port, trunk groups, VLANs, and spanning trees is described in Table 11:

Table 11: Relationship Between Port, Trunk Groups, VLANs, and Spanning Trees

Alteon Element	Belongs to
Port	Trunk group or one or more VLANs
Trunk group	One or more VLANs
VLAN	One STP group

Note: Due to STP's sequence of listening, learning, and forwarding or blocking, lengthy delays may occur. For more information on using STP in cross-redundant topologies, see Eliminating Loops with STP and VLANs, page 568.

Bridge Protocol Data Units (BPDUs)

To create a spanning tree, Alteon generates a configuration Bridge Protocol Data Unit (BPDU), which it then forwards out of its ports. All devices in the Layer 2 network participating in the spanning tree gather information about other devices in the network through an exchange of BPDUs.

A BPDU is a 64-byte packet that is sent out at a configurable interval, which is typically set at 2 seconds. The BPDU is used to establish a path, much like a "hello" packet in IP routing. BPDUs contain information about the transmitting bridge and its ports, including bridge and MAC addresses, bridge priority, port priority, and path cost. If the ports are tagged, each port sends out a special BPDU containing the tagged information.

The generic action of an Alteon on receiving a BPDU is to compare the received BPDU to its own BPDU that it transmits. If the received BPDU is better than its own BPDU, it will replace its BPDU with the received BPDU. Then, Alteon adds its own bridge ID number and increments the path cost of the BPDU. Alteon uses this information to block any necessary ports.

Determining the Path for Forwarding BPDUs

When determining which port to use for forwarding and which port to block, Alteon uses information in the BPDU, including each bridge priority ID. A technique based on the "lowest root cost" is then computed to determine the most efficient path for forwarding.

For more information on bridge priority, port priority, and port cost, refer to the *Alteon Application Switch Operating System Command Reference*. Much like least-cost routing, root cost assigns lower values to high-bandwidth ports, such as Gigabit Ethernet, to encourage their use. For example, a 10 Mbps link has a “cost” of 100, a 100 Mbps (Fast Ethernet) link carries a cost of 10, and a 1000 Mbps (or Gigabit Ethernet) link has a cost of 1. The objective is to use the fastest links so that the route with the lowest cost is chosen.

Bridge Priority

The bridge priority parameter controls which bridge on the network is the STP root bridge. To make one Alteon the root bridge, configure the bridge priority lower than all other switches and bridges on your network. The lower the value, the higher the bridge priority. The bridge priority is configured using the **`/cfg/l2/stg/brg/prior`** command.

Port Priority

The port priority helps determine which bridge port becomes the designated port. In a network topology that has multiple bridge ports connected to a single segment, the port with the lowest port priority becomes the designated port for the segment. The port priority is configured using the **`/cfg/l2/stg/port/prior`** command.

Port Path Cost

The port path cost assigns lower values to high-bandwidth ports, such as Gigabit Ethernet, to encourage their use. The cost of a port also depends on whether the port operates at full-duplex (lower cost) or half-duplex (higher cost). For example, if a 100 Mbps (Fast Ethernet) link has a "cost" of 10 in half-duplex mode, it will have a cost of 5 in full-duplex mode. The objective is to use the fastest links so that the route with the lowest cost is chosen. A value of 0 indicates that the default cost will be computed for an auto-negotiated link speed.

Spanning Tree Group Configuration Guidelines

This section provides guidelines for configuring STGs, including:

- Adding a VLAN to a Spanning Tree Group, page 101
- Creating a VLAN, page 101
- Rules for VLAN-Tagged Ports, page 101
- Adding and Removing Ports to and from STGs, page 101
- Spanning Tree Implementations in Trunk Groups, page 102

Adding a VLAN to a Spanning Tree Group

- If no VLANs exist beyond the default VLAN 1, see Creating a VLAN, page 101 for information on adding ports to VLANs.
- Add the VLAN to the STG using the **/cfg/l2/stg <stg-#> /add <vlan-number>** command.

Creating a VLAN

- When you create a VLAN, that VLAN belongs to STG 1, the default STG. If you want the VLAN in another STG, you must move the VLAN by assigning it to another STG.

  Move a newly created VLAN to an existing STG by following this order:

  a. Create the VLAN

  b. Add the VLAN to an existing STG
- If ports are tagged, all trunked ports can belong to multiple STGs.
- A port that is not a member of any VLAN cannot be added to any STG. The port must be added to a VLAN, and that VLAN added to the desired STG.

Rules for VLAN-Tagged Ports

- Tagged ports can belong to more than one STG, but untagged ports can belong to only one STG.
- When a tagged port belongs to more than one STG, the egress BPDUs are tagged to distinguish the BPDUs of one STG from those of another STG.
- An untagged port cannot span multiple STGs.

Adding and Removing Ports to and from STGs

This section includes the following sub-sections:

Adding a Port

When you add a port to a VLAN that belongs to an STG, the port is also added to the STG. However, if the port you are adding is an untagged port and is already a member of an STG, that port is not added to an additional STG because an untagged port cannot belong to more that one STG.

Example

VLAN1 belongs to STG1. You add an untagged port, port 1, that does not belong to any STG to VLAN1, and port 1 becomes part of STG1.

If you add untagged port 5 (which is a member to STG2) to STG1, Alteon prompts you to change the PVID from 2 to 1:

```
"Port 5 is an UNTAGGED port and its current PVID is 2. Confirm changing PVID
from 2 to 1 [y/n]:" y
```

Removing a Port

When you remove a port from a VLAN that belongs to an STG, that port will also be removed from the STG. However, if that port belongs to another VLAN in the same STG, the port remains in the STG.

Example

Port 1 belongs to VLAN1, and VLAN1 belongs to STG1. When you remove port 1 from VLAN1, port 1 is also removed from STG1.

However, if port 1 belongs to both VLAN1 and VLAN2 and both VLANs belong to STG1, removing port 1 from VLAN1 does not remove port 1 from STG1 because VLAN2 is still a member of STG1.

Disabling an STG

An STG cannot be deleted, only disabled. If you disable the STG while it still contains VLAN members, STP will be off on all ports belonging to that VLAN.

Spanning Tree Implementations in Trunk Groups

In both Cisco and Alteon spanning tree implementations as described in Spanning Tree Group Configuration Guidelines, page 100, the trunking methodology applies to both the default and non-default STGs. Make sure that all members of the trunk group are configured to the correct STG parameters, and determine whether to enable use of the Alteon multiple STG mode.

Caution: All ports that are within a trunk group should be configured to have the same spanning tree and VLAN parameters. Spanning tree parameters should not be changed on individual ports that belong to a trunk group. To change spanning tree parameters on one or more ports belonging to a trunk group, first remove individual members from the trunk group.

Multiple Spanning Trees

Alteon supports the Multiple Spanning Tree Protocol (MSTP) and Rapid Spanning Tree Protocol (RSTP) as defined in the IEEE 802.1S (MSTP) and 802.1W (RSTP) standards. This is an improvement over previous spanning tree implementations in that it is a standards-based approach to implementing this functionality.

Before the 802.1S standard, MSTP was implemented through a variety of proprietary protocols such as Alteon MSTP and Cisco PVST+. Each one of these proprietary protocols had advantages and disadvantages but they were never interoperable. The 801.S standard solves this by creating standards-based MSTP. The 802.1W standard takes the same approach in creating standards-based RSTP.

In this implementation of MSTP, up to 2048 VLANs can be mapped to any of the 16 spanning tree instances. Each spanning tree instance handles multiple VLANs that have the same Layer 2 topology but each spanning tree instance can have a topology independent of other instances. Also, MSTP provides multiple forwarding paths for data traffic, enables load balancing, and improves overall network fault tolerance.

This implementation of RSTP improves upon previous implementations by addressing slow convergence times.

Note: By default, all newly created VLANs are members of STG1.

For specific information on MSTP and RSTP, see:

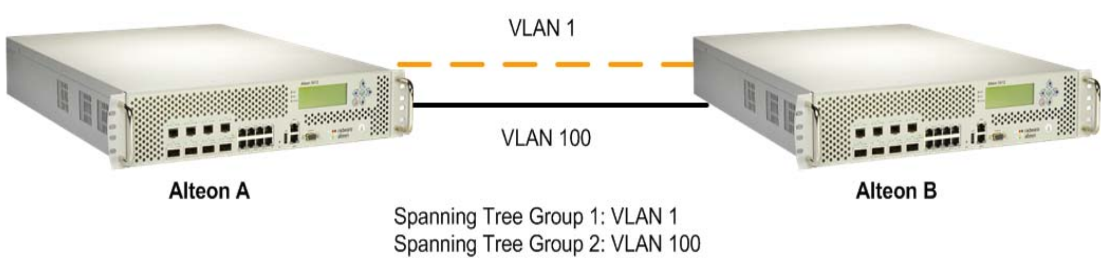
- Rapid Spanning Tree Protocol, page 105
- Multiple Spanning Tree Protocol, page 107

Purpose of Multiple Spanning Trees

Figure 7 - Example Multiple Spanning Tree Configuration, page 103 illustrates the purpose of multiple spanning trees. Two VLANs, VLAN 1 and VLAN 100 exist between Alteon A and Alteon B. If you have a single STG, the Alteons detect an apparent loop, and one VLAN may become blocked, affecting connectivity, even though no actual loop exists.

If VLAN 1 and VLAN 100 belong to different STGs, then the two spanning tree instances separate the topology without forming a loop. Both VLANs can forward packets between the Alteons without losing connectivity.

Figure 7: Example Multiple Spanning Tree Configuration



Four-Alteon Topology with a Single Spanning Tree

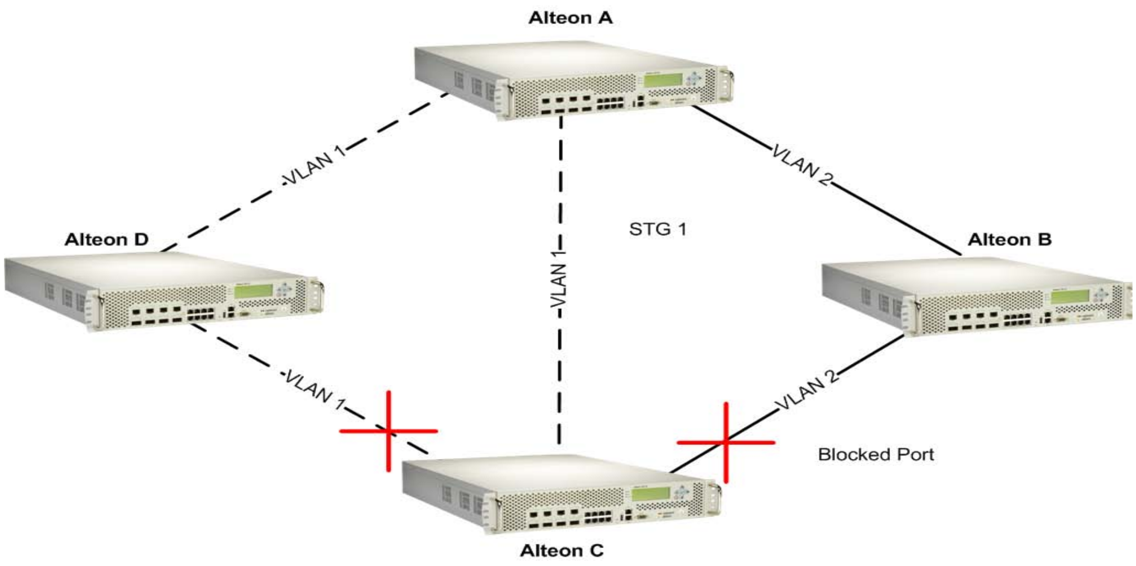
In a four-Alteon topology (see Figure 8 - Four-Alteon Topology with a Single Spanning Tree, page 104), and assuming Alteon A has a higher priority, you can have at least three loops on the network:

- Data flowing from Alteons A to B to C and back to Alteon A.
- Data flowing from Alteons A to C to D and back to Alteon A
- Data flowing from Alteons A to B to C to D and back to Alteon A.

With a single spanning tree environment, you have two links blocked to prevent loops on the network. It is possible that the blocks may be between Alteons C and D and between Alteons B and C, depending on the bridge priority, port priority, and port cost. The two blocks would prevent looping on the network, but the blocked link between Alteons B and C will inadvertently isolate VLAN 3 altogether.

For more information on bridge priority, port priority, and port cost, see the *Alteon Application Switch Operating System Command Reference*.

Figure 8: Four-Alteon Topology with a Single Spanning Tree

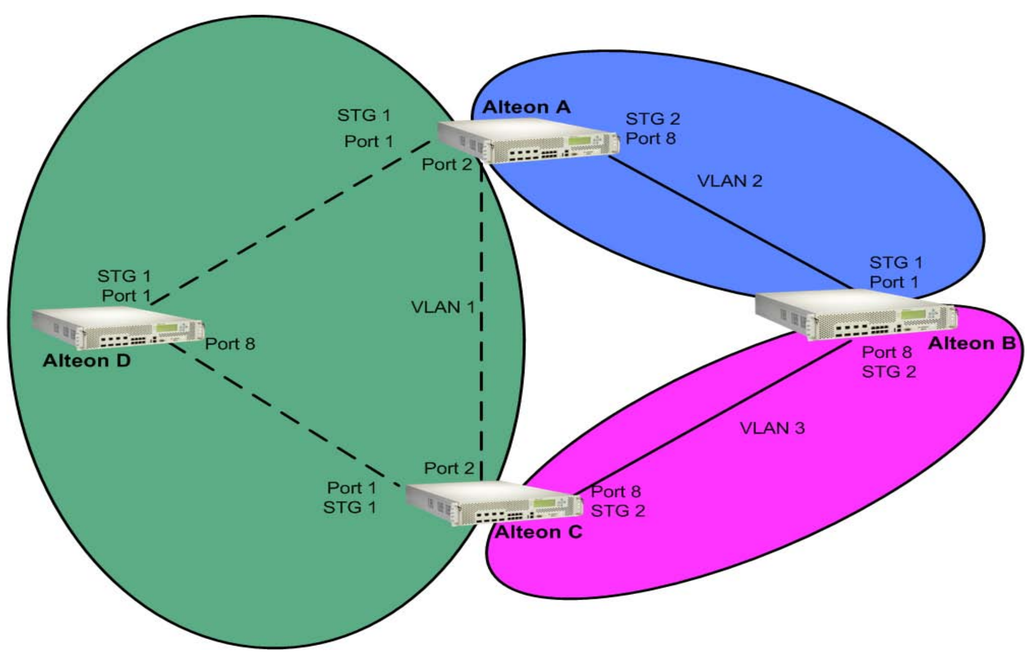


Four-Alteon Topology with Multiple Spanning Trees

If multiple spanning trees are implemented and each VLAN is on a different spanning tree, elimination of logical loops will not isolate any VLAN.

Figure 9 - Four-Alteon Topology with a Multiple Spanning Tree, page 104 shows the same four-Alteon topology as in Figure 8 - Four-Alteon Topology with a Single Spanning Tree, page 104, but with multiple spanning trees enabled. The VLANs are identified on each of the three shaded areas connecting the Alteons. The port numbers are shown next to each Alteon. The STG number for each VLAN is shown at each Alteon.

Figure 9: Four-Alteon Topology with a Multiple Spanning Tree



Two spanning tree instances are configured in this example. To identify the STG a VLAN is participating in for each Alteon, see Multiple Spanning Tree Groups per VLAN Example, page 105.

Table 12 provides a summary of this example:

Table 12: Multiple Spanning Tree Groups per VLAN Example

Alteon	VLAN 1	VLAN 2	VLAN 3
Alteon A	STG1 Ports 1 and 2	STG2 Port 8	
Alteon B		STG1 Port 1	STG2 Port 8
Alteon C	STG1 Ports 1 and 2		STG2 Port 8
Alteon D	STG1 Ports 1 and 8		

Alteon-Centric Spanning Tree Protocol

In Figure 9 - Four-Alteon Topology with a Multiple Spanning Tree, page 104, VLAN 2 is shared by Alteons A and B on ports 8 and 1 respectively. Alteon A identifies VLAN 2 in STG2 and Alteon B identifies VLAN 2 in STG1. An STG is Alteon-centric. It is used to identify the VLANs participating in the STGs. The STG ID is not transmitted in the BPDU. Each spanning tree decision is based on the configuration of that Alteon.

VLAN Participation in Spanning Tree Groups

The VLAN participation for each STG in Figure 9 - Four-Alteon Topology with a Multiple Spanning Tree, page 104 is summarized as follows:

- **VLAN 1 Participation**—If Alteon A is the root bridge, then Alteon A transmits the BPDU for VLAN 1 on ports 1 and 2. Alteon C receives the BPDU on its port 2 and Alteon D receives the BPDU on its port 1. Alteon D blocks port 8 or Alteon C blocks port 1 depending on the information provided in the BPDU.
- **VLAN 2 Participation**—Alteon A, the root bridge generates another BPDU for STG2 and forwards it out from port 8. Alteon B receives this BPDU on its port 1. Port 1 on Alteon B is on VLAN 2, STG1. Because Alteon B has no additional ports participating in STG1, this BPDU is not be forwarded to any additional ports and Alteon A remains the designated root.
- **VLAN 3 Participation**—For VLAN 3 you can have Alteon B or C to be the root bridge. If Alteon B is the root bridge for VLAN 3, STG2, then Alteon B transmits the BPDU out from port 8. Alteon C receives this BPDU on port 8 and is identified as participating in VLAN 3, STG2. Since Alteon C has no additional ports participating in STG2, this BPDU is not forwarded to any additional ports and Alteon B remains the designated root.

Rapid Spanning Tree Protocol

Rapid Spanning Tree Protocol (RSTP) provides rapid convergence of the spanning tree and provides for the fast reconfiguration critical for networks carrying delay-sensitive traffic such as voice and video. RSTP significantly reduces the time to reconfigure the active topology of the network when changes occur to the physical topology or its configuration parameters. RSTP reduces the bridged-LAN topology to a single spanning tree.

RSTP parameters are configured in STG1. STP Groups 2 through 32 do not apply to RSTP, and must be cleared. There are new STP parameters to support RSTP, and some values to existing parameters are different.

RSTP is compatible with devices that run 802.1d Spanning Tree Protocol. If Alteon detects 802.1d BPDUs, it responds with 802.1d-compatible data units. RSTP is not compatible with the Per VLAN Spanning Tree (PVST+) protocol.

Port State Changes

The port state controls the forwarding and learning processes of a spanning tree. In RSTP, the port state is consolidated as follows: discarding, learning, and forwarding. Table 13 - Comparison of Port States Between STP and RSTP, page 106 compares the port states between 802.1d Spanning Tree and 802.1w Rapid Spanning Trees.

Table 13: Comparison of Port States Between STP and RSTP

Operational status	STP Port State	RSTP Port State
Enabled	Blocking	Discarding
Enabled	Listening	Discarding
Enabled	Learning	Learning
Enabled	Forwarding	Forwarding
Disabled	Disabled	Discarding

Port Type and Link Type

The spanning tree configuration includes the **edge port** and **link type** parameters to support RSTP and MSTP. Although these parameters are configured for STGs 1 through 32, they only take effect when RSTP/MSTP is turned on.

- **Edge Port**—A port that does not connect to a bridge is called an **edge port**. Edge ports are generally connected to a server. Edge ports can start forwarding as soon as the link is up.

  Edge ports do not take part in a spanning tree configuration, and should not receive BPDUs. If a port with edge enabled does receive a BPDU, it begins STP processing only if it is connected to a spanning tree bridge. If it is connected to a host, the edge port ignores BPDUs.
- **Link Type**—The **link type** determines how the port behaves with rapid spanning trees. The link type corresponds to the duplex mode of the port. A full-duplex link is point-to-point (**p2p**), while a half-duplex link should be configured as **shared**. If you select **auto** as the link type, the port dynamically configures the link type.

RSTP Configuration Guidelines

Follow these guidelines when configuring Rapid Spanning Tree Groups:

- When RSTP is turned on, STP parameters apply only to STP Group 1.
- When RSTP is turned on, all VLANs (including the management VLAN 4095) are moved to STG1.

Example RSTP Configuration

1. Create VLAN and add ports.

   Once ports have been readied for VLAN membership, VLAN 3 can be created and the ports added to the VLAN.

```
>> Main# /cfg/l2/vlan 2
<If the VLAN was not already created, it would be created with this command.>
>> VLAN 2# add 2
>> VLAN 2# add 3
>> VLAN 2# add 4
```

2. Disable and clear STP groups 2 through 32.

`>> Main# /cfg/l2/stg 2`	(Select STP Group 2)
`>> Spanning Tree Group 2# clear`	(Clear STP Group 2 parameters)
`>> Spanning Tree Group 2# off`	(Turn off STP Group 2)

3. Set the spanning tree mode to rapid spanning tree.

`>> Main# /cfg/l2/mrst`	(Select *Multiple Spanning Tree* menu)
`>> Multiple Spanning Tree# mode rstp`	(Set mode to Rapid Spanning Tree)
`>> Multiple Spanning Tree# on`	(Turn Rapid Spanning Tree on)

4. Configure STP Group 1 parameters.

`>> /cfg/l2/stg 1`	(Select *Spanning Tree Protocol* menu)
`>> Spanning Tree Group 1# add 2`	(Add VLAN 2 to STP Group 1)
`>> Spanning Tree Group 1# apply`	(Apply the configurations)
`>> Spanning Tree Group 1# save`	(Save the configuration)

Multiple Spanning Tree Protocol

IEEE 802.1s Multiple Spanning Tree Protocol (MSTP) extends the IEEE 802.1w RSTP through multiple STGs. MSTP maintains up to 16 spanning-tree instances that correspond to STP Groups 1 through 16.

In MSTP, several VLANs can be mapped to each spanning tree instance. Each spanning tree instance is independent of other instances. MSTP allows frames assigned to different VLANs to follow separate paths, each path based on an independent spanning tree instance. This approach provides multiple forwarding paths for data traffic, enabling load balancing, and reducing the number of spanning tree instances required to support a large number of VLANs.

By default, the spanning tree on the management ports is turned off in both STP/PVST+ mode and in MSTP/RSTP mode.

MSTP Region

A group of interconnected bridges that share the same attributes is called a Multiple Spanning Tree (MST) region. Each bridge within the region must share the following attributes:

- Alphanumeric name
- Version number
- VLAN-to-STG mapping scheme

MSTP provides rapid reconfiguration, scalability and control due to the support of regions, and multiple spanning tree instances support within each region.

Common Internal Spanning Tree

The Common Internal Spanning Tree (CIST) provides a common form of STP, with one spanning tree instance that can be used throughout the MSTP region. CIST allows Alteon to operate with legacy equipment, including devices that run IEEE 802.1d (STP).

CIST allows the MSTP region to act as a virtual bridge to other bridges outside of the region, and provides a single spanning tree instance to interact with them.

CIST port configuration includes Hello time, edge port enable/disable, and link type. These parameters do not affect STGs 1 through 16. They apply only when the CIST is used.

MSTP Configuration Guidelines

Follow these guidelines when configuring MSTP:

- When MSTP is turned on, Alteon moves management VLAN 4095 to the CIST. When MSTP is turned off, Alteon moves VLAN 4095 from the CIST to STG16.
- When enabling MSTP, the region name must be configured, and the default version number set to 1. Each bridge in the region must have the same name, version number, and VLAN mapping.

Example MSTP Configuration

1. Ready ports for VLAN membership.

   To create a VLAN, ports must first be readied for VLAN membership. To do this, the port PVID is changed from the default of 1 to 2, indicating that the ports are a part of VLAN 2.

```
>> Main# /cfg/port 2/pvid 2
>> Main# /cfg/port 3/pvid 2
>> Main# /cfg/port 4/pvid 2
```

2. Create VLAN and add ports. Once ports have been readied for VLAN membership, VLAN 3 can be created and the ports added to the VLAN.

   Note: If the VLAN was not already created, it would be created with this command.

```
>> Main# /cfg/l2/vlan 2
>> VLAN 2# add 2
>> VLAN 2# add 3
>> VLAN 2# add 4
```

3. Set the mode to Multiple Spanning Tree, and configure MSTP region parameters.

`>> Main# /cfg/l2/mrst`	(Select *Multiple Spanning Tree* menu)
`>> Multiple Spanning Tree# mode mstp`	(Set mode to Multiple Spanning Trees)
`>> Multiple Spanning Tree# on`	(Turn Multiple Spanning Trees on)
`>> Multiple Spanning Tree# name xxxxxx`	(Define the region name)

4. Assign VLANs to STGs.

```
>> Main# /cfg/l2/stg 2
>> Spanning Tree Group 2# add 2
```

5. Turn off Layer 3 forwarding.

```
>> Main# /cfg/l3/frwd off
>> IP Forwarding# apply
>> IP Forwarding# save
```

Chapter 8 – Basic IP Routing

This chapter provides configuration background and examples for performing IP routing functions, and includes the following topics:

IP Routing Benefits

Alteon uses a combination of configurable IP interfaces and IP routing options. The IP routing capabilities provide the following benefits:

- Connects the server IP subnets to the rest of the backbone network.
- Performs Server Load Balancing (using both Layer 3 and Layer 4 in combination) to server subnets that are separate from backbone subnets.
- Routing IP traffic between multiple Virtual Local Area Networks (VLANs) configured on Alteon.

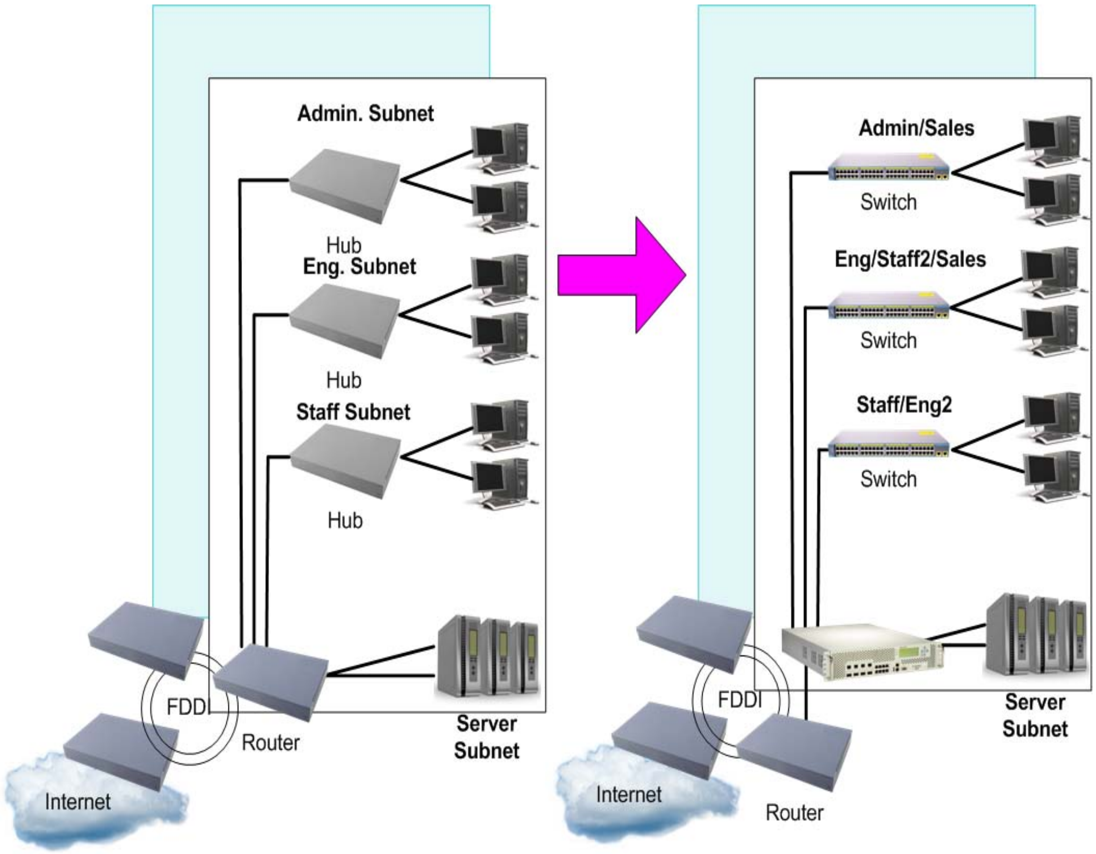
Routing Between IP Subnets

The physical layout of most corporate networks has evolved over time. Classic hub and router topologies have given way to faster switched topologies, particularly now that switches are increasingly intelligent. Alteon is intelligent and fast enough to perform routing functions on a par with wire speed Layer 2 switching.

The combination of faster routing and switching in a single Alteon provides another service—it enables you to build versatile topologies that account for legacy configurations.

Figure 10 - Example Topology Migration, page 112 illustrates an example topology migration:

Figure 10: Example Topology Migration



In this example, a corporate campus has migrated from a router-centric topology to a faster, more powerful, switch-based topology. The legacy of network growth and redesign has left the system with a mix of illogically distributed subnets.

This is a situation that switching alone cannot normalize. Instead, the router is flooded with cross-subnet communication. This compromises efficiency in two ways:

- Routers can be slower than switches. The cross-subnet side trip from the switch to the router and back again adds two hops for the data, slowing throughput considerably.
- Traffic to the router increases, increasing congestion.

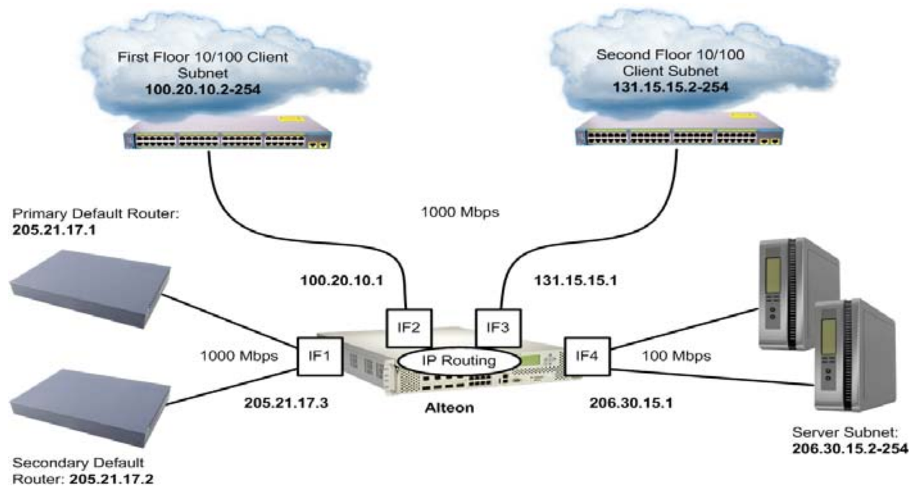
Even if every end-station could be moved to better logical subnets, competition for access to common server pools on different subnets still burdens the routers.

This problem is solved by using Alteon with built-in IP routing capabilities. Cross-subnet LAN traffic can now be routed within Alteon with wire speed Layer 2 switching performance. This not only eases the load on the router but saves the network administrators from reconfiguring each and every end-station with new IP addresses.

Subnet Routing Example

The following is an example of IP subnet routing using Alteon:

Figure 11: Example Configuration IP Subnet Routing with Alteon



Alteon connects the Gigabit Ethernet trunks from various switched subnets throughout one building. Common servers are placed on another subnet attached to Alteon. A primary and backup router are attached to Alteon on yet another subnet.

Without Layer 3 IP routing, cross-subnet communication is relayed to the default gateway (in this case, the router) for the next level of routing intelligence. The router fills in the necessary address information and sends the data back to Alteon, which then relays the packet to the proper destination subnet using Layer 2 switching.

With Layer 3 IP routing in place, routing between different IP subnets can be accomplished entirely within Alteon. This leaves the routers free to handle inbound and outbound traffic for this group of subnets.

Example IP Subnet Routing Configuration

Notes

- Prior to configuration, you must be connected to the CLI as the administrator.
- For details about accessing and using any of the menu commands described in this example, see the *Alteon Application Switch Operating System Command Reference*.

1. Assign an IP address (or document the existing one) for each real server, router, and client workstation.

In the example configuration in Figure 11 - Example Configuration IP Subnet Routing with Alteon, page 113, the following IP addresses are used:

Table 14: Subnet Routing Example IP Address Assignments

Subnet	Devices	IP Addresses
1	Primary and Secondary Default Routers	205.21.17.1 and 205.21.17.2
2	First Floor Client Workstations	100.20.10.2-254
3	Second Floor Client Workstations	131.15.15.2-254
4	Common Servers	206.30.15.2-254

2. Assign an IP interface for each subnet attached to Alteon. Since there are four IP subnets connected to Alteon, four IP interfaces are needed:

Table 15: Subnet Routing Example: IP Interface Assignments

Interface	Devices	IP Interface Address
IF 1	Primary and Secondary Default Routers	205.21.17.3
IF 2	First Floor Client Workstations	100.20.10.1
IF 3	Second Floor Client Workstations	131.15.15.1
IF 4	Common Servers	206.30.15.1

Use the following commands to configure the IP interfaces:

```
>> #  /cfg/l3/if 1                          (Select IP interface 1)
>> IP Interface 1#  addr 205.21.17.3        (Assign IP address for the interface)
>> IP Interface 1#  ena                     (Enable IP interface 1)
>> IP Interface 1#  /cfg/l3/if 2            (Select IP interface 2)
>> IP Interface 2#  addr 100.20.10.1        (Assign IP address for the interface)
>> IP Interface 2#  ena                     (Enable IP interface 2)
>> IP Interface 2#  /cfg/l3/if 3            (Select IP interface 3)
>> IP Interface 3#  addr 131.15.15.1        (Assign IP address for the interface)
>> IP Interface 3#  ena                     (Enable IP interface 3)
>> IP Interface 3#  /cfg/l3/if 4            (Select IP interface 4)
>> IP Interface 4#  addr 206.30.15.1        (Assign IP address for the interface)
>> IP Interface 4#  ena                     (Enable IP interface 5)
```

3. Set each server and workstation's default gateway to the appropriate IP interface (the one in the same subnet as the server or workstation).

4. Configure the default gateways to the routers' addresses. This allows Alteon to send outbound traffic to the routers:

```
>> IP Interface 5#  /cfg/l3/gw 1            (Select primary default gateway)
>> Default gateway 1#  addr 205.21.17.1     (Assign IP address for primary router)
>> Default gateway 1#  ena                  (Enable primary default gateway)
>> Default gateway 1#  /cfg/l3/gw 2         (Select secondary default gateway)
```

`>> Default gateway 2# addr 205.21.17.2`	(Assign address for secondary router)
`>> Default gateway 2# ena`	(Enable secondary default gateway)

5. Enable, apply, and verify the configuration.

`>> Default gateway 2# /cfg/l3/fwrd`	(Select the *IP Forwarding* Menu)
`>> IP Forwarding# on`	(Turn IP forwarding on)
`>> IP Forwarding# apply`	(Make your changes active)
`>> IP Forwarding# /cfg/l3/cur`	(View current IP settings)

Examine the resulting information. If any settings are incorrect, make the appropriate changes.

6. Save your new configuration changes.

`>> IP# save`	(Save for restore after reboot)

Using VLANs to Segregate Broadcast Domains

In Figure 10 - Example Topology Migration, page 112, devices that share a common IP network are all in the same broadcast domain. If you want to limit the broadcasts on your network, you could use VLANs to create distinct broadcast domains. For example, as shown in the following procedure, you could create one VLAN for the client trunks, one for the routers, and one for the servers.

To segregate broadcast domains using VLANs

Note: This procedure uses the configuration in Figure 10 - Example Topology Migration, page 112 as its baseline.

1. Determine which ports and IP interfaces belong to which VLANs. Table 16 includes port and VLAN information used in this example:

Table 16: Subnet Routing Example Optional VLAN Ports

VLAN	Devices	IP Interface	Port	VLAN #
1	First Floor Client Workstations	2	1	1
	Second Floor Client Workstations	3	2	1
2	Primary Default Router	1	3	2
	Secondary Default Router	1	4	2
3	Common Servers 1	4	5	3
	Common Servers 2	4	6	3

2. Add the ports to their respective VLANs. The VLANs are configured as follows:

`>> # /cfg/l2/vlan 1`	(Select VLAN 1)
`>> VLAN 1# add port 1`	(Add port for 1st floor to VLAN 1)
`>> VLAN 1# add port 2`	(Add port for second floor to VLAN 1)
`>> VLAN 1# ena`	(Enable VLAN 1)
`>> VLAN 1# /cfg/l2/vlan 2`	(Select VLAN 2)
`>> VLAN 2# add port 3`	(Add port for default router 1)
`>> VLAN 2# add port 4`	(Add port for default router 2)
`>> VLAN 2# ena`	(Enable VLAN 2)
`>> VLAN 2# /cfg/l2/vlan 3`	(Add port for default router 3)
`>> VLAN 3# add port 5`	(Select VLAN 3)
`>> VLAN 3# add port 6`	(Select port for common server 1)
`>> VLAN 3# ena`	(Enable VLAN 3)

Each time you add a port to a VLAN, you may get the following prompt:

```
Port 4 is an untagged port and its current PVID is 1.
Confirm changing PVID from 1 to 2  [y/n] ?
```

Enter **y** to set the default Port VLAN ID (PVID) for the port.

3. Add each IP interface to the appropriate VLAN.

Now that the ports are separated into three VLANs, the IP interface for each subnet must be placed in the appropriate VLAN. Based on the configuration in step 2, the settings are made as follows:

`>> VLAN 3# /cfg/l3/if 1`	(Select IP interface 1 for default routers)
`>> IP Interface 1# vlan 2`	(Set to VLAN 2)
`>> IP Interface 1# /cfg/l3/if 2`	(Select IP interface 2 for first floor)
`>> IP Interface 2# vlan 1`	(Set to VLAN 1)
`>> IP Interface 2# /cfg/l3/if 3`	(Select IP interface 3 for second floor)
`>> IP Interface 3# vlan 1`	(Set to VLAN 1)
`>> IP Interface 3# /cfg/l3/if 4`	(Select IP interface 4 for servers)
`>> IP Interface 4# vlan 3`	(Set to VLAN 3)

4. Apply and verify the configuration.

`>> IP Interface 5# apply`	(Make your changes active)
`>> IP Interface 5# /info/l2/vlan`	(View current VLAN information)
`>> Layer 2# /info/port`	(View current port information)

Examine the resulting information. If any settings are incorrect, make the appropriate changes.

5. Save your new configuration changes.

```
>> Information#  save
```

Defining IP Address Ranges for the Local Route Cache

A local route cache lets you use Alteon resources more efficiently. The local network address and local network mask parameters (accessed via the **/cfg/l3/frwd/local/add** command) define a range of addresses that are cached on Alteon. The *local network address* is used to define the base IP address in the range that will be cached. The *local network mask* is applied to produce the range. To determine if a route should be added to the memory cache, the destination address is masked (bit-wise AND) with the local network mask and checked against the local network address.

By default, the local network address and local network mask are both set to 0.0.0.0. This produces a range that includes all Internet addresses for route caching: 0.0.0.0 through 255.255.255.255.

To limit the route cache to your local hosts, you could configure the parameters as shown in Table 17:

Table 17: Example Local Routing Cache Address Ranges

Local Host Address Range	Local Network Address	Local Network Mask
0.0.0.0–127.255.255.255	0.0.0.0	128.0.0.0
128.0.0.0–128.255.255.255	128.0.0.0	128.0.0.0 or 255.0.0.0
205.32.0.0–205.32.255.255	205.32.0.0	255.255.0.0

Dynamic Host Configuration Protocol

The Dynamic Host Configuration Protocol (DHCP) is a transport protocol that provides a framework for assigning IP addresses and configuration information to other IP hosts or clients in a large TCP/IP network. Without DHCP, the IP address must be entered manually for each network device. DHCP allows a network administrator to distribute IP addresses from a central point and send a new IP address when a device is connected to a different place in the network.

DHCP is an extension of another network IP management protocol, the Bootstrap Protocol (BOOTP), with an additional capability of dynamically allocating reusable network addresses and configuration parameters for client operation.

Built on the client/server model, DHCP allows hosts or clients on an IP network to obtain their configurations from a DHCP server, thereby reducing the network administration effort. The most significant configuration the client receives from the server is its required IP address. Other optional parameters include the "generic" file name to be booted, the address of the default gateway, and so on.

The DHCP relay agent eliminates the need to have DHCP/BOOTP servers on every subnet. It allows the administrator to reduce the number of DHCP servers deployed on the network and to centralize them. Without the DHCP relay agent, there must be at least one DHCP server deployed at each subnet that has hosts that need to perform the DHCP request.

DHCP Relay Agent

DHCP is described in *RFC 2131*, and the DHCP relay agent supported on Alteon is described in *RFC 1542*. DHCP uses UDP as its transport protocol. The client sends messages to the server on port 67 and the server sends messages to the client on port 68.

DHCP defines the methods through which clients can be assigned an IP address for a finite lease period and allows reassignment of the IP address to another client later. Additionally, DHCP provides the mechanism for a client to gather other IP configuration parameters it needs to operate in the TCP/IP network.

In the DHCP environment, Alteon acts as a relay agent. The DHCP relay feature (**/cfg/l3/bootp**) enables Alteon to forward a client request for an IP address to two BOOTP servers with configured IP addresses.

When Alteon receives a UDP broadcast on port 67 from a DHCP client requesting an IP address, Alteon acts as a proxy for the client, replacing the client source IP (SIP) and destination IP (DIP) addresses. The request is then forwarded as a UDP Unicast MAC layer message to two BOOTP servers with configured IP addresses. The servers respond with a UDP Unicast message back to Alteon, with the default gateway and IP address for the client. The destination IP address in the server response represents the interface address that received the client request. This interface address instructs Alteon on which VLAN to send the server response to the client.

DHCP Relay Agent Configuration

To enable Alteon as the BOOTP forwarder, you need to configure the DHCP/BOOTP server IP addresses. Generally, you should configure the command IP interface closest to the client so that the DHCP server knows from which IP subnet the newly allocated IP address should come.

Figure 12 - Example Basic DHCP Network, page 118 illustrates a basic DHCP network example:

Figure 12: Example Basic DHCP Network

In this Alteon implementation, there is no need for primary or secondary servers. The client request is forwarded to the BOOTP servers configured. Using two servers provides failover redundancy. However, no health checking is supported.

To configure a DHCP relay agent

1. Use the following commands to configure Alteon as a DHCP relay agent:

`>> # /cfg/l3/bootp`		
`>> Bootstrap Protocol Relay#`	`addr`	(Set IP address of BOOTP server)
`>> Bootstrap Protocol Relay#`	`addr2`	(Set IP address of second BOOTP server)
`>> Bootstrap Protocol Relay#`	`on`	(Globally turn BOOTP relay on)
`>> Bootstrap Protocol Relay#`	`off`	(Globally turn BOOTP relay off)
`>> Bootstrap Protocol Relay#`	`cur`	(Display the current configuration)

2. Additionally, DHCP Relay functionality can be assigned on a per-interface basis. Use the following command to enable the Relay functionality:

```
>> #/cfg/l3/if <interface number> /relay ena
```

Gratuitous ARP (GARP) Command

Gratuitous ARP packets are used to force a next-hop router to learn an IP and MAC pair. For security reasons, this command can only be used for an IP address belonging to a VIP, PIP, or interface.

Use the GARP command as follows:

```
>> Main#/oper/ip/garp <IP Address> <VLAN Number>
```

Static Routes

Alteon has two basic mechanisms for learning networking routes:

- **Dynamic routes**—The primary mechanism is through the use of routing protocols like the Routing Information Protocol (RIP) and Open Shortest Path First (OSPF) protocol. Routes learned in this manner are often referred to as dynamic routes because they are updated periodically by the routing protocols to reflect the current conditions in the network. For more information on these protocols and their use, see Routing Information Protocol, page 121 and Open Shortest Path First (OSPF), page 137.
- **Static routes**—Alteon also learns networking routes through static routes. Static routes are manually entered into Alteon by an administrator. Although whole networks could be built upon static routes, they do not have the capacity to change without user intervention and therefore do not adequately represent the ever-changing reality of an enterprise network. It is because of this that static routes have an important but limited role in the enterprise network. Typically, static routes are used in situations when a protocol like RIP or OSPF cannot provide the information necessary to create connectivity between two nodes.

  For example, a node in a network that is running OSPF may need to know the route to a node in a network that is not running OSPF. OSPF would not provide information about either network to its counterpart. In this situation, a static route should be used to provide connectivity.

  Alteon supports both IPv4 and IPv6 static routes through the *Layer 3 Configuration* menu. Up to 128 IPv4 and 128 IPv6 static routes are supported.

IPv4 Static Routes

IPv4 static routes are used to support static connectivity to an IPv4 network.

To add an IPv4 static route

```
>> Main#/cfg/l3/route/ip4/add <destination> <mask> <gateway> [interface number]
```

Note: When adding an IPv4 static route, in most cases you do not have to specify an interface number. However, if you are using Firewall Load Balancing (FWLB) and you define two IP interfaces on the same subnet, where one IP interface has a subnet of the host which is also included in the subnet of the second interface, you must specify the interface.

To remove an IPv4 static route

```
>> Main#/cfg/l3/route/ip4/rem <destination> <mask>
```

The IPv4 static routes that are currently part of the configuration can be displayed using the **/cfg/l3/route/ip4/cur** command.

IPv6 Static Routes

IPv6 static routes support static connectivity to an IPv6 network. IPv6 static routes are conceptually identical to their IPv4 counterparts and only differ in the addressing format used. For information about IPv6 concepts and addressing formats, see IPv6, page 835.

IPv6 static routes are added using the **/cfg/l3/route/ip6/add** command, using the following syntax:

```
>> Main#/cfg/l3/route/ip6/add <destination> <prefix length> <next hop>
[interface number]
```

IPv6 static routes are removed from the switch using the **/cfg/l3/route/ip6/rem** command, using the following syntax:

```
>> Main#/cfg/l3/route/ip6/rem <destination> <prefix length> <next hop>
```

The IPv6 static routes that are currently part of the switch configuration can be displayed using the **/cfg/l3/route/ip6/cur** command.

Chapter 9 – Routing Information Protocol

This chapter discusses the Alteon implementation of the Routing Information Protocol (RIP). It includes the following topics:

In a routed environment, routers communicate with one another to keep track of available routes. Routers can learn about available routes dynamically using the Routing Information Protocol (RIP). Alteon supports RIP version 1 (RIPv1) and RIP version 2 (RIPv2) for exchanging TCP/IP route information with other routers.

Distance Vector Protocol

RIP is known as a distance vector protocol. The vector is the network number and next hop, and the distance is the cost associated with the network number. RIP identifies network reachability based on cost, and cost is defined as the hop count. One hop is considered to be the distance from one Alteon to the next, which is typically 1. This cost or hop count is known as the metric.

When Alteon receives a routing update that contains a new or changed destination network entry, it adds 1 to the metric value indicated in the update and enters the network in the routing table. The IP address of the sender is used as the next hop.

Stability

RIP includes a number of stability features that are common to many routing protocols. For example, RIP implements the split horizon and hold-down mechanisms to prevent incorrect routing information.

RIP prevents routing loops from continuing indefinitely by implementing a limit on the number of hops allowed in a path from the source to a destination. The maximum number of hops in a path is 15. The network destination network is considered unreachable if increasing the metric value by 1 causes the metric to be 16 (that is, infinity). This limits the maximum diameter of a RIP network to less than 16 hops.

RIP is often used in stub networks and in small autonomous systems that do not have many redundant paths.

Routing Updates

RIP sends routing update messages at regular intervals and when the network topology changes. Each router "advertises" routing information by sending a routing information update every 30 seconds. If a router does not receive an update from another router for 180 seconds, the routes provided by that router are declared invalid. After another 120 seconds without receiving an update for those routes, the routes are removed from the routing table and respective regular updates.

When a router receives a routing update that includes changes to an entry, it updates its routing table to reflect the new route. The metric value for the path is increased by 1, and the sender is indicated as the next hop. RIP routers maintain only the best route (the route with the lowest metric value) to a destination.

For details on configuring routing updates, see the explanation of the *Configuration* menu, Routing Information Protocol Configuration (**`/cfg/l3/rip`** command) in the *Alteon Application Switch Operating System Command Reference*.

RIP Versions

This section includes the following sub-sections:

RIP Version 1

RIP version 1 (RIPv1) uses broadcast User Datagram Protocol (UDP) data packets for the regular routing updates. The main disadvantage is that the routing updates do not carry subnet mask information. Therefore, the router cannot determine whether the route is a subnet route or a host route. It is of limited use after the introduction of RIPv2.

For more information about RIPv1 and RIPv2, refer to RFC 1058 and RFC 2453.

RIP Version 2

RIP version 2 (RIPv2) is the most popular and preferred configuration for most networks. RIPv2 expands the amount of useful information carried in RIP messages and provides a measure of security.

RIPv2 improves efficiency by using multicast UDP (address 224.0.0.9) data packets for regular routing updates. Subnet mask information is provided in the routing updates. A security option is added for authenticating routing updates by using a shared password. Alteon supports using clear text passwords for RIPv2.

RIPv2 supports the following enhancements to RIPv1:

- Variable length subnet masks for classless inter-domain routing.
- RIPv2 updates always include the next-hop router address.
- Routing updates can be sent to a multicast address.
- Routing updates can be authenticated using a simple password scheme.

For a detailed explanation of RIPv2, refer to RFC 1723 and RFC 2453.

RIP Version 2 in RIP Version 1 Compatibility Mode

Alteon allows for RIP version 2 (RIPv2) configuration and RIP version 1 (RIPv1) compatibility mode to use both RIPv2 and RIPv1 routers within a network. In this mode, the regular routing updates use broadcast UDP data packets to allow RIPv1 routers to receive those packets. With RIPv1 routers as recipients, the routing updates have to carry a natural or host mask. Therefore, it is not a recommended configuration for most network topologies.

Note: When using both RIPv1 and RIPv2 within a network, use a single subnet mask throughout the network.

RIP Features

Alteon provides the following features to support RIPv1 and RIPv2:

- Poison, page 123
- Triggered Updates, page 123
- Multicast, page 123
- Default, page 123
- Metric, page 123
- Authentication, page 123

Poison

Simple split horizon in the RIP scheme omits routes learned from one neighbor in updates sent to that neighbor. That is the most common configuration used in RIP network topology. Split horizon with poisoned reverse includes such routes in updates, but sets their metrics to 16. The disadvantage of using this feature is the increase of size in the routing updates. Therefore, Radware recommends disabling split horizon with poisoned reverse.

Triggered Updates

Triggered updates are an attempt to speed up convergence. When triggered updates are enabled, whenever a router changes the metric for a route, it sends update messages almost immediately without waiting for the regular update interval. Radware recommends enabling triggered updates.

Multicast

RIPv2 messages use the IP multicast address (224.0.0.9) for periodic broadcasts. Multicast RIPv2 announcements are not processed by RIPv1 routers.

To configure RIPv2 in RIPv1 compatibility mode, set multicast to **disable**.

Default

The RIP router can listen and supply a default route, usually represented as 0.0.0.0 in the routing table. When a router does not have an explicit route to a destination network in its routing table, it uses the default route to forward those packets.

Metric

The metric field contains a configurable value between 1 and 15 which specifies the current metric for the interface. The metric value typically indicates the total number of hops to the destination. The metric value of 16 represents an unreachable destination.

Authentication

RIPv2 authentication uses clear text passwords for authentication. If configured using an authentication password, then it is necessary to enter an authentication key value.

The following method is used to authenticate a RIP message:

- If the router is not configured to authenticate RIPv2 messages, then RIPv1 and unauthenticated RIPv2 messages are accepted. Authenticated RIPv2 messages are discarded.
- If the router is configured to authenticate RIPv2 messages, then RIPv1 messages and RIPv2 messages which pass authentication testing are accepted. Unauthenticated and failed authentication RIPv2 messages are discarded.

For maximum security, RIPv1 messages are ignored when authentication is enabled. If not, the routing information from authenticated messages is propagated by RIPv1 routers in an unauthenticated manner.

RIP Configuration Example

Note: A disabled RIP interface uses all the default values of the RIP, no matter how the RIP parameters are configured for that interface. RIP sends RIP regular updates to include an **up** interface, but not a **down** interface.

1. Add VLANs for routing interfaces.

`>> Main# cfg/l2/vlan 2/ena`	(Enable VLAN 2)
`>> VLAN 2# add 2`	(Add port 2 to VLAN 2)
`Port 2 is an UNTAGGED port and its current PVID is 1.` `Confirm changing PVID from 1 to 2 [y/n]: y`	
`>> VLAN 2# /cfg/l2/vlan 3/ena`	(Enable VLAN 3)
`>> VLAN 3# add 3`	(Add port EXT3 to VLAN 3)
`Port 3 is an UNTAGGED port and its current PVID is 1.` `Confirm changing PVID from 1 to 3 [y/n]: y`	

2. Add IP interfaces to VLANs.

`>> Main# cfg/l3/if 2/ena`	(Enable interface 2)
`>> IP Interface 2# addr 102.1.1.1`	(Define IP address for interface 2)
`>> IP Interface 2# vlan 2`	(Add interface 2 to VLAN 2)
`>> IP Interface 2# /cfg/l3/if 3/ena`	(Enable interface 3)
`>> IP Interface 3# addr 103.1.1.1`	(Define IP address for interface 3)
`>> IP Interface 3# vlan 3`	(Add interface 3 to VLAN 3)

3. Turn on RIP globally and enable RIP for each interface.

`>> Main# cfg/l3/rip on`	(Turn on RIP globally)
`>> Routing Information Protocol# if 2/ena`	(Enable RIP on IP interface 2)
`>> RIP Interface 2# ..`	
`>> Routing Information Protocol# if 3/ena`	(Enable RIP on IP interface 3)
`>> RIP Interface 3# apply`	(Apply your changes)
`>> RIP Interface 3# save`	(Save the configuration)

4. Use the **`/maint/route/dump`** command to check the current valid routes in the routing table.

   For those RIP-learned routes within the garbage collection period, routes phasing out of the routing table with metric 16, use the **`/info/l3/route/dump`** command. Locally configured static routes do not appear in the RIP routing table.

Chapter 10 – Border Gateway Protocol

The Border Gateway Protocol (BGP) enables routers on a network to share and advertise routing information with each other about the segments of the IP address space they can access within their network, and with routers on external networks. BGP allows you to decide what is the "best" route for a packet to take from your network to a destination on another network, rather than simply setting a default route from your border routers to your upstream providers. BGP is defined in RFC 1771.

Alteon can advertise its IP interfaces and virtual server IP addresses using BGP and take BGP feeds from as many as 16 BGP router peers. This allows more resilience and flexibility in balancing traffic from the Internet.

The following topics are addressed in this chapter:

BGP-based Global Server Load Balancing (GSLB) uses the Internet's routing protocols to localize content delivery to the most efficient and consistent site. For more information on BGP-based GSLB, see Using Border Gateway Protocol for GSLB, page 758.

Internal Routing Versus External Routing

To ensure effective processing of network traffic, every router on your network needs to be configured to correctly send a packet (directly or indirectly) to any other location or destination in your network. This is referred to as *internal routing,* and can be done with static routes or using active internal dynamic routing protocols, such as the Routing Information Protocol (RIP), RIPv2, and the Open Shortest Path First (OSPF) protocol.

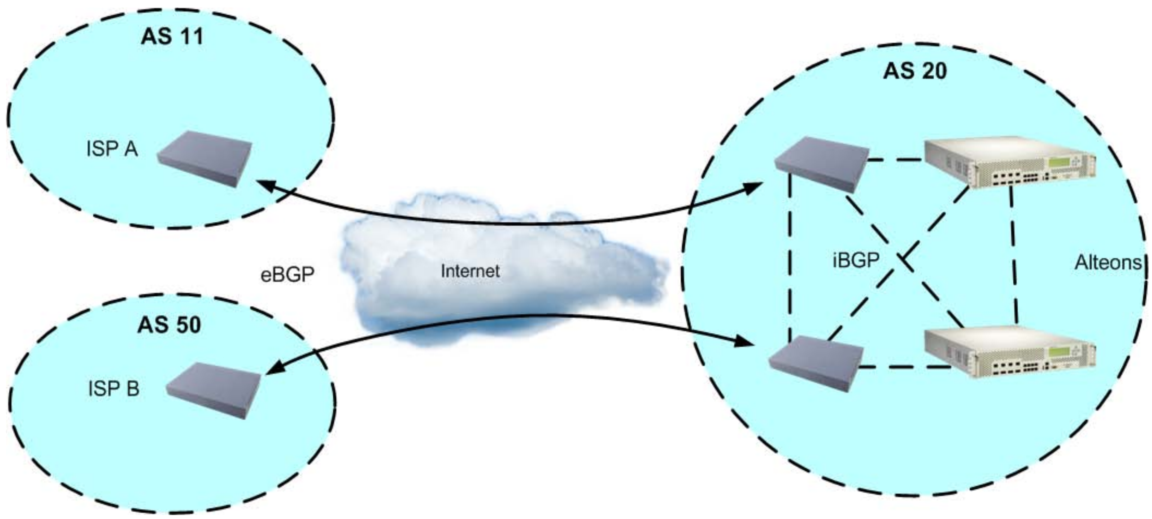
Static routes should have a higher degree of precedence than dynamic routing protocols. If the destination route is not in the route cache, then the packets are forwarded to the default gateway, which may be incorrect if a dynamic routing protocol is enabled.

It is also useful to expose the routes you can access in your network to routers outside your network (upstream providers, or *peers*). External networks (those outside your own) that are under the same administrative control, are referred to as *autonomous systems* (AS). Sharing of routing information between autonomous systems is known as *external routing*.

External BGP (eBGP) is used to exchange routes between different autonomous systems, while internal BGP (iBGP) is used to exchange routes within the same autonomous system. An iBGP is a type of internal routing protocol you can use to perform active routing inside your network. It also carries AS path information, which is important when you are an ISP or performing BGP transit.

The iBGP peers must be part of a fully meshed network, as shown in Figure 13 - Example Topology using the Border Gateway Protocol (BGP), page 126:

Figure 13: Example Topology using the Border Gateway Protocol (BGP)



Typically, an AS has one or more border routers (that is, peer routers that exchange routes with other ASs) and an internal routing scheme that enables routers in that AS to reach every other router and destination within that AS. When Alteon *advertises* routes to border routers on other autonomous systems, it is effectively committing to carry data to the IP space represented in the route being advertised. For example, if Alteon advertises 192.204.4.0/24, it is declaring that if another router sends it data destined for any address in 192.204.4.0/24, Alteon knows how to carry that data to its destination.

Forming BGP Peer Routers

Two BGP routers become peers, or neighbors, once you establish a TCP connection between them. For each new route, if a peer is configured to connect to that route (for example, if a peer is configured to receive static routes and the new route is static), an update message is sent to that peer containing the new route. For each route removed from the routing table, if the route has already been sent to a peer, an update message containing the route to withdraw is sent to that peer.

For each Internet host, your system must send a packet to that host, and that host must have a path back to your system. Whatever system provides Internet connectivity to that host must have a path to your system. Ultimately, the system providing the Internet connectivity must "hear a route" which covers the section of the IP space your system is using. Otherwise, your system will not have connectivity to the host in question.

Route Maps

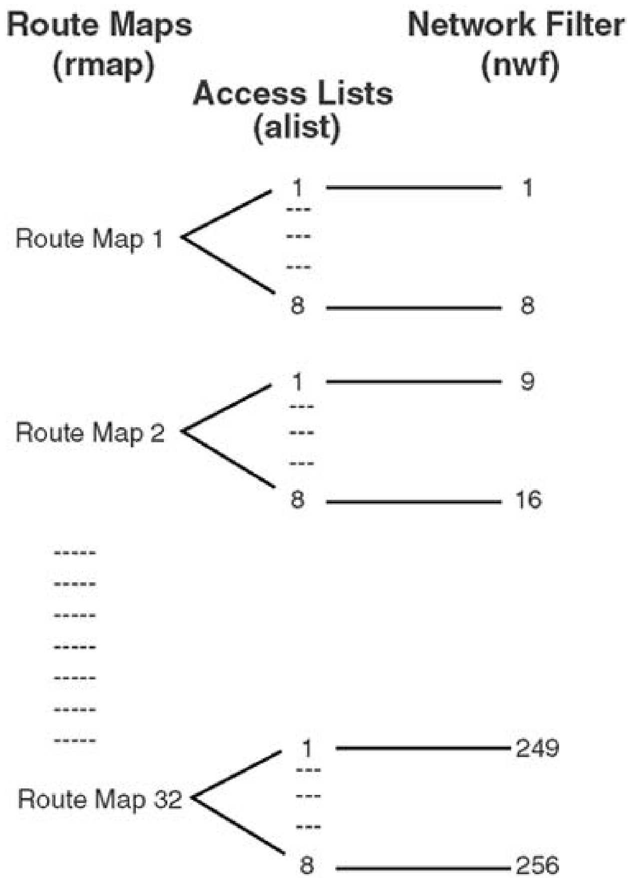
A route map is used to control and modify routing information. Route maps define conditions for redistributing routes from one routing protocol to another, or controlling routing information when injecting it in and out of BGP. Route maps are used by OSPF only for redistributing routes. For example, a route map is used to set a preference value for a specific route from a peer router and another preference value for all other routes learned via the same peer router. The following command is used to define a route map:

```
>> # /cfg/l3/rmap 1                    (Select a route map)
```

A route map lets you match attributes, such as metric, network address, and the AS number. It also lets you overwrite the local preference metric and to append the AS number in the AS route. For more information, see BGP Failover Configuration, page 131.

Alteon lets you configure up to 32 route maps. Each route map can have up to eight access lists. Each access list consists of a network filter. A network filter defines an IP address and subnet mask of the network that you want to include in the filter. Figure 14 - Relationship Between Route Maps, Access Lists, and Network Filters, page 127 illustrates the relationship between route maps, access lists and network filters.

Figure 14: Relationship Between Route Maps, Access Lists, and Network Filters



Incoming and Outgoing Route Maps

You can have two types of route maps: incoming and outgoing. A BGP peer router can be configured to support up to eight route maps in the incoming route map list and outgoing route map list.

If a route map is not configured in the incoming route map list, the router imports all BGP updates. If a route map is configured in the incoming route map list, the router ignores all unmatched incoming updates.

Route maps in an outgoing route map list behave similar to route maps in an incoming route map list. If a route map is not configured in the outgoing route map list, all routes are advertised or permitted. If a route map is configured in the outgoing route map list, matched routes are advertised and unmatched routes are ignored.

Precedence

You can set a priority to a route map by specifying a precedence value with the following command:

>> /cfg/l3/rmap <x> /pre	(Specify a precedence)

The lower the value, the higher the precedence. If two route maps have the same precedence value, the lower number has higher precedence.

Configuration Overview

To configure route maps

1. Define the network filter.

>> # /cfg/l3/nwf 1	(Specify a network filter number)
>> IP Network Filter 1# addr <IP address>	(Specify network address)
>> IP Network Filter 1# mask <IP mask>	(Specify network mask)
>> IP Network Filter 1# ena	(Enable network filter)

   Enter a filter number from 1 to 256. Specify the IP address and subnet mask of the network that you want to match. Enable the network filter. You can distribute up to 256 network filters among 32 route maps each containing eight access lists.

2. Optionally, define the criteria for the access list and enable it.

   Specify the access list and associate the network filter number configured in step 1.

>> # /cfb/13/rmap 1	(Specify a route map number)
>> IP Route Map 1# alist 1	(Specify the access list number)
>> IP Access List 1# nwf 1	(Specify the network filter number)
>> IP Access List 1# metric	(Define a metric)
>> IP Access List 1# action deny	(Specify action for the access list)
>> IP Access List 1# ena	(Enable the access list)

   This step and step 3 are optional, depending on the criteria that you want to match. In this step, the network filter number is used to match the subnets defined in the network filter. In step 3, the autonomous system number is used to match the subnets. Alternately, you can use both step 2 and step 3 criteria (access list [network filter] and access path [AS filter]) to configure the route maps.

3. Optionally, configure the attributes in the AS filter menu.

>> # cfg/13/rmap 1/aspath	(Specify the attributes in the filter)
>> AS Filter 1# as 1	(Specify the AS number)
>> AS Filter 1# action deny	(Specify the action for the filter)
>> AS Filter 1# ena	(Enable the AS filter)

4. Set up the BGP attributes.

   If you want to overwrite the attributes that the peer router is sending, define the following BGP attributes:

   — Specify the AS numbers that you want to prepend to a matched route and the local preference for the matched route.

   — Specify the metric for the matched route.

`>> # /cfg/l3/rmap 1`	(Specify a route map number)
`>> IP Route Map 1# ap 1`	(Specify the AS numbers to prepend)
`>> IP Route Map 1# lp`	(Specify the local preference)
`>> IP Route Map 1# met`	(Specify the metric)

5. Enable the route map.

```
>> # /cfg/l3/rmap 1/en
```

6. Assign the route map to a peer router. Select the peer router and then add the route map to one of the following:

   — Incoming route map list:

```
>> # /cfg/l3/bgp/peer 1/addi
```

   — Outgoing route map list:

```
>> # /cfg/l3/bgp/peer 1/addo
```

Aggregating Routes

Aggregation is the process of combining several different routes in such a way that a single route can be advertised, minimizing the size of the routing table. You can configure aggregate routes in BGP either by redistributing an aggregate route into BGP or by creating an aggregate entry in the BGP routing table.

When a subnet is redistributed from an Interior Gateway Protocol (IGP) into BGP, only the network route is injected into the BGP table. By default, this automatic summarization is disabled.

To define the route to aggregate

`>> # /cfg/l3/bgp`	(Specify BGP)
`>> Border Gateway Protocol# aggr 1`	(Specify aggregate list number)
`>> BGP aggr 1 # addr`	(Enter aggregation network address)
`>> BGP aggr 1 # mask`	(Enter aggregation network mask)
`>> BGP aggr 1 # ena`	(Enable aggregation)

For an example of creating a BGP aggregate route, see Default Redistribution and Route Aggregation Example, page 134.

Redistributing Routes

In addition to running multiple routing protocols simultaneously, Alteon can redistribute information from one routing protocol to another. For example, you can instruct Alteon to use BGP to readvertise static routes. This applies to all of the IP-based routing protocols.

You can also conditionally control the redistribution of routes between routing domains by defining a method known as route maps between the two domains. For more information on route maps, see Route Maps, page 126. Redistributing routes is another way of providing policy control over whether to export OSPF routes, fixed routes, static routes, and virtual IP address routes. For an example configuration, see Default Redistribution and Route Aggregation Example, page 134.

Default routes can be configured using the following methods:

- Import
- **Originate**—The router sends a default route to peers even though it does not have any default routes in its routing table.
- **Redistribute**—Default routes are either configured through the default gateway or learned via other protocols and redistributed to peer routers. If the default routes are from the default gateway, enable the static routes because default routes from the default gateway are static routes. Similarly, if the routes are learned from another routing protocol, enable that protocol for redistribution.
- None

BGP Attributes

The following two BGP attributes are discussed in this section:

- Local Preference Attribute, page 130
- Metric (Multi-Exit Discriminator) Attribute, page 130

Local Preference Attribute

When there are multiple paths to the same destination, the local preference attribute indicates the preferred path. The path with the higher preference is preferred (the default value of the local preference attribute is 100). Unlike the weight attribute, which is only relevant to the local router, the local preference attribute is part of the routing update and is exchanged among routers in the same AS.

The local preference attribute can be set in one of two ways:

- `/cfg/l3/bgp/pref`—This command uses the BGP default local preference method, affecting the outbound direction only.
- `/cfg/l3/rmap/lp`—This command uses the route map local preference method, which affects both inbound and outbound directions.

Metric (Multi-Exit Discriminator) Attribute

This attribute is a hint to external neighbors about the preferred path into an AS when there are multiple entry points. A lower metric value is preferred over a higher metric value. The default value of the metric attribute is 0.

Unlike local preference, the metric attribute is exchanged between ASs. However, a metric attribute that comes into an AS does not leave the AS.

When an update enters the AS with a certain metric value, that value is used for decision making within the AS. When BGP sends that update to another AS, the metric is reset to 0.

Unless otherwise specified, the router compares metric attributes for paths from external neighbors that are in the same AS.

Selecting Route Paths in BGP

BGP selects only one path as the best path. It does not rely on metrics attributes to determine the best path. When the same network is learned via more than one BGP peer, BGP uses its policy for selecting the best route to that network. The BGP implementation in Alteon uses the following criteria to select a path when the same route is received from multiple peers:

1. Local fixed and static routes are preferred over learned routes.
2. With iBGP peers, routes with higher local preference values are selected.
3. In the case of multiple routes of equal preference, the route with lower AS path weight is selected, using the following algorithm:

   AS path weight = 128 x AS path length (number of autonomous systems transversed)
4. In the case of equal weight and routes learned from peers that reside in the same AS, the lower metric is selected.

   A route with a metric is preferred over a route without a metric.
5. The lower cost to the next hop of routes is selected.
6. In the case of equal cost, the eBGP route is preferred over iBGP.
7. If all routes are from eBGP, the route with the lower router ID is selected.

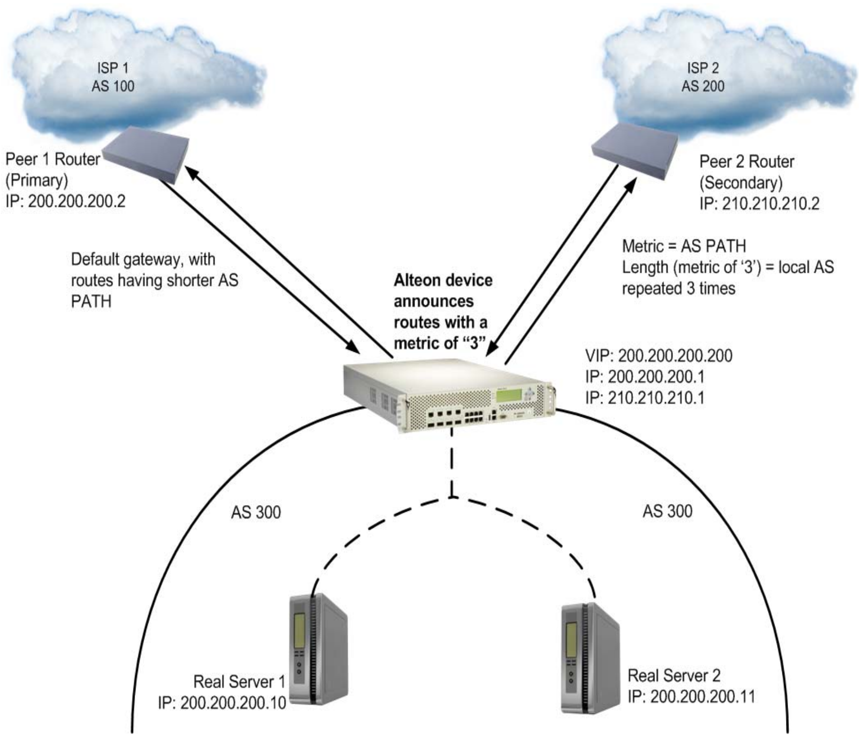
   When the path is selected, BGP puts the selected path in its routing table and propagates the path to its neighbors.

BGP Failover Configuration

This section describes an example configuration to create redundant default gateways for Alteons at a Web Host/ISP site, eliminating the possibility, should one gateway go down, that requests are forwarded to an upstream router unknown to Alteon.

As shown in Figure 15 - Example BGP Failover Configuration, page 132, Alteon is connected to ISP 1 and ISP 2. The customer negotiates with both ISPs to allow Alteon to use the ISPs' peer routers as default gateways. The ISP peer routers announce themselves as default gateways to Alteon.

Figure 15: Example BGP Failover Configuration



On Alteon, one peer router (the secondary one) is configured with a longer AS path than the other, so that the peer with the shorter AS path will be seen by Alteon as the primary default gateway. ISP 2, the secondary peer, is configured with a metric of *3*, appearing to Alteon to be three router *hops* away.

Example

1. Configure Alteon as you normally would for Server Load Balancing (SLB).
   — Assign an IP address to each of the real servers in the server pool.
   — Define each real server.
   — Define a real server group.
   — Define a virtual server.
   — Define the port configuration.

   For more information about SLB configuration, see Server Load Balancing, page 165.
2. Define the VLANs.

   For simplicity, both default gateways are configured on the same VLAN in this example. The gateways could be in the same VLAN or different VLANs.

`>> # /cfg/l2/vlan 1`	(Select VLAN 1)
`>> vlan 1# add <port number>`	(Add a port to the VLAN membership)

3. Define the IP interfaces.

Alteon needs an IP interface for each default gateway to which it is connected. Each interface needs to be placed in the appropriate VLAN. These interfaces are used as the primary and secondary default gateways for Alteon.

`>> /cfg/l3/arp/rearp 10`	(Set the re-ARP period for interface to 10)
`>> IP# /cfg/l3/metric strict`	(Set metric for default gateway)
`>> IP# if 1`	(Select default gateway interface 1)
`>> IP Interface 1# ena`	(Enable Interface 1)
`>> IP Interface 1# addr 200.200.200.1`	(Configure IP address of Interface 1)
`>> IP Interface 1# mask 255.255.255.0`	(Configure IP subnet address mask)
`>> IP Interface 1# /cfg/l3/if 2`	(Select default gateway interface 2)
`>> IP Interface 2# ena`	(Enable Interface 2)
`>> IP Interface 2# addr 210.210.210.1`	(Configure IP address of Interface 2)
`>> IP Interface 2# mask 255.255.255.0`	(Configure IP subnet address mask)

4. IP forwarding is enabled by default and is used for VLAN-to-VLAN (non-BGP) routing. Make sure IP forwarding is enabled if the default gateways are on different subnets or if Alteon is connected to different subnets and those subnets need to communicate through Alteon.

```
>> /cfg/l3/frwd/on
```

Note: To help eliminate the possibility for a Denial of Service (DoS) attack, the forwarding of directed broadcasts is disabled by default.

5. Globally turn on BGP.

```
>> # /cfg/l3/bgp/on
```

6. Configure BGP peer router 1 and 2. Peer 1 is the primary gateway router. Peer 2 is configured with a metric of 3. The **`metric`** option is key to ensuring gateway traffic is directed to peer 1, as it makes peer 2 appear to be three router hops away from Alteon. Therefore, Alteon should never use it unless peer 1 goes down.

`>> # /cfg/l3/bgp/peer 1`	(Select BGP peer router 1)
`>> BGP Peer 1# ena`	(Enable this peer configuration)
`>> BGP Peer 1# addr 200.200.200.2`	(Set IP address for peer router 1)
`>> BGP Peer 1# if 200.200.200.1`	(Set IP interface for peer router 1)
`>> BGP Peer 1# ras 100`	(Set remote AS number)
`>> BGP Peer 1# /cfg/l3/bgp/peer 2`	(Select BGP peer router 2)
`>> BGP Peer 2# ena`	(Enable this peer configuration)
`>> BGP Peer 2# addr 210.210.210.2`	(Set IP address for peer router 2)
`>> BGP Peer 2# if 210.210.210.1`	(Set IP interface for peer router 2)
`>> BGP Peer 2# ras 200`	(Set remote AS number)
`>> BGP Peer 2# redist/metric 3`	(Set AS path length to 3 router hops)

The metric command in the *Peer* menu causes Alteon to create an AS path of *3* when advertising via BGP.

7. Apply and save your configuration changes.

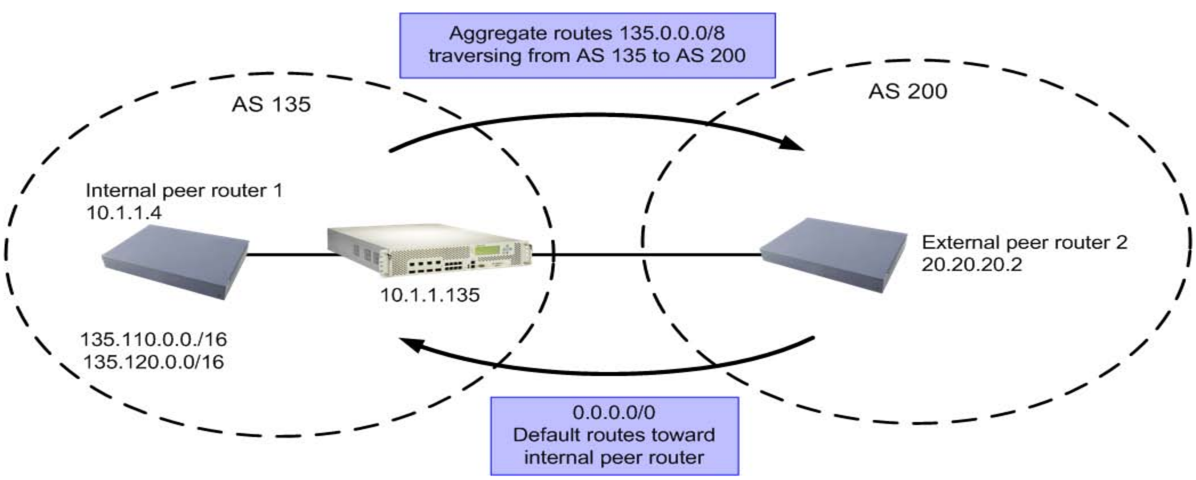
`>> BGP Peer 2# apply`	(Make your changes active)
`>> save`	(Save for restore after reboot)

Default Redistribution and Route Aggregation Example

This example illustrates how to configure Alteon to redistribute information from one routing protocol to another and create an aggregate route entry in the BGP routing table to minimize the size of the routing table.

As illustrated in Figure 16 - Default Redistribution and Route Aggregation Example, page 134, there are two peer routers: an internal and an external peer router. Alteon is configured to redistribute the default routes from AS 200 to AS 135. At the same time, route aggregation condenses the number of routes traversing from AS 135 to AS 200.

Figure 16: Default Redistribution and Route Aggregation Example



Example

1. Configure the IP interface.
2. Configure the AS number (AS 135) and router ID number (10.1.1.135).

   The router ID number must be a unique number and does not have to be an IP address. However, for convenience, this ID is typically one of IP addresses assigned in IP interfaces.

`>> # /cfg/l3/bgp`	(Select the BGP menu)
`>> Border Gateway Protocol# as 135`	(Specify an AS number)
`>> Border Gateway Protocol# as /cfg/l3/rtrid 10.1.1.135`	(Specify the router ID number)

3. Configure internal peer router 1 and external peer router 2.

`>> # /cfg/l3/bgp/peer 1`	(Select internal peer router 1)
`>> BGP Peer 1# ena`	(Enable this peer configuration)
`>> BGP Peer 1# addr 10.1.1.4`	(Set IP address for peer router 1)
`>> BGP Peer 1# ras 135`	(Set remote AS number)
`>> BGP Peer 1# /cfg/l3/bgp/peer 2`	(Select external peer router 2)
`>> BGP Peer 2# ena`	(Enable this peer configuration)
`>> BGP Peer 2# addr 20.20.20.2`	(Set IP address for peer router 2)
`>> BGP Peer 2# ras 200`	(Set remote AS number)

4. Configure redistribution for peer 1.

`>> # /cfg/l3/bgp/peer 1/redist`	(Select redistribute)
`>> BGP Peer 1# default redistribute`	(Set default to redistribute)
`>> BGP Peer 1# fixed ena`	(Enable fixed routes)

5. Configure aggregation policy control. Configure the routes that you want aggregated.

`>> # /cfg/l3/bgp/aggr 1`	(Set aggregation number)
`>> BGP Aggr 1# addr 135.0.0.0`	(Add IP address to aggregate 1)
`>> BGP Aggr 1# mask 255.0.0.0`	(Add IP mask to aggregate 1)
`>> BGP Aggr 1# ena`	(Enable route aggregation)

6. Apply and save the configuration.

```
>> # apply
>> # save
```

Chapter 11 – Open Shortest Path First (OSPF)

Alteon supports versions 2 and 3 of the Open Shortest Path First (OSPF) routing protocol.

The Alteon OSPF version 2 implementation conforms to the specifications detailed in Internet RFC 1583.

The Alteon OSPF version 3 implementation conforms to the specifications detailed in Internet RFC 2740.

The following topics are addressed in this chapter:

- OSPF Overview, page 137—This section explains OSPF concepts, such as types of OSPF areas, types of routing devices, neighbors, adjacencies, link state database, authentication, and internal versus external routing.
- OSPF Implementation, page 141—This section describes how OSPF is implemented, such as configuration parameters, electing the designated router, summarizing routes, and defining route maps.
- OSPF Configuration Examples, page 150—This section provides step-by-step instructions on configuring four different configuration examples:
  - Example 1: Simple OSPF Domain, page 151
  - Example 2: Virtual Links, page 152
  - Example 3: Summarizing Routes, page 156
  - Example 4: Host Routes, page 158

Note: CLI command paths in this chapter reflect OSPF version 2. For OSPF version 3 paths, it is sufficient in most cases to replace the **ospf** parameter with **ospfv3**. For example:

- OSPF version 2 CLI path:

```
>> # /cfg/l3/ospf/aindex
```

- Corresponding OSPF version 3 CLI path:

```
>> # /cfg/l3/ospfv3/aindex
```

OSPF Overview

OSPF is designed for routing traffic within a single IP domain called an Autonomous System (AS). The AS can be divided into smaller logical units known as *areas*.

All routing devices maintain link information in their own Link State Database (LSDB). The LSDB for all routing devices within an area is identical but is not exchanged between different areas. Only routing updates are exchanged between areas, thereby significantly reducing the overhead for maintaining routing information on a large, dynamic network.

The following key OSPF concepts are described in this section:

- Equal Cost Multipath Routing Support, page 138
- Types of OSPF Areas, page 138
- Types of OSPF Routing Devices, page 139
- Neighbors and Adjacencies, page 139
- The Link-State Database, page 140

Equal Cost Multipath Routing Support

Alteon supports equal-cost multipath (ECMP), which is a routing technique for routing packets along multiple paths of equal cost. The routing table contains multiple next hops for any given destination. The router load balances packets along the multiple next hops.

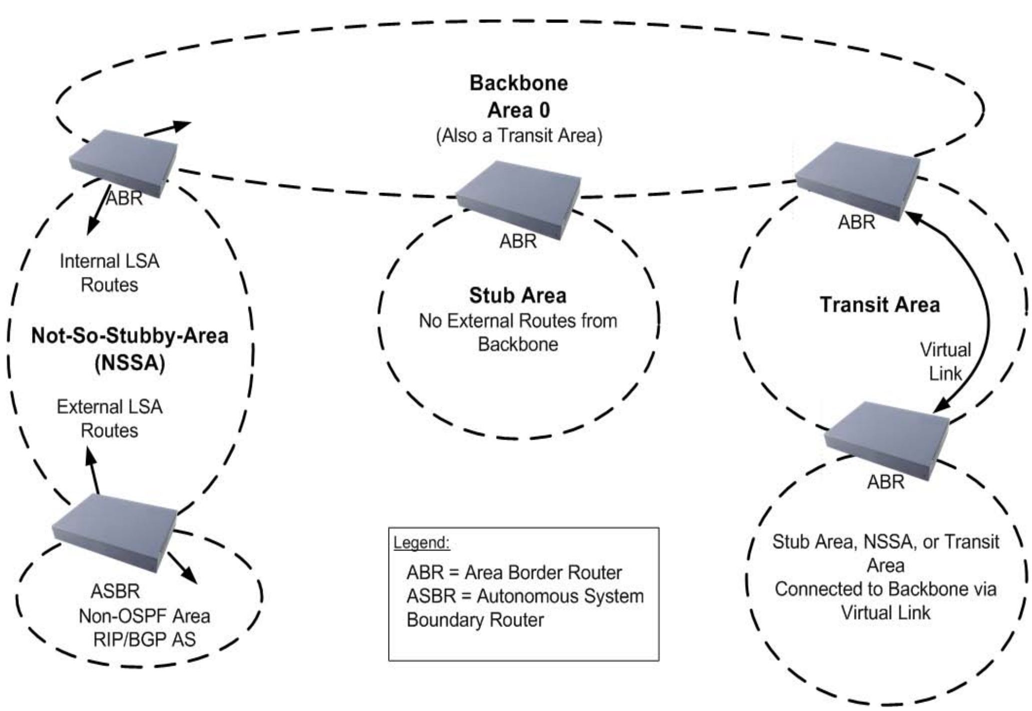
Types of OSPF Areas

An AS can be broken into logical units known as *areas*. In any AS with multiple areas, one area must be designated as area 0, known as the *backbone*. The backbone acts as the central OSPF area. All other areas in the AS must be connected to the backbone. Areas inject summary routing information into the backbone, which then distributes it to other areas as needed.

As shown in Figure 17 - OSPF Areas, page 138, OSPF defines the following types of areas:

- **Stub Area**—An area that is connected to only one other area. External route information is not distributed into stub areas.
- **Not-So-Stubby-Area (NSSA)**—An area similar to a stub area with additional capabilities. Routes originating from within the NSSA can be propagated to adjacent transit and backbone areas. External routes from outside the AS can be advertised within the NSSA but are not distributed into other areas.
- **Transit Area**—An area that allows area summary information to be exchanged between routing devices. The backbone (area 0), any area that contains a virtual link to connect two areas, and any area that is not a stub area or an NSSA, are considered transit areas.

Figure 17: OSPF Areas

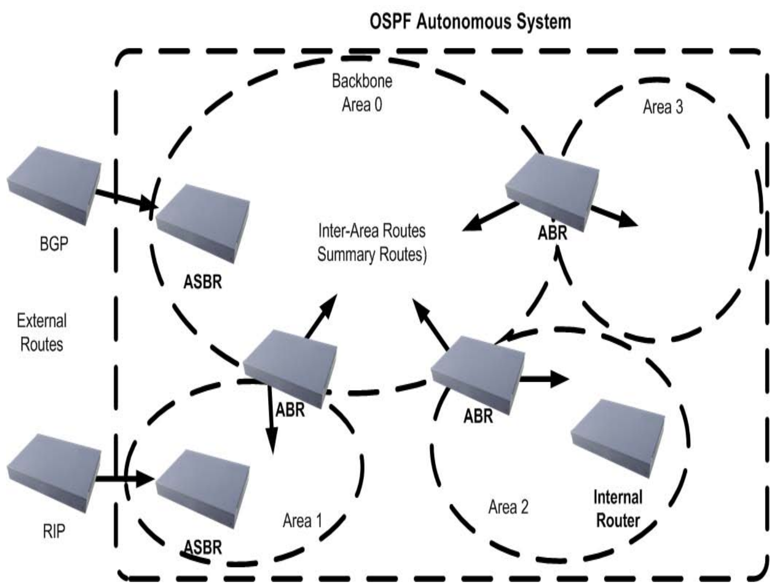


Types of OSPF Routing Devices

As shown in Figure 18 - OSPF Routing Device Types, page 139, OSPF uses the following types of routing devices:

- **Internal Router (IR)**—A router that has all of its interfaces within the same area. IRs maintain LSDBs identical to those of other routing devices within the local area.
- **Area Border Router (ABR)**—A router that has interfaces in multiple areas. ABRs maintain one LSDB for each connected area and disseminate routing information between areas.
- **Autonomous System Boundary Router (ASBR)**—A router that acts as a gateway between the OSPF domain and non-OSPF domains, such as RIP, BGP, and static routes.

Figure 18: OSPF Routing Device Types



Neighbors and Adjacencies

In areas with two or more routing devices, *neighbors* and *adjacencies* are formed.

Neighbors are routing devices that maintain information about each others' health. To establish neighbor relationships, routing devices periodically send hello packets on each of their interfaces. All routing devices that share a common network segment, appear in the same area, and have the same health parameters (*hello* and *dead* intervals), and authentication parameters respond to each other's hello packets and become neighbors. Neighbors continue to send periodic hello packets to advertise their health to neighbors. In turn, they listen to hello packets to determine the health of their neighbors and to establish contact with new neighbors.

The hello process is used for electing one of the neighbors as the area's Designated Router (DR) and one as the area's Backup Designated Router (BDR). The DR is adjacent to all other neighbors and acts as the central contact for database exchanges. Each neighbor sends its database information to the DR, which relays the information to the other neighbors.

The BDR is adjacent to all other neighbors (including the DR). Each neighbor sends its database information to the BDR just as with the DR, but the BDR merely stores this data and does not distribute it. If the DR fails, the BDR takes over the task of distributing database information to the other neighbors.

Note: The Alteon IPv6 component runs OSPFv3 adjacency per VLAN and not per Layer 3 interface. This is because OSPFv3 requires a link local address, which is available with a VLAN, but not with a Layer 3 interface.

The Link-State Database

OSPF is a link-state routing protocol. A *link* represents an interface (or routable path) from the routing device. By establishing an adjacency with the DR, each routing device in an OSPF area maintains an identical Link-State Database (LSDB) describing the network topology for its area.

Each routing device transmits a Link-State Advertisement (LSA) on each of its interfaces. LSAs are entered into the LSDB of each routing device. OSPF uses *flooding* to distribute LSAs between routing devices.

When LSAs result in changes to the routing device's LSDB, the routing device forwards the changes to the adjacent neighbors (the DR and BDR) for distribution to the other neighbors.

OSPF routing updates occur only when changes occur, instead of periodically. For each new route, if an adjacency is interested in that route (for example, if configured to receive static routes and the new route is indeed static), an update message containing the new route is sent to the adjacency. For each route removed from the routing table, if the route has already been sent to an adjacency, an update message containing the route to withdraw is sent.

The Shortest Path First Tree

The routing devices use a link-state algorithm (Dijkstra's algorithm) to calculate the shortest path to all known destinations, based on the cumulative *cost* required to reach the destination.

The cost of an individual interface in OSPF is an indication of the overhead required to send packets across it. The cost is inversely proportional to the bandwidth of the interface. A lower cost indicates a higher bandwidth.

Internal versus External Routing

To ensure effective processing of network traffic, every routing device on your network needs to be configured to correctly send a packet (directly or indirectly) to any other location or destination in your network. This is referred to as *internal routing,* and can be done with static routes or using active internal routing protocols, such as the Routing Information Protocol (RIP), RIPv2, and the Open Shortest Path First (OSPF) protocol.

It is also useful to expose the routes you can access outside your network (upstream providers or *peers*) about the routes you have access to in your network. Sharing of routing information between autonomous systems is known as *external routing*.

Typically, an AS has one or more border routers (peer routers that exchange routes with other OSPF networks) as well as an internal routing system enabling every router in that AS to reach every other router and destination within that AS.

When a routing device *advertises* routes to boundary routers on other autonomous systems, it is effectively committing to carry data to the IP space represented in the route being advertised. For example, if the routing device advertises 192.204.4.0/24, it is declaring that if another router sends data destined for any address in the 192.204.4.0/24 range, it will carry that data to its destination.

OSPF Implementation

Alteon supports a single instance of OSPF and up to 4 K routes on the network. The following sections describe Alteon OSPF implementation:

- Defining Areas, page 141
- Interface Cost, page 143
- Electing the Designated Router and Backup, page 143
- Summarizing Routes, page 143
- Default Routes, page 143
- Virtual Links, page 145
- Router ID, page 145
- Authentication, page 146
- Host Routes for Load Balancing, page 148
- Redistributing Routes into OSPF, page 148

Defining Areas

If you are configuring multiple areas in your OSPF domain, one of the areas must be designated as area 0, known as the *backbone*. The backbone is the central OSPF area and is usually physically connected to all other areas. The areas inject routing information into the backbone which, in turn, disseminates the information into other areas.

Since the backbone connects the areas in your network, it must be a contiguous area. If the backbone is partitioned (possibly as a result of joining separate OSPF networks), parts of the AS will be unreachable, and you will need to configure *virtual links* to reconnect the partitioned areas (see Virtual Links, page 145).

Up to three OSPF areas can be connected to Alteon. To configure an area, the OSPF number must be defined and then attached to a network interface on Alteon. The full process is explained in this section.

An OSPF area is defined by assigning two pieces of information—an *area index* and an *area ID*. The command to define an OSPF area is as follows:

```
>> # /cfg/l3/ospf/aindex <area index> /areaid <n.n.n.n>
```

Note: The **aindex** option is an arbitrary index used only by Alteon, and does not represent the actual OSPF area number. The actual OSPF area number is defined in the **areaid** portion of the command.

Assigning the Area Index

The **aindex <area index>** option is an arbitrary index (0 to 2) used only by Alteon. This index does not necessarily represent the OSPF area number, though for configuration simplicity, it should where possible.

For example, both of the following sets of commands define OSPF area 0 (the backbone) and area 1 because that information is held in the area ID portion of the command. However, the first set of commands is easier to maintain because the arbitrary area indexes agree with the area IDs:

- Area index and area ID agree

`/cfg/l3/ospf/aindex 0/areaid 0.0.0.0`	(Use index 0 to set area 0 in ID octet format)
`/cfg/l3/ospf/aindex 1/areaid 0.0.0.1`	(Use index 1 to set area 1 in ID octet format)

- Area index set to an arbitrary value

`/cfg/l3/ospf/aindex 1/areaid 0.0.0.0`	(Use index 1 to set area 0 in ID octet format)
`/cfg/l3/ospf/aindex 2/areaid 0.0.0.1`	(Use index 2 to set area 1 in ID octet format)

Using the Area ID to Assign the OSPF Area Number

The OSPF area number is defined in the **areaid <IP address>** option. The octet format is used in order to be compatible with two different notation systems used by other OSPF network vendors. There are two valid ways to designate an area ID:

- **Placing the area number in the last octet (0.0.0.*n*)**—Most common OSPF vendors express the area ID number as a single number. For example, the Cisco IOS-based router command **network 1.1.1.0 0.0.0.255 area 1** defines the area number simply as **area 1.** In Alteon, using the last octet in the area ID, **area 1** is equivalent to **areaid 0.0.0.1**.
- **Multi-octet (*IP address*)**—Some OSPF vendors express the area ID number in multi-octet format. For example, **area 2.2.2.2** represents OSPF area 2, and can be specified directly in Alteon as **areaid 2.2.2.2**.

Note: Although both types of area ID formats are supported, ensure that the area IDs are in the same format throughout an area.

Attaching an Area to a Network

Once an OSPF area has been defined, it must be associated with a network. To attach the area to a network, you must assign the OSPF area index to an IP interface that participates in the area. The format for the command is as follows:

```
>> # /cfg/l3/ospf/if <interface number> /aindex <area index>
```

Example

The following commands could be used to configure IP interface 14 for a presence on the 10.10.10.1/24 network, to define OSPF area 1, and to attach the area to the network:

`>> # /cfg/l3/if 14`	(Select menu for IP interface 14)
`>> IP Interface 14# addr 10.10.10.1`	(Define IP address on backbone network)
`>> IP Interface 14# mask 255.255.255.0`	(Define IP mask on backbone)
`>> IP Interface 14# ena`	(Enable IP interface 14)
`>> IP Interface 14# /cfg/l3/ospf/aindex 1`	(Select menu for area index 1)
`>> OSPF Area (index) 1 # areaid 0.0.0.1`	(Define area ID as OSPF area 1)
`>> OSPF Area (index) 1 # ena`	(Enable area index 1)
`>> OSPF Area (index) 1 # /cfg/l3/ospf/if 14`	(Select OSPF menu for interface 14)

`>> OSPF Interface 14# aindex 1`	(Attach area to network on interface 14)
`>> OSPF Interface 14# enable`	(Enable interface 14 for area index 1)

Interface Cost

The OSPF link-state algorithm (Dijkstra's algorithm) places each routing device at the root of a tree and determines the cumulative *cost* required to reach each destination. Usually, the cost is inversely proportional to the bandwidth of the interface. A low cost indicates high bandwidth. You can manually enter the cost for the output route with the following command:

```
>> # /cfg/l3/0spf/if <OSPF interface number> /cost <cost value (1-65535)
```

Electing the Designated Router and Backup

In any area with more than two routing devices, a Designated Router (DR) is elected as the central contact for database exchanges among neighbors, and a Backup Designated Router (BDR) is elected in case the DR fails.

DR and BDR elections are made through the hello process. The election can be influenced by assigning a priority value to the OSPF interfaces with the following command:

```
>> # /cfg/l3/ospf/if <OSPF interface number> /prio <priority value (0-255)>
```

A priority value of 255 is the highest, and 1 is the lowest. A priority value of 0 specifies that the interface cannot be used as a DR or BDR. In case of a tie, the routing device with the highest router ID wins.

Summarizing Routes

Route summarization condenses routing information. Without summarization, each routing device in an OSPF network would retain a route to every subnet in the network. With summarization, routing devices can reduce some sets of routes to a single advertisement, reducing both the load on the routing device and the perceived complexity of the network. The importance of route summarization increases with network size.

Summary routes can be defined for up to 16 IP address ranges using the following command:

```
>> # /cfg/l3/ospf/range <range number> /addr <IP address> /mask <mask>
```

- *range number* is a number 1 to 16
- *IP address* is the base IP address for the range
- *mask* is the IP address mask for the range

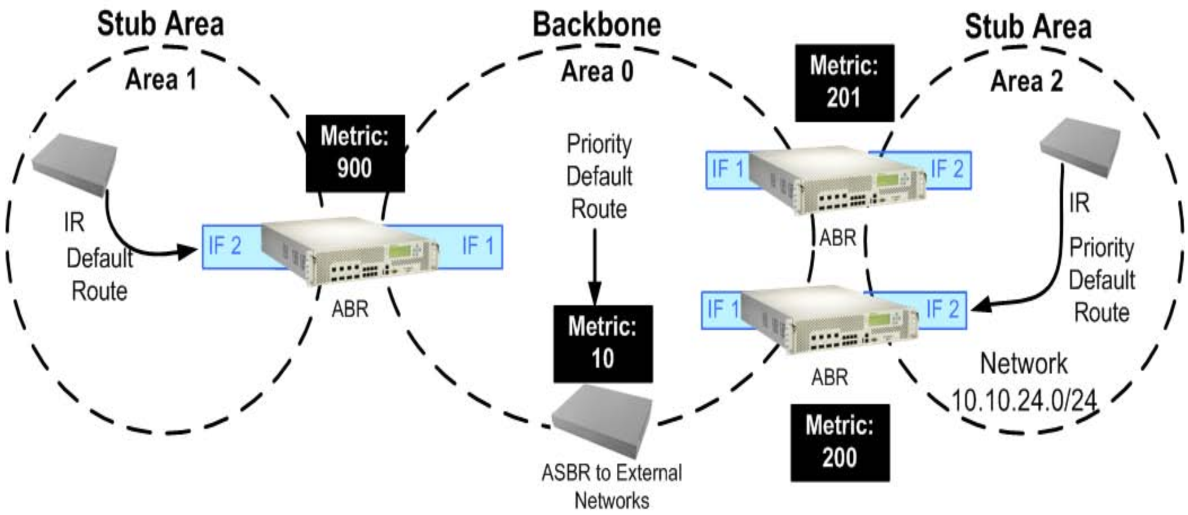
For a detailed configuration example, see Example 3: Summarizing Routes, page 156.

Default Routes

When an OSPF routing device encounters traffic for a destination address it does not recognize, it forwards that traffic along the *default route*. Typically, the default route leads upstream toward the backbone until it reaches the intended area or an external router.

Each Alteon acting as an ABR inserts a default route into each attached area. In simple OSPF stub areas or NSSAs with only one ABR leading upstream (see Area 1 in Figure 19 - Default Routes Example, page 144), any traffic for IP address destinations outside the area is forwarded to Alteon's IP interface, and then into the connected transit area (usually the backbone). Since this is automatic, no further configuration is required for such areas.

Figure 19: Default Routes Example



In more complex OSPF areas with multiple ABRs or ASBRs (such as area 0 and area 2 in Figure 19 - Default Routes Example, page 144), there are multiple routes leading from the area. In such areas, traffic for unrecognized destinations cannot determine which route leads upstream without further configuration.

To resolve the situation and select one default route among multiple choices in an area, you can manually configure a metric value on each ABR. The metric assigns a priority to the ABR for its selection as the priority default route in an area.

To set the metric value

```
>> # /cfg/l3/ospf/default <metric value> <metric type (1 or 2)>
```

- *metric value* sets the priority for choosing this device for the default route.
  - — The value **none** sets no default.
  - — The value **1** sets the highest priority for the default route.
- *metric type* determines the method for influencing routing decisions for external routes.

To clear a default route metric

```
>> # /cfg/l3/ospf/default none
```

Virtual Links

Usually, all areas in an OSPF AS are physically connected to the backbone. In some cases where this is not possible, you can use a *virtual link*. Virtual links are created to connect one area to the backbone through another non-backbone area (see Figure 19 - Default Routes Example, page 144).

The area which contains a virtual link must be a transit area and have full routing information. Virtual links cannot be configured inside a stub area or NSSA. The area type must be defined as *transit* using the following command:

```
>> # /cfg/l3/ospf/aindex <area index> /type transit
```

The virtual link must be configured on the routing devices at each endpoint of the virtual link, though they may traverse multiple routing devices.

To configure Alteon as one end-point of a virtual link

```
>> # /cfg/l3/ospf/virt <link number> /aindex <area index> /nbr <router ID>
```

- *link number* is a value between 1 and 3.
- *area index* is the OSPF area index of the transit area.
- *router ID* is the IP address of the virtual neighbor (*nbr*), the routing device at the target end-point.

Another router ID is needed when configuring a virtual link in the other direction. To provide Alteon with a router ID, see Router ID, page 145.

For a detailed configuration example on Virtual Links, see Example 2: Virtual Links, page 152.

Router ID

Routing devices in OSPF areas are identified by a router ID. The router ID is expressed in IP address format. The IP address of the router ID is not required to be included in any IP interface range or in any OSPF area.

The router ID can be configured in one of the following two ways:

- **Dynamically**—By default, OSPF protocol configures the lowest IP interface IP address as the router ID.
- **Statically**—Use the following command to manually configure the router ID:

```
>> # /cfg/l3/rtrid <IP address>
```

To modify the router ID from static to dynamic

Set the router ID to 0.0.0.0, save the configuration, and reboot Alteon.

To view the router ID

```
>> # /info/13/ospf/gen
```

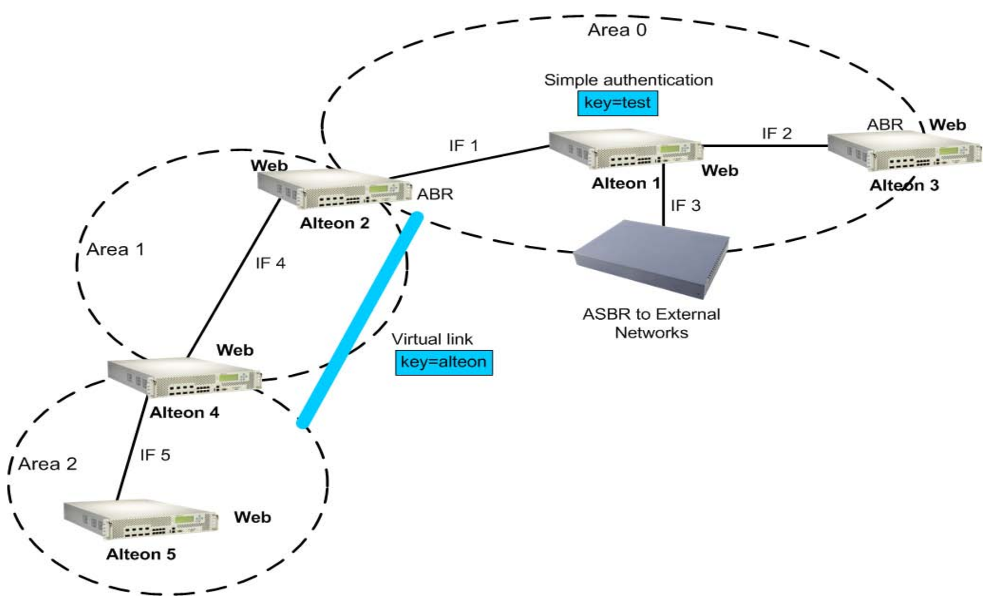
Authentication

OSPF protocol exchanges can be authenticated so that only trusted routing devices can participate. This ensures less processing on routing devices that are not listening to OSPF packets.

OSPF allows packet authentication and uses IP multicast when sending and receiving packets. Routers participate in routing domains based on predefined passwords. Alteon supports simple password (type 1 plain text passwords) and MD5 cryptographic authentication for OSPF version 2. This type of authentication allows a password to be configured per area.

Figure 20 - Authentication Example, page 146 shows authentication configured for area 0 with the password test. Simple authentication is also configured for the virtual link between area 2 and area 0. Area 1 is not configured for OSPF authentication.

Figure 20: Authentication Example



Example Configure Simple Plain Text OSPF Passwords

This example uses the configuration illustrated in Figure 20 - Authentication Example, page 146.

1. Enable OSPF authentication for Area 0 on Alteons 1, 2, and 3.

```
>> # /cfg/l3/ospf/aindex 0/auth password
```

2. Configure a simple text password up to eight characters for each OSPF IP interface in Area 0 on Alteons 1, 2, and 3.

```
>> # /cfg/l3/ospf/if 1
>> OSPF Interface 1 # key test
>> OSPF Interface 1 # /cfg/l3/ospf/if 2
>> OSPF Interface 2 # key test
>> OSPF Interface 1 # /cfg/l3/ospf/if 3
>> OSPF Interface 3 # key test
```

3. Enable OSPF authentication for Area 2 on Alteon 4.

```
>> # /cfg/l3/ospf/aindex 2/auth password
```

4. Configure a simple text password up to eight characters for the virtual link between Area 2 and Area 0 on Alteons 2 and 4.

```
>> # /cfg/l3/ospf/virt 1/key alteon
```

Example Configure MD5 Authentication

This example uses the configuration illustrated in Figure 20 - Authentication Example, page 146.

1. Enable OSPF MD5 authentication for Area 0 on Alteons 1, 2, and 3.

```
>> # /cfg/l3/ospf/aindex 0/auth md5
```

2. Configure MD5 key ID for Area 0 on Alteons 1, 2, and 3.

```
>> # /cfg/l3/ospf/md5key 1/key test
```

3. Assign MD5 key ID to OSPF interfaces on Alteons 1, 2, and 3.

```
>> # /cfg/l3/ospf/if 1
>> OSPF Interface 1 # mdkey 1
>> OSPF Interface 1 # /cfg/l3/ospf/if 2
>> OSPF Interface 2 # mdkey 1
>> OSPF Interface 1 # /cfg/l3/ospf/if 3
>> OSPF Interface 3 # mdkey 1
```

4. Enable OSPF MD5 authentication for Area 2 on Alteon 4.

```
>> # /cfg/l3/ospf/aindex 2/autn md5
```

5. Configure MD5 key for the virtual link between Area 2 and Area 0 on Alteons 2 and 4.

```
>> # /cfg/l3/ospf/md5key 2/key alteon
```

6. Assign MD5 key ID to OSPF virtual link on Alteons 2 and 4.

```
>> # /cfg/l3/ospf/virt 1/mdkey 2
```

Host Routes for Load Balancing

Alteon implementation of OSPF includes host routes. Host routes are used for advertising network device IP addresses to external networks, accomplishing the following goals:

- **Server Load Balancing (SLB) within OSPF**—Host routes advertise virtual server IP addresses to external networks. This allows standard SLB between Alteon and the server pools in an OSPF environment. For more information on SLB, see Server Load Balancing, page 165 and the *Alteon Application Switch Operating System Command Reference*.
- **ABR Load Sharing**—As a second form of load balancing, host routes can be used for dividing OSPF traffic among multiple ABRs. To accomplish this, each Alteon provides identical services but advertises a host route for a different virtual server IP address to the external network. If each virtual server IP address serves a different and equal portion of the external world, incoming traffic from the upstream router should be split evenly among ABRs.
- **ABR Failover**—Complementing ABR load sharing, identical host routes can be configured on each ABR. These host routes can be given different costs so that a different ABR is selected as the preferred route for each virtual server and the others are available as backups for failover purposes.

If redundant routes via multiple routing processes (such as OSPF, RIP, BGP, or static routes) exist on your network, Alteon defaults to the OSPF-derived route. For a configuration example, see 4: Host Routes, page 158.

Redistributing Routes into OSPF

Alteon lets you emulate an ASBR by redistributing information from other routing protocols (static, RIP, iBGP, eBGP, and fixed routes) into OSPF. For information on ASBR, see Types of OSPF Routing Devices, page 139. For example, you can instruct OSPF to readvertise a RIP-derived route into OSPF as an AS-External LSA. Based on this LSA, other routers in the OSPF routing domain installs an OSPF route.

Use the following command to redistribute a protocol into OSPF:

```
>> /cfg/l3/ospf/redist <protocol name>
```

- *protocol name* is static, RIP, iBGP, eBGP, or fixed. By default, these protocol routes are not redistributed into OSPF.

Use one of the following three methods to redistribute the routes of a particular protocol into OSPF:

- Exporting all the routes of the protocol
- Using route maps

  Route maps allow you to control the redistribution of routes between routing domains. For conceptual information on route maps, see Route Maps, page 126.
- Exporting all routes of the protocol except a few selected routes

Each of these methods is discussed in detail in the following sections.

Note: Alteon does not redistribute Layer 3 interface IPv6 addresses when the address has a prefix length of 128.

Exporting All Routes

Use the following command to redistribute all routes of a protocol:

```
>> /cfg/l3/ospf/redist <protocol name> /export <metric> <metric type>
```

- *metric* sets the OSPF cost for the route
- *metric type* (either 1 or 2) determines whether the route's cost includes or excludes external costs of the route

If you want to remove a previous configuration to export all the routes of a protocol, use the parameter **none** to the export command:

```
>> /cfg/l3/ospf/redist <protocol name> /export none
```

Using Route Maps to Export Selected Routes

Use route maps to specify which routes of the protocol that you want exported into OSPF. Table 18 - Commands for Using Route Maps, page 149 lists the tasks that you can perform using route maps:

Table 18: Commands for Using Route Maps

Task	Command
Adding a route map for a particular protocol	`/cfg/l3/ospf/redist <protocol name> /add <route map numbers>`
Adding all 32 route maps	`/cfg/l3/ospf/redist <protocol name> /add all`
Removing a route map for a particular protocol	`/cfg/l3/ospf/redist <protocol name> /rem <route map numbers>`
Removing all 32 route maps for a particular protocol	`/cfg/l3/ospf/redist <protocol name> /rem all`

OSPF does not require you to set all the fields in the route map menu. The following procedure includes the route maps and network filter parameter that must be set:

1. Enable the route map.

```
>> /cfg/l3/rmap <route map number> /ena
```

2. Assign the metric value in the AS-External LSA.

```
>> /cfg/l3/rmap <route map number> /metric <metric value>
```

   If a route map is added to a protocol for redistribution, and if the routes of that protocol match any of the routes in the access lists, and if action is set to permit, then those routes are redistributed into OSPF using the metric and metric type assigned for that route map. Metric sets the priority for choosing this device for the default route.

3. Enable the access list.

```
>> /cfg/l3/rmap <route map number> /alist <access list number> /ena
```

4. Set the action to **permit** for the access list.

```
>> /cfg/l3/rmap <route map number> /alist <access list number> /action permit
```

   To redistribute routes matched by the route map, the action in the *alist* must be set to **permit**. If the action is set to **deny**, the routes matched by the route map are not redistributed.

5. Link a network filter to the access list.

```
>> /cfg/l3/rmap <route map number> /alist <access list number> /nwf <network
filter number>
```

6. Enable the network filter.

```
>> /cfg/l3/nwf <network filter number> /ena
```

7. Specify the IP address and mask for the network filter.

```
>> /cfg/l3/nwf 1/addr <IP address> /mask <IP mask>
```

Optional Parameters for Route Maps

Set the following optional parameters (metric type and metric) for route redistribution into OSPF:

1. Assign the metric type in the AS-External LSA.

```
>> /cfg/l3/rmap <route map number> /type [1|2]
```

   The *type* is the method for influencing routing decisions for external routes.

2. Match the metric of the protocol route.

```
>> /cfg/l3/rmap <l> /alist <access list number> /metric <metric value>
```

   The *metric value* sets the priority for choosing this device for the route. The value **none** sets no default, and 1 sets the highest priority for the route.

Exporting All Routes Except a Few Selected Routes

This method is a combination of Exporting All Routes, page 149 and Using Route Maps to Export Selected Routes, page 149). The basic steps to configure this method are outlined below:

1. Configure OSPF to export all routes of the protocol using the export command as described in Exporting All Routes, page 149.
2. Use route maps to configure routes to be denied by setting the action in the access list of the route map to **deny**.

The configuration of the route map is similar to that described in the second method except that the action is set to **deny**.

OSPF Configuration Examples

Each of the configuration examples in this section are constructed using the following basic steps:

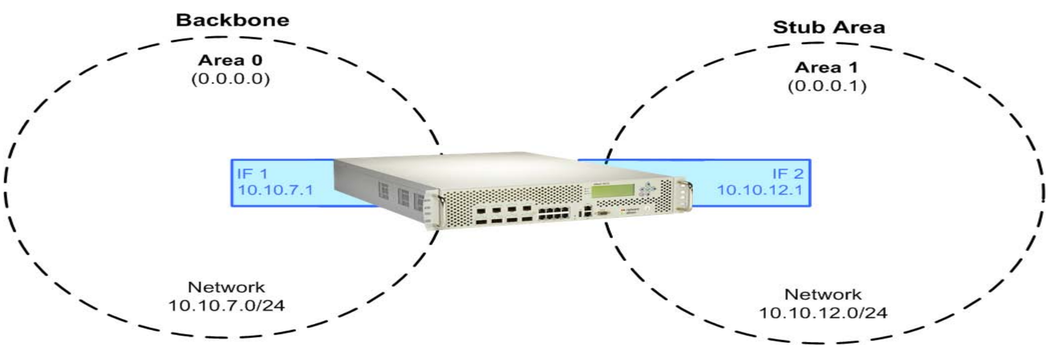
1. Configure IP interfaces—One IP interface is required for each desired network (range of IP addresses) being assigned to an OSPF area on Alteon.
2. Optionally configure the router ID—The router ID is required only when configuring virtual links on Alteon.
3. Enable OSPF on Alteon.
4. Define the OSPF areas.
5. Configure OSPF interface parameters—IP interfaces are used for attaching networks to the various areas.

6. Optionally configure route summarization between OSPF areas.
7. Optionally configure virtual links.
8. Optionally configure host routes.

Example 1: Simple OSPF Domain

In this example, two OSPF areas are defined: the backbone and the stub area. A stub area does not allow advertisements of external routes, thus reducing the size of the database. Instead, a default summary route of IP address 0.0.0.0 is inserted into the stub area. Any traffic for IP address destinations outside the stub area is forwarded to the stub area's IP interface, and then into the backbone.

Figure 21: Simple OSPF Domain Example



1. Configure IP interfaces on each network that is attached to OSPF areas.

   Two IP interfaces are needed: one for the backbone network on 10.10.7.0/24, and one for the stub area network on 10.10.12.0/24.

`>> # /cfg/l3/if 1`	(Select menu for IP interface 1)
`>> IP Interface 1 # addr 10.10.7.1`	(Set IP address on backbone network)
`>> IP Interface 1 # mask 255.255.255.0`	(Set IP mask on backbone network)
`>> IP Interface 1 # enable`	(Enable IP interface 1)
`>> IP Interface 1 # /cfg/l3/if 2`	(Select menu for IP interface 2)
`>> IP Interface 2 # addr 10.10.12.1`	(Set IP address on stub area network)
`>> IP Interface 2 # mask 255.255.255.0`	(Set IP mask on stub area network)
`>> IP Interface 2 # enable`	(Enable IP interface 2)

2. Enable OSPF.

`>> IP Interface 2 # /cfg/l3/ospf/on`	(Enable OSPF on Alteon)

3. Define the backbone. Always configure the backbone as a transit area using `areaid 0.0.0.0`.

`>> Open Shortest Path First # aindex 0`	(Select menu for area index 0)
`>> Open Area (index) 0 # areaid 0.0.0.0`	(Set the ID for backbone area 0)
`>> Open Area (index) 0 # type transit`	(Define backbone as transit type)
`>> OSPF Area (index) 0 # enable`	(Enable the area)

4. Define the stub area.

`>> OSPF Area (index) 0 # /cfg/l3/ospf/aindex 1`	(Select menu for area index 1)
`>> OSPF Area (index) 1 # areaid 0.0.0.1`	(Set the area ID for OSPF area 1)
`>> OSPF Area (index) 1 # type stub`	(Define area as stub type)
`>> OSPF Area (index) 1 # enable`	(Enable the area)

5. Attach the network interface to the backbone.

`>> OSPF Area 1 # /cfg/l3/ospf/if 1`	(Select OSPF menu for IP interface 1)
`>> OSPF Interface 1 # aindex`	(Attach network to backbone index)
`>> OSPF Interface 1 # enable`	(Enable the backbone interface)

6. Attach the network interface to the stub area.

`>> OSPF Interface 1 # /cfg/l3/ospf/if 2`	(Select OSPF menu for IP interface 2)
`>> OSPF Interface 2 # aindex 1`	(Attach network to stub area index)
`>> OSPF Interface 2 # enable`	(Enable the stub area interface)

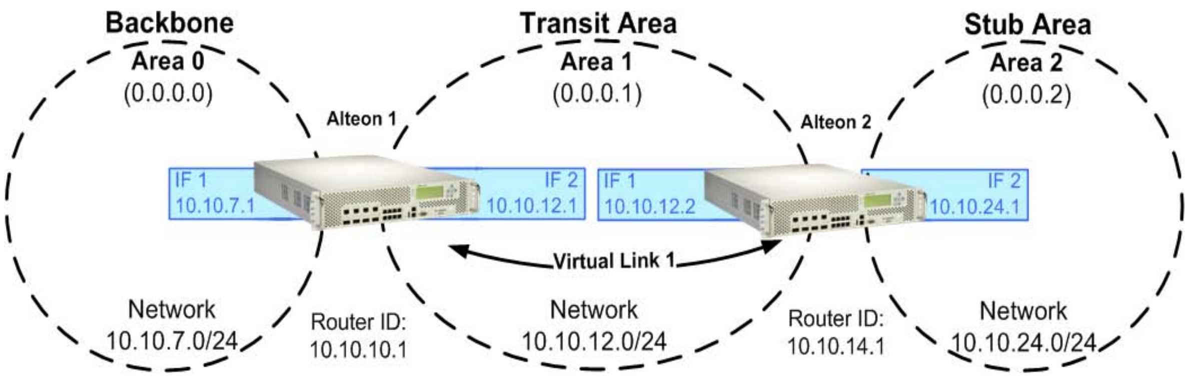
7. Apply and save the configuration changes.

`>> OSPF Interface 2 # apply`	(Global command to apply all changes)
`>> OSPF Interface 2 # save`	(Global command to save all changes)

Example 2: Virtual Links

In the example shown in Figure 22 - Virtual Links Example, page 153, area 2 is not physically connected to the backbone as is usually required. Instead, area 2 is connected to the backbone through a virtual link through area 1. The virtual link must be configured at each endpoint.

Figure 22: Virtual Links Example



Configuring OSPF for a Virtual Link on Alteon 1

1. Configure IP interfaces on each network that is attached to Alteon.

   In this example, two IP interfaces are needed on Alteon 1: the backbone network on 10.10.7.0/24, and the transit area network on 10.10.12.0/24.

`>> # /cfg/l3/if 1`	(Select menu for IP interface 1)
`>> IP Interface 1 # addr 10.107.1`	(Set IP address on backbone network)
`>> IP Interface 1 # mask 255.255.255.0`	(Set IP mask on backbone network)
`>> IP Interface 1 # enabled`	(Enable IP interface 1)
`>> IP Interface 1 # /cfg/l3/if 2`	(Select menu for IP interface 2)
`>> IP Interface 2 # addr 10.10.12.1`	(Set IP address on transit area network)
`>> IP Interface 2 # mask 255.255.255.0`	(Set IP mask on transit area network)
`>> IP Interface 2 # enable`	(Enable interface 2)

2. Configure the router ID. A router ID is required when configuring virtual links. Later, when configuring the other end of the virtual link on Alteon 2, the router ID specified here is used as the target virtual neighbor (*nbr*) address.

`>> IP Interface 2 # /cfg/l3/rtrid 10.10.10.1`	(Set static router ID on Alteon 1)

3. Enable OSPF.

`>> IP # /cfg/l3/ospf/on`	(Enable OSPF on Alteon 1)

4. Define the backbone.

`>> Open Shortest Path First # aindex 0`	(Select menu for area index 0)
`>> OSPF Area (index) 0 # areaid 0.0.0.0`	(Set the area ID for backbone area 0)
`>> OSPF Area (index) 0 # type transit`	(Define backbone as transit type)
`>> OSPF Area (index) 0 # enable`	(Enable the area)

5. Define the transit area. The area that contains the virtual link must be configured as a transit area.

`>> OSPF Area (index) 0 # /cfg/l3/ospf/aindex 1`	(Select menu for area index 1)
`>> OSPF Area (index) 1 # areaid 0.0.0.1`	(Set the area ID for OSPF area 1)
`>> OSPF Area (index) 1 # type transit`	(Define area as transit type)
`>> OSPF Area (index) 1 # enable`	(Enable the area)

6. Attach the network interface to the backbone.

`>> OSPF Area (index) 1 # /cfg/l3/ospf/if 1`	(Select OSPF menu for IP interface 1)
`>> OSPF Interface 1 # aindex 0`	(Attach network to backbone index)
`>> OSPF Interface 1 # enable`	(Enable the backbone interface)

7. Attach the network interface to the transit area.

`>> OSPF Interface 1 # /cfg/l3/ospf/if 2`	(Select OSPF menu for IP interface 2)
`>> OSPF Interface 2 # aindex 1`	(Attach network to transit area index)
`>> OSPF Interface 2 # enable`	(Enable the transit area interface)

8. Configure the virtual link. The nbr router ID configured in this step must be the same as the router ID that is configured for step 2 in the procedure for Alteon 2.

`>> OSPF Interface 2 # /cfg/l3/ospf/virt 1`	(Specify a virtual link number)
`>> OSPF Virtual Link 1 # aindex 1`	(Specify the transit area for the virtual link)
`>> OSPF Virtual Link 1 # nbr 10.10.14.1`	(Specify the router ID of the recipient)
`>> OSPF Virtual Link 1 # enable`	(Enable the virtual link)

9. Apply and save the configuration changes.

`>> OSPF Interface 2 # apply 1`	(Global command to apply all changes)
`>> OSPF Interface 2 # save`	(Global command to save all changes)

Configuring OSPF for a Virtual Link on Alteon 2

1. Configure IP interfaces on each network that is attached to OSPF areas.

   Two IP interfaces are needed on Alteon 2: the transit area network on 10.10.12.0/24, and the stub area network on 10.10.24.0/24.

`>> # /cfg/l3/if 1`	(Select menu for IP interface 1)
`>> IP Interface 1 # addr 10.10.12.2`	(Set IP address on transit area net-work)
`>> IP Interface 1 # mask 255.255.255.0`	(Set IP mask on transit area network)
`>> IP Interface 1 # enable`	(Enable IP interface 1)
`>> IP Interface 1 # /cfg/l3/if 2`	(Select menu for IP interface 2)
`>> IP Interface 2 # 10.10.24.1`	(Set IP address on stub area network)

`>> IP Interface 2 # mask 255.255.255.0`	(Set IP mask on stub area network)
`>> IP Interface 2 # enable`	(Enable IP interface 2)

2. Configure the router ID. A router ID is required when configuring virtual links. This router ID should be the same one specified as the target virtual neighbor (*nbr*) in step 8 for Alteon 1.

```
>> IP Interface 2 # /cfg/l3/rtrid 10.10.14.1
```

3. Enable OSPF.

```
>> IP cfg/13/ospf/on
```

4. Configure the backbone index on the non-backbone end of the virtual link.

`>> Open Shortest Path First # aindex 0`	(Select the menu for area index 0)
`>> OSPF Area (index) 0 # areaid 0.0.0.0`	(Set the area ID for OSPF area 0)
`>> OSPF Area (index) 0 # enable`	(Enable the area)

5. Define the transit area.

`>> OSPF Area (index) 0 # /cfg/l3/ospf/aindex 1`	(Select menu for area index 1)
`>> OSPF Area (index) 1 # areaid 0.0.0.1`	(Set the area ID for OSPF area 1)
`>> OSPF Area (index) 1 # type transit`	(Define area as transit type)
`>> OSPF Area (index) 1 # enable`	(Enable the area)

6. Define the stub area.

`>> OSPF Area (index) 1 # /cfg/l3/ospf/aindex 2`	(Select menu for area index 2)
`>> OSPF Area (index) 2 # areaid 0.0.0.2`	(Set the area ID for OSPF area 2)
`>> OSPF Area (index) 2 # type stub`	(Define area as stub type)
`>> OSPF Area (index) 2 # enable`	(Enable the area)

7. Attach the network interface to the backbone.

`>> OSPF Area (index) 2 # /cfg/l3/ospf/if 1`	(Select OSPF menu for IP interface 1)
`>> OSPF Interface 1 # aindex 1`	(Attach network to transit area index)
`>> OSPF Interface 1 # enable`	(Enable the transit area interface)

8. Attach the network interface to the transit area.

`>> OSPF Interface 1 # /cfg/l3/ospf/if 2`	(Select OSPF menu for IP interface 2)
`>> OSPF Interface 2 # aindex 2`	(Attach network to stub area index)
`>> OSPF Interface 2 # enable`	(Enable the stub area interface)

9. Configure the virtual link. The nbr router ID configured in this step must be the same as the router ID that was configured in step 2 for Alteon 1.

`>> OSPF Interface 2 # /cfg/l3/ospf/virt 1`	(Specify a virtual link number)
`>> OSPF Virtual Link 1 # aindex 1`	(Specify the transit area for the virtual link)
`>> OSPF Virtual Link 1 # nbr 10.10.10.1`	(Specify the router ID of the recipient)
`>> OSPF Virtual Link 1 # enable`	(Enable the virtual link)

10. Apply and save the configuration changes.

`>> OSPF Interface 2 # apply 1`	(Global command to apply all changes)
`>> OSPF Interface 2 # save`	(Global command to save all changes)

Notes **Other Virtual Link Options**

- You can use redundant paths by configuring multiple virtual links.
- Only the endpoints of the virtual link are configured. The virtual link path may traverse multiple routers in an area as long as there is a routable path between the endpoints.

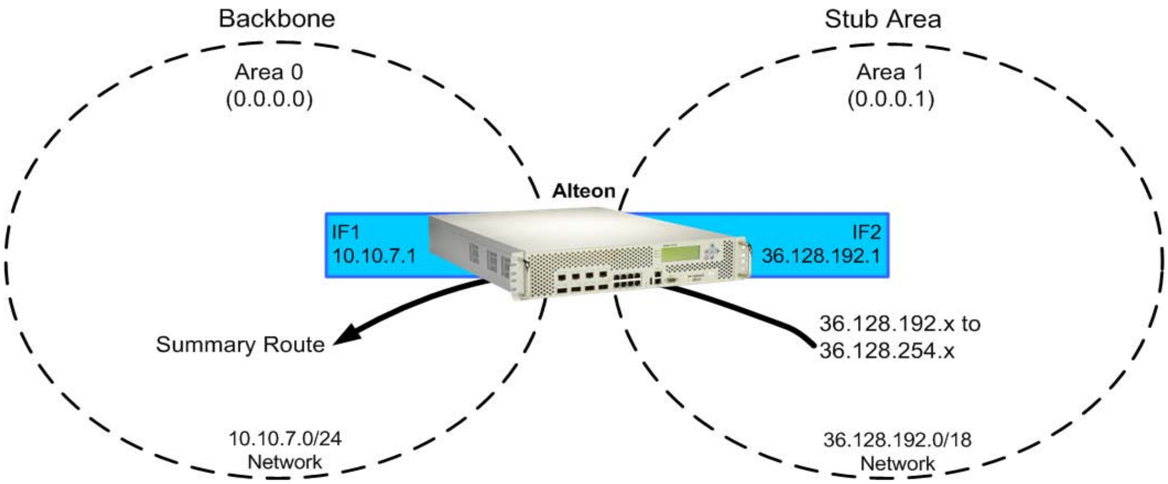
Example 3: Summarizing Routes

By default, ABRs advertise all the network addresses from one area into another area. Route summarization can be used for consolidating advertised addresses and reducing the perceived complexity of the network.

If the network IP addresses in an area are assigned to a contiguous subnet range, you can configure the ABR to advertise a single summary route that includes all the individual IP addresses within the area.

Figure 23 - Summarizing Routes Example, page 157 illustrates one summary route from area 1 (stub area) injected into area 0 (the backbone). The summary route consists of all IP addresses from 36.128.192.0 through 36.128.254.255, except for the routes in the range 36.128.200.0 through 36.128.200.255.

Figure 23: Summarizing Routes Example



Note: You can specify a range of addresses to *prevent* advertising by using the hide option. In this example, routes in the range 36.128.200.0 through 36.128.200.255 are kept private.

1. Configure IP interfaces for each network which is attached to OSPF areas.

`>> OSPF Virtual Link 1 # aindex 1`	(Select menu for IP interface 1)
`>> IP Interface 1 # addr 10.10.7.1`	(Set IP address on backbone network)
`>> IP Interface 1 # mask 255.255.255.0`	(Set IP mask on backbone network)
`>> IP Interface 1 # ena`	(Enable IP interface 1)
`>> IP Interface 1 # /cfg/l3/if 2`	(Select menu for IP interface 2)
`>> IP Interface 2 # addr 36.128.192.1`	(Set IP address on stub area network)
`>> IP Interface 2 # mask 255.255.192.0`	(Set IP mask on stub area network)
`>> IP Interface 2 # ena`	(Enable IP interface 2)

2. Enable OSPF.

`>> IP Interface 2 # /cfg/l3/ospf/on`	(Enable OSPF on Alteon)

3. Define the backbone.

`>> Open Shortest Path First # aindex 0`	(Select menu for area index 0)
`>> OSPF Area (index) 0 # areaid 0.0.0.0`	(Set the ID for backbone area 0)
`>> OSPF Area (index) 0 # type transit`	(Define backbone as transit type)
`>> OSPF Area (index) 0 # enable`	(Enable the area)

4. Define the stub area.

`>> OSPF Area (index) 0 # /cfg/l3/ospf/aindex 1`	(Select menu for area index 1)
`>> OSPF Area (index) 1 # areaid 0.0.0.1`	(Set the area ID for OSPF area 1)

`>> OSPF Area (index) 1 # type stub`	(Define area as stub type)
`>> OSPF Area (index) 1 # enable`	(Enable the area)

5. Attach the network interface to the backbone.

`>> OSPF Area (index) 1 # /cfg/l3/ospf/if 1`	(Select OSPF menu for IP interface 1)
`>> OSPF Interface 1 # aindex 0`	(Attach network to backbone index)
`>> OSPF Interface 1 # enable`	(Enable the backbone interface)

6. Attach the network interface to the stub area.

`>> OSPF Interface 1 # /cfg/l3/ospf/if 2`	(Select OSPF menu for IP interface 2)
`>> OSPF Interface 2 # aindex`	(Attach network to stub area index)
`>> OSPF Interface 2 # enable`	(Enable the stub area interface)

7. Configure route summarization by specifying the starting address and mask of the range of addresses to be summarized

`>> OSPF Interface 2 # /cfg/l3/ospf/range 1`	(Select menu for summary range)
`>> OSPF Summary Range 1 # addr 36.128.192.0`	(Set base IP address of summary range)
`>> OSPF Summary Range 1 # mask 255.255.192.0`	(Set mask address for summary range)
`>> OSPF Summary Range 1 # aindex 0`	(Inject summary route into backbone)
`>> OSPF Summary Range 1 # enable`	(Enable summary range)

8. Use the hide command to prevent a range of addresses from advertising to the backbone.

`>> OSPF Interface 2 # /cfg/l3/ospf/range 2`	(Select menu for summary range)
`>> OSPF Summary Range 2 # addr 36.128.200.0`	(Set base IP address)
`>> OSPF Summary Range 2 # mask 255.255.255.0`	(Set mask address)
`>> OSPF Summary Range 2 # hide enable`	(Hide the range of addresses)

9. Apply and save the configuration changes.

`>> OSPF Summary Range 2 # apply`	(Global command to apply all changes)
`>> OSPF Summary Range 2 # save`	(Global command to save all changes)

Example 4: Host Routes

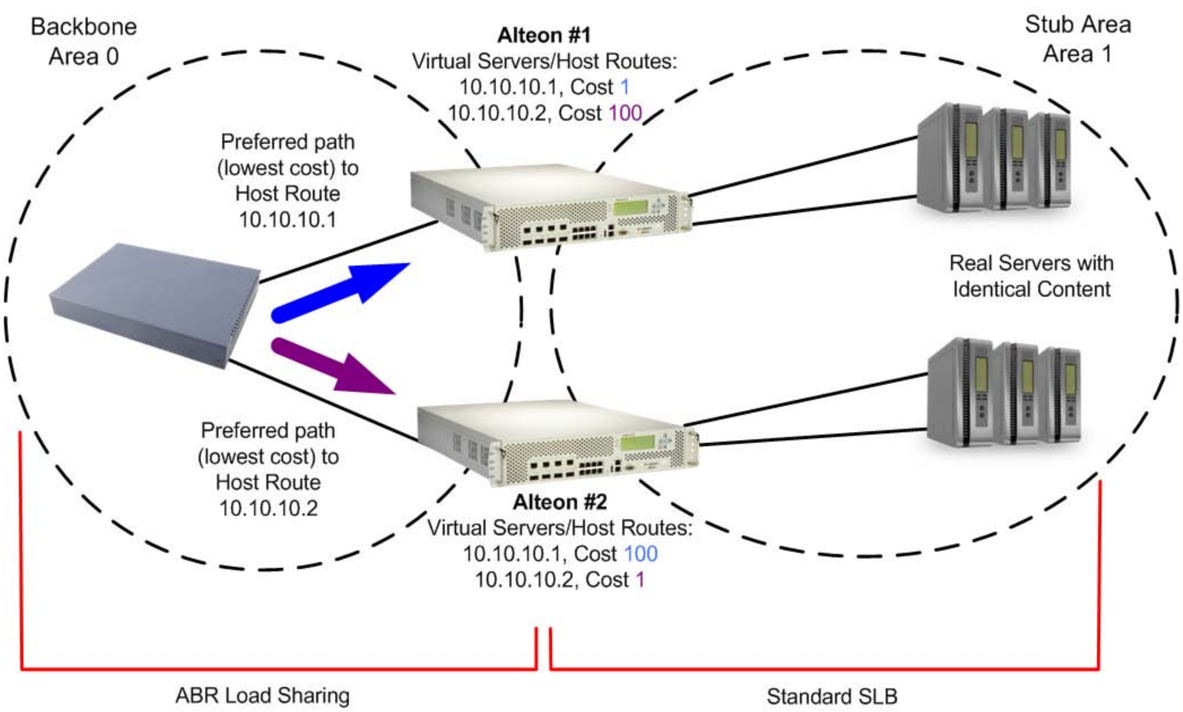
The Alteon OSPF implementation includes host routes. Host routes are used for advertising network device IP addresses to external networks and allows for Server Load Balancing (SLB) within OSPF. It also makes ABR load sharing and failover possible.

In Figure 24 - Host Routes Example, page 159, both devices have access to servers with identical content and are configured with the same virtual server IP addresses: 10.10.10.1 and 10.10.10.2. Alteon 1 is given a host route with a low cost for virtual server 10.10.10.1, and another host route with a high cost for virtual server 10.10.10.2. Alteon 2 is configured with the same hosts but with the costs reversed; one host route has a high cost for virtual server 10.10.10.1, and another has a low cost for virtual server 10.10.10.2.

All four host routes are injected into the upstream router and advertised externally. Traffic comes in for both virtual server IP addresses (10.10.10.1 and 10.10.10.2). The upstream router sees that both addresses exist on both devices and uses the host route with the lowest cost for each. Traffic for 10.10.10.1 goes to Alteon 1 because its host route has the lowest cost for that address. Traffic for 10.10.10.2 goes to Alteon 2 because its host route has the lowest cost. This effectively shares the load among ABRs. Both devices then use standard Server Load Balancing (SLB) to distribute traffic among available real servers.

In addition, if one of Alteons were to fail, the upstream routing device would forward the traffic to the ABR whose host route has the next lowest cost. The remaining device assumes the entire load for both virtual servers.

Figure 24: Host Routes Example



Configuring Host Routes on Alteon 1

1. Configure IP interfaces for each network that is attached to OSPF areas.

`>> Virtual server 1 # /cfg/l3/if 1`	(Select menu for IP interface 1)
`>> IP Interface 1 # addr 10.10.10.5`	(Set IP address on backbone network)
`>> IP Interface 1 # enable`	(Enable IP interface 1)
`>> IP Interface 1 # /cfg/l3/if 2`	(Select menu for IP interface 2)
`>> IP Interface 2 # addr 100.100.100.40`	(Set IP address on stub area network)
`>> IP Interface 2 # enable`	(Enable IP interface 2)

2. Configure basic SLB parameters. Alteon 1 is connected to two real servers. Each real server is given an IP address and is placed in the same real server group.

`>> # /cfg/slb/real 1`	(Select menu for real server 1)
`>> Real server 1 # rip 100.100.100.25`	(Set the IP address for real server 1)
`>> Real server 1 # ena`	(Enable the real server)
`>> Real server 1 # /cfg/slb/real 2`	(Select menu for real server 2)
`>> Real server 2 # rip 100.100.100.26`	(Set the IP address for real server 2)
`>> Real server 2 # ena`	(Enable the real server)
`>> Real server 2 # /cfg/slb/group 1`	(Select menu for real server group 1)
`>> Real server group 1 # add 1`	(Add real server 1 to group)
`>> Real server group 1 # add 2`	(Add real server 2 to group)
`>> Real server group 1 # enable`	(Enable the group)
`>> Real server group 1 # /cfg/slb/on`	(Turn SLB on)

3. Configure client and server processing on specific ports.

`>> Layer 4 # /cfg/slb/port 4`	(Select port 4)
`>> SLB Port 4 # client ena`	(Enable client processing on port 4)
`>> SLB Port 4 # /cfg/slb/port 5`	(Select port 5)
`>> SLB Port 5 # server ena`	(Enable server processing on port 5)

4. Enable direct access mode.

`>> Layer 4 Port 5 # /cfg/slb/adv`	(Select the SLB advance menu)
`>> Layer 4 Advanced # direct ena`	(Enable DAM)
`>> Layer 4 Advanced# ...`	(Return to the SLB menu)

5. Configure the primary virtual server. Alteon 1 is preferred for virtual server 10.10.10.1.

`>> Layer 4 # /cfg/slb/virt`	(Select menu for virtual server 1)
`>> Virtual server 1 # vip 10.10.10.1`	(Set the IP address for virtual server 1)
`>> Virtual server 1 # ena`	(Enable the virtual server)
`>> Virtual server 1 # service http`	(Select menu for service on virtual server)
`>> Virtual server 1 http service # group 1`	(Use real server group 1 for HTTP service)

6. Configure the backup virtual server. Alteon 1 acts as a backup for virtual server 10.10.10.2. Both virtual servers in this example are configured with the same real server group and provide identical services.

`>> Virtual server 2 http service # /cfg/slb/virt 2`	(Select menu for virtual server 2)
`>> Virtual server 1 # vip 10.10.10.2`	(Set the IP address for virtual server 2)
`>> Virtual server 1 # ena`	(Enable the virtual server)
`>> Virtual server 1 # service http`	(Select menu for service on virtual server)
`>> Virtual server 1 http service # group 1`	(Use real server group 1 for HTTP service)

7. Enable OSPF on Alteon 1.

`>> IP Interface 2 # /cfg/l3/ospf/on`	(Enable OSPF on Alteon 1)

8. Define the backbone.

`>> Open Shortest Path First # aindex 0`	(Select menu for area index 0)
`>> OSPF Area (index) 0 # areaid 0.0.0.0`	(Set the ID for backbone area 0)
`>> OSPF Area (index) 0 # type transit`	(Define backbone as transit type)
`>> OSPF Area (index) 0 # enable`	(Enable the area)

9. Define the stub area.

`>> OSPF Area (index) 0 # /cfg/l3/ospf/aindex 1`	(Select menu for area index 1)
`>> OSPF Area (index) 1 # areaid 0.0.0.1`	(Set the ID for stub area 1)
`>> OSPF Area (index) 1 # type stub`	(Define area as stub type)
`>> OSPF Area (index) 1 # enable`	(Enable the area)

10. Attach the network interface to the backbone.

`>> OSPF Area (index) 1 # /cfg/l3/ospf/if 1`	(Select OSPF menu for IP interface 1)
`>> OSPF Interface 1 # aindex 0`	(Attach network to backbone index)
`>> OSPF Interface 1 # enable`	(Enable the backbone interface)

11. Attach the network interface to the stub area.

`>> OSPF Interface 1 # /cfg/l3/ospf/if 2`	(Select OSPF menu for IP interface 2)
`>> OSPF Interface 2 # aindex 1`	(Attach network to stub area index)
`>> OSPF Interface 2 # enable 1`	(Enable the stub area interface)

12. Configure host routes. One host route is needed for each virtual server on Alteon 1. Since virtual server 10.10.10.1 is preferred for Alteon 1, its host route has a low cost. Because virtual server 10.10.10.2 is used as a backup in case Alteon 2 fails, its host route has a high cost.

Note: You do not need to enable redistribution (`/cfg/l3/ospf/redist`) if you configure virtual server routes as host routes.

`>> OSPF Interface 2 # /cfg/l3/ospf/host 1`	(Select menu for host route 1)
`>> OSPF Host Entry 1 # addr 10.10.10.1`	(Set IP address same as virtual server 1)
`>> OSPF Host Entry 1 # aindex 0`	(Inject host route into backbone area)
`>> OSPF Host Entry 1 # cost 1`	(Set low cost for preferred path)
`>> OSPF Host Entry 1 # enable`	(Enable the host route)
`>> OSPF Host Entry 1 # /cfg/l3/ospf/host 2`	(Select menu for host route 2)
`>> OSPF Host Entry 2 # addr 10.10.10.2`	(Set IP address same as virtual server 2)
`>> OSPF Host Entry 2 # aindex 0`	(Inject host route into backbone area)
`>> OSPF Host Entry 2 # cost 100`	(Set high cost for use as backup path)
`>> OSPF Host Entry 2 # enable`	(Enable the host route)

Note: When a service goes down, the corresponding host route is removed from advertising.

13. Apply and save the configuration changes.

`>> OSPF Host Entry 2 # apply`	(Global command to apply all changes)
`>> OSPF Host Entry 2 # save`	(Global command to save all changes)

Configuring Host Routes on Alteon 2

1. Configure basic SLB parameters. Alteon 2 is connected to two real servers. Each real server is given an IP address and is placed in the same real server group.

`>> # /cfg/slb/real 1`	(Select menu for real server 1)
`>> Real server 1 # rip 100.100.100.27`	(Set the IP address for real server 1)
`>> Real server 1 # enable`	(Enable the real server)
`>> Real server 1 # /cfg/slb/real 2`	(Select menu for real server 2)
`>> Real server 2 # rip 100.100.100.28`	(Set the IP address for real server 2)
`>> Real server 1 # rip 100.100.100.27`	(Enable the real server)
`>> Real server 2 # /cfg/slb/group 1`	(Select menu for real server group 1)
`>> Real server 1 # add 1`	(Add real server 1 to group)
`>> Real server group 1 # add 2`	(Add real server 2 to group)
`>> Real server group 1 # enable`	(Enable the group)
`>> Real server 1 # /cfg/slb/on`	(Turn SLB on)

2. Configure the virtual server parameters. The same virtual servers are configured as on Alteon 1.

`>> Layer 4 # /cfg/slb/virt 1`	(Select menu for virtual server 1)
`>> Virtual server 1 # vip 10.10.10.1`	(Set the IP address for virtual server 1)
`>> Virtual server 1 # enable`	(Enable the virtual server)

`>> Virtual server 1 # service http`	(Select menu for service on virtual server)
`>> Virtual server 1 http service # group 1`	(Use real server group 1 for http service)
`>> Virtual server 2 http service # /cfg/slb/virt 2`	(Select menu for virtual server 2)
`>> Virtual server 1 # vip 10.10.10.2`	(Set the IP address for virtual server 2)
`>> Virtual server 1 # enable`	(Enable the virtual server)
`>> Virtual server 1 # service http`	(Select menu for service on virtual server)
`>> Virtual server 1 # group`	(Use real server group 1 for http service)

3. Configure IP interfaces for each network that will be attached to OSPF areas.

`>> Virtual server 1# /cfg/l3/if 1`	(Select menu for IP Interface 1)
`>> IP Interface 1 # addr 10.10.10.6`	(Set IP address on backbone network)
`>> IP Interface 1 # enable`	(Enable IP interface 1)
`>> IP Interface 1 # /cfg/l3/if 2`	(Select menu for IP Interface 2)
`>> IP Interface 2 # addr 100.100.100.41`	(Set IP address on stub area network)
`>> IP Interface 2 # enable`	(Enable IP interface 2)

4. Enable OSPF on Alteon 2.

`>> IP Interface 2 # /cfg/l3/ospf/on`	(Enable OSPF on Alteon 2)

5. Define the backbone.

`>> Open Shortest Path# addr 10.10.10.6`	(Select menu for area index 0)
`>> OSPF Area (index) 0 # areaid 0.0.0.0`	(Set the ID for backbone area 0)
`>> OSPF Area (index) 0 # type transit`	(Define backbone as transit type)
`>> OSPF Area (index) 0 # enable`	(Enable the area)

6. Define the stub area.

`>> OSPF Area (index) 0 # /cfg/l3/ospf/aindex 1`	(Select menu for area index 1)
`>> OSPF Area (index) 1 # areaid 0.0.0.1`	(Set the ID for stub area 1)
`>> OSPF Area (index) 1 # type stub`	(Define area as stub type)
`>> OSPF Area (index) 1 # enable`	(Enable the area)

7. Attach the network interface to the backbone.

`>> OSPF Area (index) 1 # /cfg/l3/ospf/if 1`	(Select OSPF menu for IP interface 1)
`>> OSPF Interface 1 # aindex 0`	(Attach network to backbone index)
`>> OSPF Interface 1 # enable`	(Enable the backbone interface)

8. Attach the network interface to the stub area.

`>> OSPF Interface 1 # /cfg/l3/ospf/if 2`	(Select OSPF menu for IP interface 2)
`>> OSPF Interface 2 # aindex 1`	(Attach network to stub area index)
`>> OSPF Interface 2 # enable`	(Enable the stub area interface)

9. Configure host routes. Host routes are configured just like those on Alteon 1, except their costs are *reversed*. Since virtual server 10.10.10.2 is preferred for Alteon 2, its host route has been given a low cost. Because virtual server 10.10.10.1 is used as a backup in case Alteon 1 fails, its host route has been given a high cost.

`>> OSPF Interface 2 # /cfg/l3/ospf/host 1`	(Select menu for host route 1)
`>> OSPF Interface 1 # addr 10.10.10.1`	(Set IP address same as virtual server 1)
`>> OSPF Host Entry 1 # aindex 0`	(Inject host route into backbone area)
`>> OSPF Host Entry 1 # cost 100`	(Set high cost for use as backup path)
`>> OSPF Host Entry 1 # enable`	(Enable the host route)
`>> OSPF Host Entry 1 # /cfg/l3/ospf/host 2`	(Select menu for host route 2)
`>> OSPF Host Entry 2 # addr 10.10.10.2`	(Set IP address same as virtual server 2)
`>> OSPF Host Entry 2 # aindex 0`	(Inject host route into backbone area)
`>> OSPF Host Entry 2 # cost 2`	(Set low cost for primary path)
`>> OSPF Host Entry 2 # enable`	(Enable the host route)

10. Apply and save the configuration changes.

`>> OSPF Host Entry 2 # apply`	(Global command to apply all changes)
`>> OSPF Host Entry 2 # save`	(Global command to save all changes)

Verifying OSPF Configuration

Use the following commands to verify the OSPF configuration:

- `/info/l3/ospf/general`
- `/info/l3/ospf/nbr`
- `/info/l3/ospf/dbase/dbsum`
- `/info/l3/ospf/route`
- `/stats/l3/route`

Refer to the *Alteon Application Switch Operating System Command Reference* for information on these commands.

Chapter 12 – Server Load Balancing

Server Load Balancing (SLB) lets you configure Alteon to balance user session traffic among a pool of available servers that provide shared services.

This chapter includes the following sections:

For additional information on SLB commands, refer to the *Alteon Application Switch Operating System Command Reference*.

Understanding Server Load Balancing

SLB benefits your network in the following ways:

- **Increased efficiency for server utilization and network bandwidth**—With SLB, Alteon is aware of the shared services provided by your server pool and can then balance user session traffic among the available servers. Important session traffic gets through more easily, reducing user competition for connections on overused servers. For even greater control, traffic is distributed according to a variety of user-selectable rules.
- **Increased reliability of services to users**—If any server in a server pool fails, the remaining servers continue to provide access to vital applications and data. The failed server can be brought back up without interrupting access to services.
- **Increased scalability of services**—As users are added and the server pool's capabilities are saturated, new servers can be added to the pool transparently.

Identifying Your Network Needs

SLB may be the right option for addressing these vital network concerns:

- A single server no longer meets the demand for its particular application.
- The connection from your LAN to your server overloads server capacity.
- When servers hold critical application data and must remain available even in the event of a server failure.
- Your Web site is being used as a way to do business and for taking orders from customers. It must not become overloaded or unavailable.
- You want to use multiple servers or hot-standby servers for maximum server uptime.

- You must be able to scale your applications to meet client and LAN request capacity.
- You cannot afford to continue using a less effective load-balancing technique, such as DNS round-robin or a software-only system.

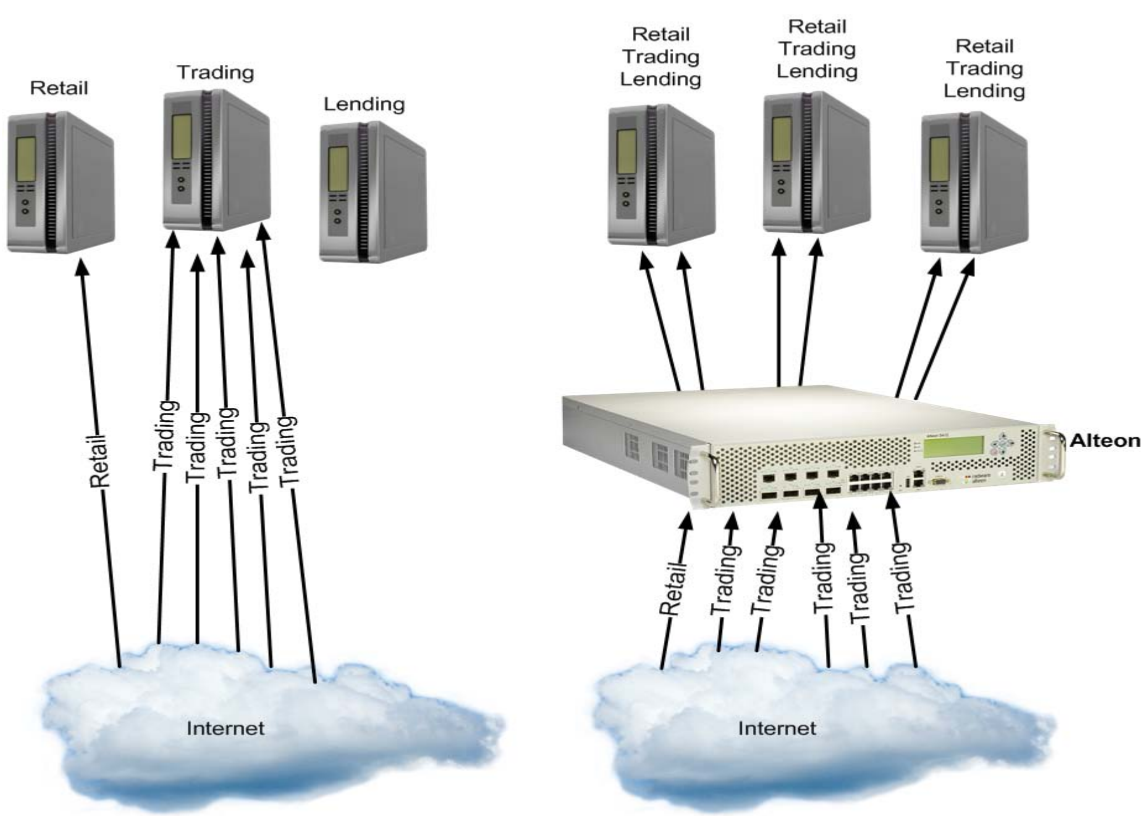
How Server Load Balancing Works

In an average network that employs multiple servers without SLB, each server usually specializes in providing one or two unique services. If one of these servers provides access to applications or data that is in high demand, it can become overused. Placing this kind of strain on a server can decrease the performance of the entire network as user requests are rejected by the server and then resubmitted by the user stations. Ironically, overuse of key servers often happens in networks where other servers are actually available.

The solution to getting the most from your servers is SLB. With this software feature, Alteon is aware of the services provided by each server. Alteon can direct user session traffic to an appropriate server, based on a variety of load-balancing algorithms.

Figure 25 - Traditional Versus SLB Network Configurations, page 166 illustrates traditional versus SLB network configurations:

Figure 25: Traditional Versus SLB Network Configurations



To provide load balancing for any particular type of service, each server in the pool must have access to identical content, either directly (duplicated on each server) or through a back-end network (mounting the same file system or database server).

Alteon with SLB software acts as a front-end to the servers, interpreting user session requests and distributing them among the available servers. Load balancing in Alteon can be done in the following ways:

- **Virtual server-based load balancing**—This is the traditional load balancing method. Alteon is configured to act as a virtual server and is given a virtual server IP address (or range of addresses) for each collection of services it distributes. Depending on your Alteon platform, there can be as many as 1023 virtual servers on Alteon, each distributing up to eight different services.

  Each virtual server is assigned a list of the IP addresses (or range of addresses) of the real servers in the pool where its services reside. When the user stations request connections to a service, they communicate with a virtual server on Alteon. When Alteon receives the request, it binds the session to the IP address of the best available real server and remaps the fields in each frame from virtual addresses to real addresses.

  HTTP, IP, FTP, RTSP, IDS, and static session WAP are examples of some of the services that use virtual servers for load balancing.
- **Filter-based load balancing**—A filter allows you to control the types of traffic permitted through Alteon. Filters are configured to allow, deny, or redirect traffic according to the IP address, protocol, or Layer 4 port criteria. In filter-based load balancing, a filter is used to redirect traffic to a real server group. If the group is configured with more than one real server entry, redirected traffic is load balanced among the available real servers in the group.

  Firewalls, WAP with RADIUS snooping, IDS, and WAN links use redirection filters to load balance traffic.
- **Content-based load balancing**—Content-based load balancing uses Layer 7 application data (such as URL, cookies, and Host Headers) to make intelligent load balancing decisions.

  URL-based load balancing, browser-smart load balancing, and cookie-based preferential load balancing are a few examples of content-based load balancing.

Implementing Server Load Balancing

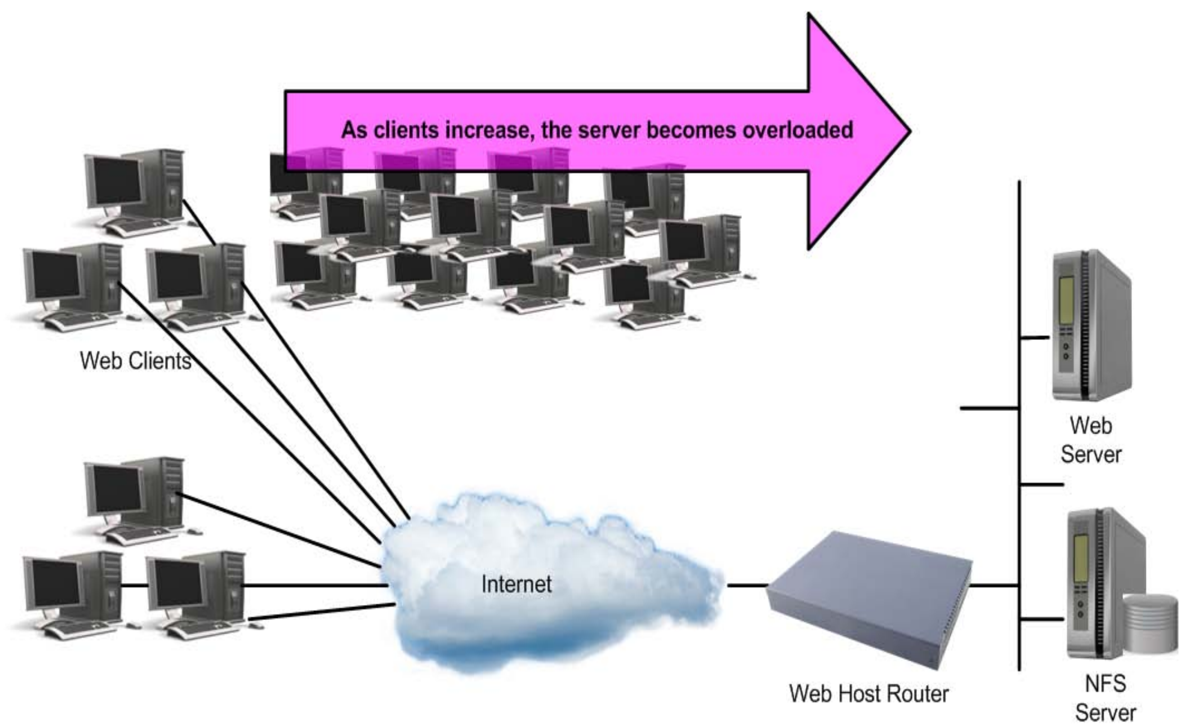
This section includes basic SLB implementation procedures, as well as customized SLB options. To implement basic Server Load Balancing, see Basic Server Load Balancing Topology, page 168.

The following configuration options can be used to customize SLB in Alteon:

Basic Server Load Balancing Topology

Consider a situation where customer Web sites are hosted by a popular Web hosting company and/or Internet Service Provider (ISP). The Web content is relatively static and is kept on a single NFS server for easy administration. As the customer base increases, the number of simultaneous Web connection requests also increases.

Figure 26: Web Hosting Configuration Without SLB

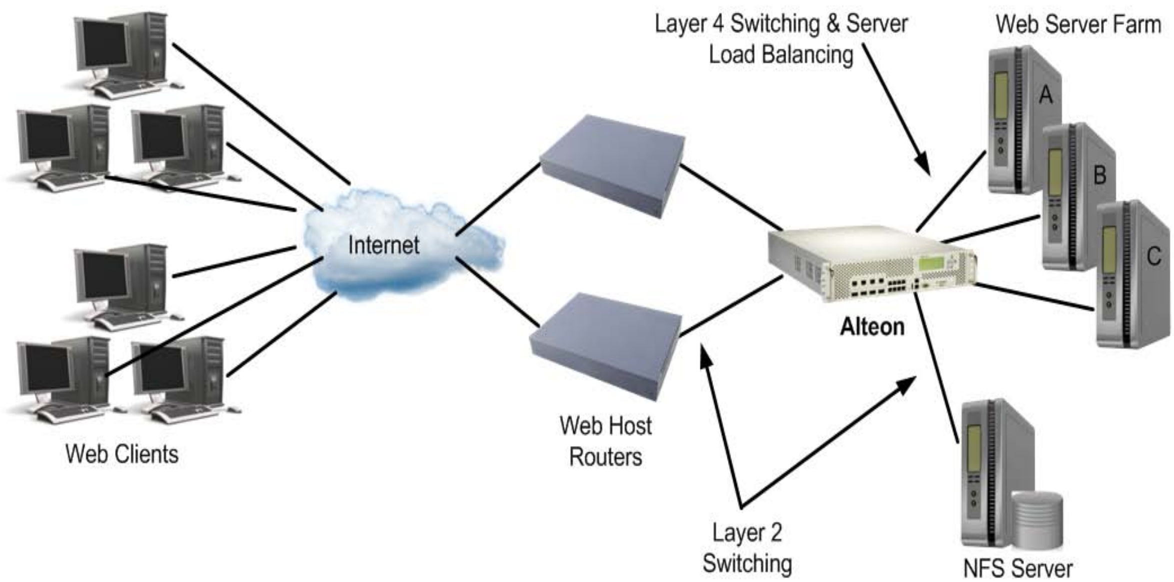


Such a company has three primary needs:

- Increased server availability
- Server performance scalable to match new customer demands
- Easy administration of network and servers

All of these issues can be addressed by adding an Alteon with SLB software, as shown in Figure 27 - Web Hosting with SLB Solutions, page 169:

Figure 27: Web Hosting with SLB Solutions



SLB accomplishes the following:

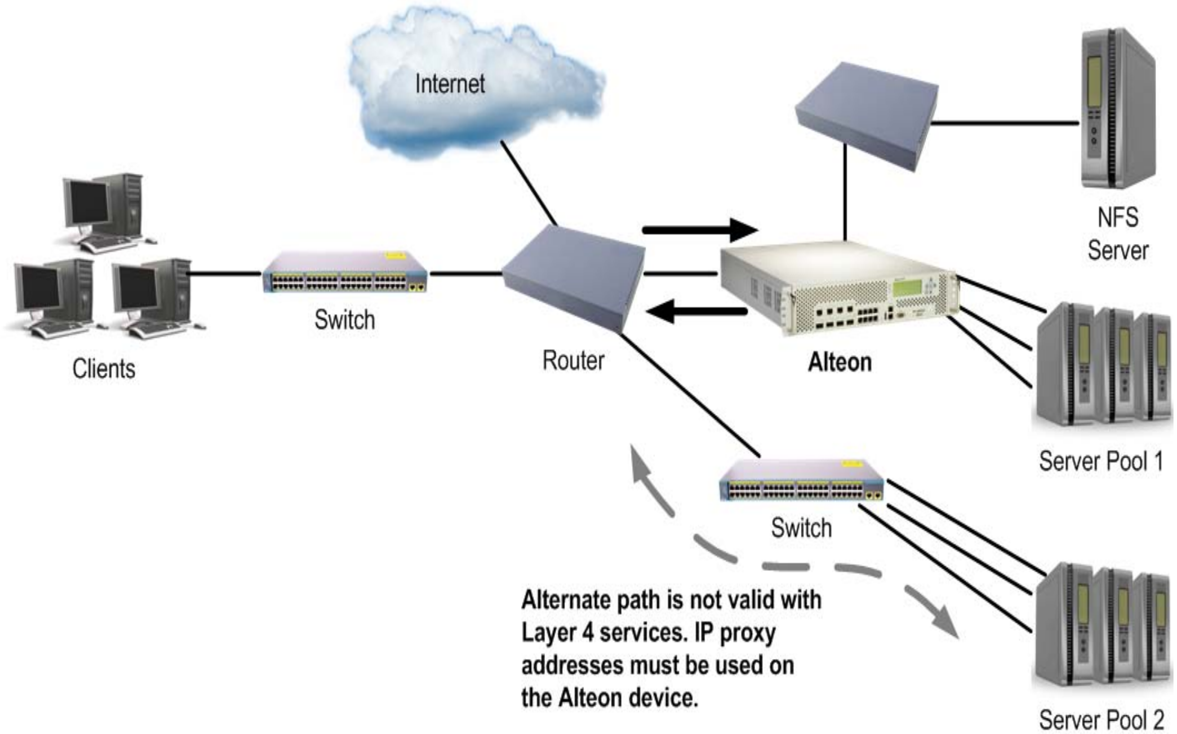
- Reliability is increased by providing multiple paths from the clients to Alteon and by accessing a pool of servers with identical content. If one server fails, the others can take up the additional load.
- Performance is improved by balancing the Web request load across multiple servers. More servers can be added at any time to increase processing power.
- For ease of maintenance, servers can be added or removed dynamically, without interrupting shared services.

Network Topology Requirements

When deploying SLB, there are a few key aspects to consider:

- In standard SLB, all client requests to a virtual server IP address and all responses from the real servers must pass through Alteon, as shown in Figure 28 - SLB Client/Server Traffic Routing, page 170. If there is a path between the client and the real servers that does not pass through Alteon, Alteon can be configured to proxy requests in order to guarantee that responses use the correct path (see Client Network Address Translation (Proxy IP), page 190).

Figure 28: SLB Client/Server Traffic Routing

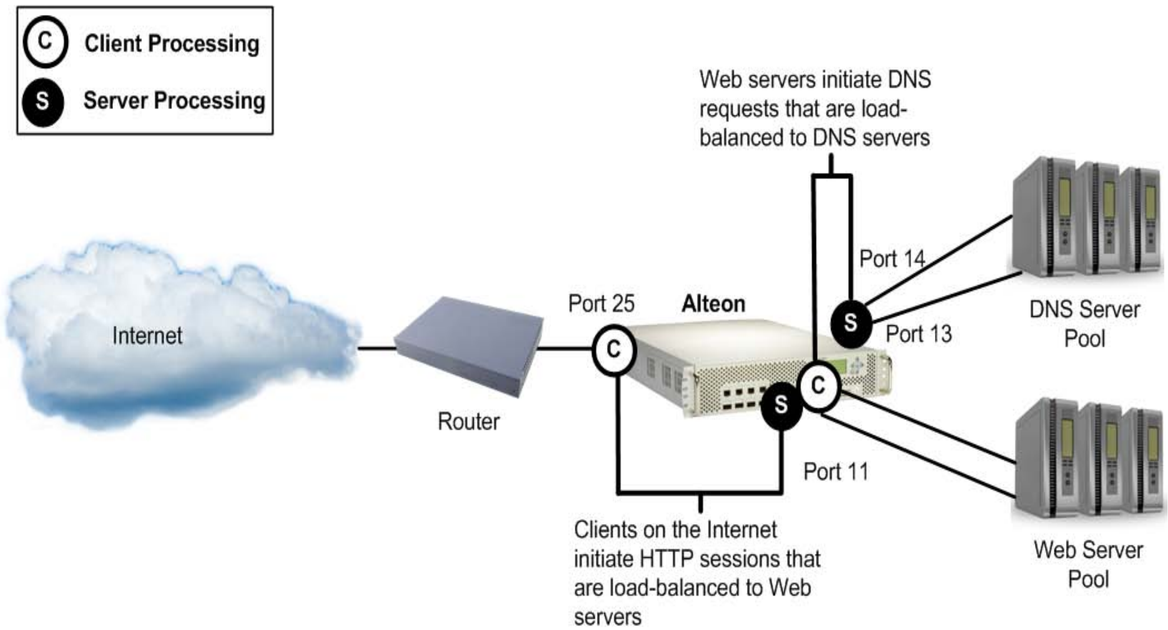


- Identical content must be available to each server in the same pool. Either of the following methods can be used:
  - Static applications and data are duplicated on each real server in the pool.
  - Each real server in the pool has access to the same data through use of a shared file system or back-end database server.
- Some services require that a series of client requests go to the same real server so that session-specific state data can be retained between connections. Services of this nature include Web search results, multi-page forms that the user fills in, or custom Web-based applications typically created using *cgi-bin* scripts. Connections for these types of services must be configured as *persistent* (see Persistence, page 583), or must use the minmisses, hash, phash metrics (see Metrics for Real Server Groups, page 180).
- Clients and servers can be connected through the same Alteon port. Each port in use can be configured to process client requests, server traffic, or both. You can enable or disable processing on a port independently for each type of Layer 4 traffic:
  - **Layer 4 client processing**—Ports that are configured to process client request traffic provide address translation from the virtual server IP to the real server IP address.
  - **Layer 4 server processing**—Ports that are configured to process server responses to client requests provide address translation from the real server IP address to the virtual server IP address. These ports require real servers to be connected to Alteon directly or through a hub, router, or another switch.

Note: Ports configured for Layer 4 client/server processing can simultaneously provide Layer 2 switching and IP routing functions.

The following is an example network topology:

Figure 29: Example Network for Client/Server Port Configuration



Alteon load balances traffic to a Web server pool and to a Domain Name System (DNS) server pool. The port connected to the Web server pool (port 11) is instructed to perform both server and client processing.

Server Load Balancing Configuration Basics

This section describes the steps for configuring an SLB Web hosting solution. In this procedure, many of the SLB options are left to their default values. For more options, see Implementing Server Load Balancing, page 167. Before you start configuring, you must be connected to the CLI as the administrator.

Note: For details about any of the menu commands described in this example, refer to the *Alteon Application Switch Operating System Command Reference.*

1. Assign an IP address to each of the real servers in the server pool.

   The real servers in any given real server group must have an IP route to Alteon that performs the SLB functions. This IP routing is most easily done by placing Alteons and servers on the same IP subnet, although advanced routing techniques can be used as long as they do not violate the topology rules outlined in Network Topology Requirements, page 169.
2. Define an IP interface on Alteon.

   Alteon must have an IP route to all of the real servers that receive switching services. For SLB, Alteon uses this path to determine the level of TCP/IP reach of the real servers.

   To configure an IP interface for this example, enter these commands from the CLI:

`>> # /cfg/l3/if 1`	(Select IP Interface 1)
`>> IP Interface 1# addr 200.200.200.100`	(Assign IP address for the interface)
`>> IP Interface 1# ena`	(Enable IP Interface 1)

Note: The IP interface and the real servers must belong to the same VLAN, if they are in the same subnet. This example assumes that all ports and IP interfaces use default VLAN 1, requiring no special VLAN configuration for the ports or IP interface.

3. Define each real server. For each real server, you must assign a real server number, specify its actual IP address, and enable the real server.

`>> IP Interface 1# /cfg/slb/real 1`	(Server A is Real Server 1)
`>> Real server 1# rip 200.200.200.2`	(Assign Server A IP address)
`>> Real server 1# ena`	(Enable Real Server 1)
`>> Real server 1# /cfg/slb/real 2`	(Server B is Real Server 2)
`>> Real server 2# rip 200.200.200.3`	(Assign Server B IP address)
`>> Real server 2# ena`	(Enable Real Server 2)
`>> Real server 2# /cfg/slb/real 3`	(Server C is Real Server 3)
`>> Real server 3# rip 200.200.200.4`	(Assign Server C IP address)
`>> Real server 3# ena`	(Enable Real Server 3)

4. Define a real server group and add the three real servers to the service group.

`>> Real server 3# /cfg/slb/group 1`	(Select Real Server group 1)
`>> Real server group 1# add 1`	(Add Real Server 1 to group 1)
`>> Real server group 1# add 2`	(Add Real Server 2 to group 1)
`>> Real server group 1# add 3`	(Add Real Server 3 to group 1)

5. Define a virtual server. All client requests are addressed to a virtual server IP address on a virtual server defined on Alteon. Clients acquire the virtual server IP address through normal DNS resolution. In this example, HTTP is configured as the only service running on this virtual server, and this service is associated with the real server group.

`>> Real server group 1# /cfg/slb/virt 1`	(Select Virtual Server 1)
`>> Virtual server 1# vip 200.200.200.1`	(Assign a virtual server IP address)
`>> Virtual server 1# ena`	(Enable the virtual server)
`>> Virtual server 1#service http`	(Select the *HTTP service* menu)
`>> Virtual server 1 http Service# group 1`	(Associate virtual port to real group)

Note: This configuration is not limited to the HTTP Web service. Other TCP/IP services can be configured in a similar fashion. For a list of other well-known services and ports, see Table 20. To configure multiple services, see Multiple Services per Real Server, page 177.

6. Define the port settings. In this example, the following ports are being used on Alteon:

Table 19: Web Host Example: Port Usage

Port	Host	L4 Processing
1	Server A serves SLB requests.	Server
2	Server B serves SLB requests.	Server
3	Server C serves SLB requests.	Server
4	Back-end NFS server provides centralized content for all three real servers. This port does not require switching features.	None
5	Client router A connects Alteon to the Internet where client requests originate.	Client
6	Client router B connects Alteon to the Internet where client requests originate.	Client

The ports are configured as follows:

`>> Virtual server 1# /cfg/slb/port 1`	(Select physical port 1)
`>> SLB port 1# server ena`	(Enable server processing on port 1)
`>> SLB port 1# /cfg/slb/port 2`	(Select physical port 2)
`>> SLB port 2# server ena`	(Enable server processing on port 2)
`>> SLB port 1# /cfg/slb/port 3`	(Select physical port 3)
`>> SLB port 3# server ena`	(Enable server processing on port 3)
`>> SLB port 5# /cfg/slb/port 5`	(Select physical port 5)
`>> SLB port 5# client ena`	(Enable client processing on port 1)
`>> SLB port 6# /cfg/slb/port 6`	(Select physical port 6)
`>> SLB port 6# client ena`	(Enable server processing on port 6)

7. Enable, apply, and verify the configuration.

`>> SLB port 6# /cfg/slb`	(Select the *SLB* Menu)
`>> Layer 4# on`	(Turn SLB on)
`>> Layer 4# apply`	(Make your changes active)
`>> Layer 4#cur`	(View current settings)

Examine the resulting information. If any settings are incorrect, make the appropriate changes.

8. Save your new configuration changes.

```
>> Layer 4# save
```

Note: You must **apply** any changes in order for them to take effect, and you must **save** changes if you want them to remain in effect after reboot.

9. Check the SLB information.

```
>> Layer 4# /info/slb/dump
```

Check that all SLB parameters are working as expected. If necessary, make any appropriate configuration changes and then check the information again.

Physical and Logical Real Server Modes

Alteon supports multiple real servers having the same IP address. To do this, you can define numerous "physical" or "logical" real servers, all with the same IP address associated with the same real, physical server.

Such real servers must either have different rports configured or associated to different server groups, enabling Alteon to differentiate the destination ports on the server.

When using logical servers, you must enable DAM on the virtual service to which a logical server is attached, or you must enable PIP for that logical server.

PIP is enabled for a server when PIP mode is enabled at the server level, and a PIP address is configured either at server level or at virtual service level (when PIP mode is set to **Ingress** or **Egress**, PIP must be configured at the port or VLAN level).

This feature provides greater flexibility in a number of the Alteon SLB operations:

- **Layer 7 content switching to specific application port**—If you have multiple HTTP applications running on the same real server differentiated by the listening port on the server, the applications are identified by HTTP (Layer 7) content switching rules that review requesting URL content to determine destination application port.

  Alteon lets you define different real servers with the same IP address and different ports where every HTTP application is configured on a separate real server with its own ports, all with the same IP address. The real servers are associated with groups, each dedicated to a Layer 7 content switching rule on the virtual service.
- **Health check**—Lets you configure scripted health checks for a server with multiple ports.
- **Maximum connections**
  - **Physical server**—If you need to limit the maximum number of connections per physical server (maximum TCP capacity), you can define multiple real servers with the same IP address and set each real server mode to **physical** and its maximum connections (**maxcon**) to the required value.
  - **Logical server**—If you need to limit the maximum number of connections per logical server running on the same physical server, you can define multiple real servers with the same IP address and set each real mode to **logical** and its maximum connections (**maxcon**) to the required value.

  You can also set the max connections mode to **physical** (default) or **logical**. Real servers with the same IP address must be set to the same maxcon connection mode.

  Real servers with the same IP address set to maximum connection mode **physical** must all have the same maxcon value. The maxcon value is the maximum number of connections that the real servers both support.

  Real servers with the same IP address set to maximum connection mode **logical** can each have different maxcon values. The maxcon value is the maximum number of connections that each logical real server supports individually.
- **Logical server weight**—The **weight** parameter is defined only per real server and not per port. To prioritize multiple logical servers (daemons) with different processing requirements, you can define multiple real servers, with different rports or in different groups, all with the same IP address. You can then set each real server weight to its desired value.

- **Client NAT behavior**—For an IIS server running multiple logical servers, some applications (such as HTTP) need the client IP addresses to be masked using Proxy-IP/Client-NAT and perform persistency using other methods (cookies) or allow multiplexing to be used to improve the server's efficiency, and other applications (such as FTP) require that the client IP addresses not be masked to allow client-IP persistency. Using a logical server proxy mode, you can define multiple real servers with desired rports (or in different groups), all with the same IP address, and set each real server proxy mode to its desired value.

Notes

- DAM must be turned on or a proxy must be used to support multiple real servers with the same IP address.
- Multiple real servers with the same IP address cannot share the same addports. Multiple real servers with the same IP address with no addport configured must be associated to different server groups.

Supported Services and Applications

Each virtual server can be configured to support up to eight services, limited to a total of 1023 services per Alteon. Using the **`/cfg/slb/virt <virtual server number> /service`** option, the following TCP/UDP applications can be specified:

Note: The service number specified on Alteon must match the service specified on the server.

Table 20: Well-Known Application Ports

Number	TCP/UDP Application	Number	TCP/UDP Application	Number	TCP/UDP Application
20	ftp-data	79	finger	179	bgp
21	ftp	80	http	194	irc
22	ssh	109	pop2	389	ldap
23	telnet	110	pop3	443	https
25	smtp	119	nntp	520	rip
37	time	123	ntp	554	rtsp
42	name	143	imap	1812	RADIUS
43	whois	144	news	1813	RADIUS Accounting
53	domain	161	snmp	1985	hsrp
69	tftp	162	snmptrap		

Notes

- Load balancing some applications (such as FTP and RTSP) require special configuration. For more information, see Load Balancing Special Services, page 279.
- For all applications without a well-known port, you can select **Basic-SLB** as the application.

Disabling and Enabling Real Servers

If you need to reboot a server, ensure that new sessions are not sent to the real server and that current sessions are not discarded before shutting down the server, using one of the following methods:

- Use the following command with the **n** (none) option to suspend connection assignments to the real server:

```
>> # /oper/slb/dis <real server number> n
```

  When the current session count on your server falls to zero, you can shut down your server.

- If you have configured persistence on the real server, use the following command with the **p** (persistent) option to suspend connection assignments (except for persistent http 1.0 sessions) to the real server:

```
>> # /oper/slb/dis <real server number> p
```

  When the current session count on your server falls to zero and when persistent sessions for the real server have aged out (refer to the persistence parameters you have set for this real server), you can shut down your server. For more information, see Persistence, page 583.

- When maintenance is complete, use the following command to enable the real server:

```
>> # /oper/slb/ena <real server number>
```

  Alteon resumes assignment of connections to this real server immediately.

Table 21 compares the behavior of the **>> # /oper/slb/dis** and **>> # /cfg/slb/dis** commands:

Table 21: Disabling Commands Behavior

Behavior	>> # /oper/slb/dis	>> # /cfg/slb/dis
Clearing all old sessions immediately after executing command	No	Yes
Allowing persistent HTTP 1.0 sessions	Yes/No	N/A

The grace option is enabled only if the real server is in "failed" state and not in "disabled" state (failed by health check). For example, consider HTTP service when the grace option is enabled. After handling client requests for some time, the real server is marked failed by the health check, but the remaining sessions to the real server are still kept to maintain previous connections from client to the real server.

Health Checks for Real Servers

Determining the health for each real server is a basic function for SLB. By default, Alteon checks the health of a real server using ICMP.

Once servers are attached to groups which, in turn, are attached to services, Alteon checks the availability of the services running on the server using the health checks configured for the group. However, it is possible to override this behavior and configure for each real server its own health checks.

Alteon checks the availability of real servers using timers defined in the health check. However, it is possible to override timers per real server. The following example illustrates how the health check interval and the number of retries can be changed:

`>> # /cfg/slb/real <real server number>`	(Select the real server)
`>> Real server# inter 4`	(Check real server every 4 seconds)
`>> Real server# retry 6`	(Declare down after 6 checks fail)

For details on health monitoring capabilities, see Health Checking, page 481.

Multiple Services per Real Server

When you configure multiple services in the same group, their health checks are dependent on each other. If a real server fails a health check for a service, then the status of the real server for the second service appears as “blocked.”

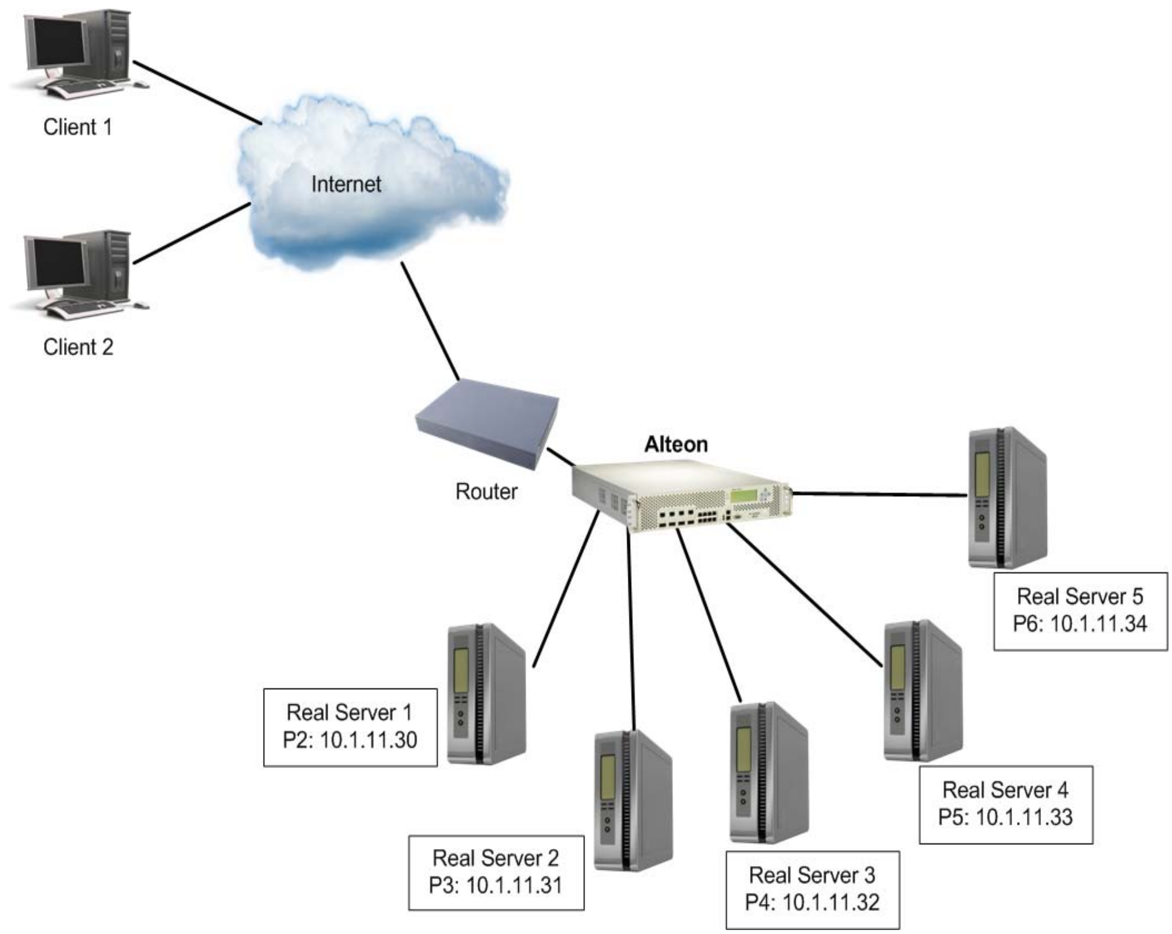
- **Independent Services**—If you are configuring two independent services such as FTP and SMTP, where the real server failure on one service does not affect other services that the real server supports, then configure two groups with the same real servers, but with different services. If a real server configured for both FTP and SMTP fails FTP, the real server is still available for SMTP. This allows the services to act independently even though they are using the same real servers.
- **Dependent Services**—If you are configuring two dependent services such as HTTP and HTTPS, where the real server failure on one service blocks the real server for other services, then configure a single group with multiple services. If a real server configured for both HTTP and HTTPS fails for the HTTP service, then the server is blocked from supporting any HTTPS requests. Alteon blocks HTTPS requests, (even though HTTPS has not failed) until the HTTP service becomes available again. This helps in troubleshooting so you know which service has failed.

Buddy Server Health Checks

Alteon enables the administrator to tie the health of a real server to another real server. This real server can be in the same real server group, but also can be in a separate group. In this configuration, a real server is only considered healthy if the buddy it is associated with is also healthy.

Figure 30 - Example Buddy Server Health Check Configuration, page 178 illustrates an example network topology using Buddy Server health checking:

Figure 30: Example Buddy Server Health Check Configuration



To add a real server as a buddy server for another real server

```
>> Main# /cfg/slb/real <real server number> /adv/buddyhc/addbd <real server
number> <real server group> <service>
```

To remove a real server as a buddy server

```
>> Main# /cfg/slb/real <real server number> /adv/buddyhc/delbd <real server
number> <real server group> <service>
```

To view the current buddy server settings for a real server

```
>> Main# /cfg/slb/real <real server number> /adv/buddyhc/cur
```

To configure buddy server health checking

1. Configure an interface.

```
>>Main# /cfg/l3/if 1/addr 10.1.11.1/mask 255.255.255.0/ena
```

2. Enable SLB

```
>>Main# /cfg/slb/on
```

3. Enable ports for SLB.

```
>> Main# /cfg/slb/port 2/server en
>> Main# /cfg/slb/port 3/server en
>> Main# /cfg/slb/port 4/server en
>> Main# /cfg/slb/port 5/server en
>> Main# /cfg/slb/port 6/server en
```

4. Configure and enable real servers.

```
>> Main# /cfg/slb/real 1/rip 10.1.11.30/ena
>> Main# /cfg/slb/real 2/rip 10.1.11.31/ena
>> Main# /cfg/slb/real 3/rip 10.1.11.32/ena
>> Main# /cfg/slb/real 4/rip 10.1.11.33/ena
>> Main# /cfg/slb/real 5/rip 10.1.11.34/ena
```

5. Configure real server groups and assign real servers to them.

```
>> Main# /cfg/slb/group 2/add 2
>> Main# /cfg/slb/group 2/add 3
>> Main# /cfg/slb/group 2/add 4
>> Main# /cfg/slb/group 2/add 5
>> Main# /cfg/slb/group 2/health tcp
```

6. Apply and save the configuration

```
>> Main# Apply
>> Main# Save
```

7. Configure virtual servers and enable HTTP service.

```
>> Main # /cfg/slb/virt 1/vip 120.10.10.10/ena
>> Main # /cfg/slb/virt 1/service http
>> Main # /cfg/slb/virt 1/service http/group 1
>> Main # /cfg/slb/virt 2/vip 120.10.10.11/ena
>> Main # /cfg/slb/virt 2/service http
>> Main # /cfg/slb/virt 2/service http/group 2
```

8. Add Real Servers 2 to 5 in Group 2 to Real Server 1 as buddy servers.

```
>> Main# /cfg/slb/real 1/adv/buddyhc/addbd 2 2 80
>> Main# /cfg/slb/real 1/adv/buddyhc/addbd 3 2 80
>> Main# /cfg/slb/real 1/adv/buddyhc/addbd 4 2 80
>> Main# /cfg/slb/real 1/adv/buddyhc/addbd 5 2 80
```

9. Apply and save configuration.

```
>> Main# Apply
>> Main# Save
```

Note: It is not mandatory for a buddy server group to be part of any virtual service.

Metrics for Real Server Groups

Metrics are used for selecting which real server in a group receives the next client connection. This section includes:

- Changing the Real Server Group Metric, page 180
- The available metrics, including:
  - Minimum Misses, page 181
  - Hash, page 181
  - Persistent Hash, page 182
  - Tunable Hash, page 182
  - Weighted Hash, page 182
  - Least Connections, page 182
  - Least Connections Per Service, page 182
  - Round-Robin, page 182
  - Response Time, page 183
  - Bandwidth, page 183

Changing the Real Server Group Metric

The default metric is least connections (leastconns). You can change the metric using the **metric** command, as shown in the following example:

```
>> # /cfg/slb/group <group number>
>> Real server group# metric minmisses
```

Minimum Misses

The minmisses metric is optimized for cache redirection. It uses IP address information in the client request to select a server. When selecting a server, Alteon calculates a value for each available real server based on the relevant IP address information. The server with the highest value is assigned the connection. This metric attempts to minimize the disruption of persistency when servers are removed from service. This metric should be used only when persistence is required.

By default, the minmiss algorithm uses the upper 24 bits of the source IP address to calculate the real server that the traffic should be sent to when the minmisses metric is selected. Alteon allows the selection of all 32 bits of the source IP address to hash to the real server.

The source or destination IP address information used depends on the application:

- For application redirection, the client destination IP address is used. All requests for a specific IP destination address are sent to the same server. This metric is particularly useful in caching applications, helping to maximize successful cache hits. Best statistical load balancing is achieved when the IP address destinations of load-balanced frames are spread across a broad range of IP subnets.
- For SLB, the client source IP address and real server IP address are used. All requests from a specific client are sent to the same server. This metric is useful for applications where client information must be retained on the server between sessions. With this metric, server load becomes most evenly balanced as the number of active clients with different source or destination addresses increases.

  To select all 32 bits of the source IP address, use the command, **`/cfg/slb/group x/mhash 32`**. This 32-bit hash is most useful in the wireless world.

The minmisses metric cannot be used for Firewall Load Balancing (FWLB), since the real server IP addresses used in calculating the score for this metric are different on each side of the firewall.

Hash

The hash metric uses IP address information in the client request to select a server. The specific IP address information used depends on the application:

- For application redirection, the client destination IP address is used. All requests for a specific IP destination address are sent to the same server. This is particularly useful for maximizing successful cache hits.
- For SLB, the client source IP address is used. All requests from a specific client are sent to the same server. This option is useful for applications where client information must be retained between sessions.
- For FWLB, both the source and destination IP addresses are used to ensure that the two unidirectional flows of a given session are redirected to the same firewall.

When selecting a server, a mathematical hash of the relevant IP address information is used as an index into the list of currently available servers. Any given IP address information will always have the same hash result, providing natural persistence, as long as the server list is stable. However, if a server is added to or leaves the set, then a different server might be assigned to a subsequent session with the same IP address information even though the original server is still available. Open connections are not cleared. The phash metric can be used to maintain stable server assignment. For more information, see Persistent Hash, page 182.

Note: The hash metric provides more distributed load balancing than minmisses at any given instant. It should be used if the statistical load balancing achieved using minmisses is not as optimal as desired. If the load-balancing statistics with minmisses indicate that one server is processing significantly more requests over time than other servers, consider using the phash metric.

Persistent Hash

The phash metric provides the best features of hash and minmisses metrics together. This metric provides stable server assignments like the minmiss metric and even load distribution like the hash metric.

When you select the phash metric for a group, a baseline hash is assumed based on the configured real servers that are enabled for the group. If the server selected from this baseline hash is unavailable, then the old hash metric is used to find an available server.

If all the servers are available, then phash operates exactly like hash. When a configured server becomes unavailable, clients bound to operational servers will continue to be bound to the same servers for future sessions and clients bound to unavailable servers are rehashed to an operational server using the old hash metric.

When more servers go down with phash, you will not have an even load distribution as you would with the standard hash metric.

Tunable Hash

By default, the hash metric uses the client's source IP address as the parameter for directing a client request to a real server. In environments where multiple users are sharing the same proxy, resulting in the same source IP address, a load-balancing hash on the source IP address directs all users to the same real server.

Tunable hash allows the user to select the parameters (source IP, or source IP and source port) that are used when hashing is chosen as the load-balancing metric.

Weighted Hash

Weighted hash allows real server weighting to be used in conjunction with the hash load balancing algorithm. If the configured real server weight is greater than 1, the real server weight is taken into account during the load-balancing calculation. There are no CLI commands to configure or change the weighted hash state.

Least Connections

The default metric is leastconns. With the leastconns metric, the number of connections currently open on each real server is measured in real time. The server with the fewest current connections is considered to be the best choice for the next client connection request.

This option is the most self-regulating, with the fastest servers typically getting the most connections over time.

Least Connections Per Service

The svcleast (least connections per service) metric is an extension of the leastconns metric. When using this metric, Alteon selects the real server based only on the number of active connections for the service which is load balanced, and not the total number of connections active on the server. For example, when selecting a real server for a new HTTP session, a real server serving one HTTP connection and 20 FTP connections takes precedence over a real server serving two HTTP connections only.

Round-Robin

With the roundrobin metric, new connections are issued to each server in turn. This means that the first real server in the group gets the first connection, the second real server gets the next connection, followed by the third real server, and so on. When all the real servers in this group have received at least one connection, the issuing process starts over with the first real server.

Response Time

The response metric uses the real server response time to assign sessions to servers. The response time between the servers and Alteon is used as the weighting factor. Alteon monitors and records the amount of time it takes for each real server to reply to a health check to adjust the real server weights. The weights are adjusted so they are inversely proportional to a moving average of response time. In such a scenario, a server with half the response time as another server receives a weight twice as large.

Note: The effects of the response weighting apply directly to the real servers and are not necessarily confined to the real server group. When response time-metered real servers are also used in other real server groups that use the leastconns, roundrobin, or (weighted) hash metrics, the response weights are applied on top of the metric method calculations for the affected real servers. Since the response weight changes dynamically, this can produce fluctuations in traffic distribution for the real server groups that use these metrics.

Bandwidth

The bandwidth metric uses real server octet counts to assign sessions to a server. Alteon monitors the number of octets sent between the server and Alteon. The real server weights are then adjusted so they are inversely proportional to the number of octets that the real server processes during the last interval.

Servers that process more octets are considered to have less available bandwidth than servers that have processed fewer octets. For example, the server that processes half the amount of octets over the last interval receives twice the weight of the other servers. The higher the bandwidth used, the smaller the weight assigned to the server. Based on this weighting, the subsequent requests go to the server with the highest amount of free bandwidth. These weights are assigned.

The bandwidth metric requires identical servers with identical connections.

Note: The effects of the bandwidth weighting apply directly to the real servers and are not necessarily confined to the real server group. When bandwidth-metered real servers are also used in other real server groups that use the leastconns, roundrobin, or (weighted) hash metrics, the bandwidth weights are applied on top of the metric method calculations for the affected real servers. Since the bandwidth weight changes dynamically, this can produce fluctuations in traffic distribution for the real server groups that use the above metrics.

Group Availability Threshold

This feature lets you set the thresholds that define changes to a group's availability:

- **Down threshold** (minimum threshold)—When the number of active real servers reaches this threshold, the group status changes to **down**.
- **Restore threshold** (maximum threshold)—When the number of active real servers reaches this threshold, the group status changes to **up**.

Example

A group has 10 real servers, the down threshold is 3, and the restore threshold is 5.

- As long as there are more than three real servers active, the group is **up**.
- If any of the group's real servers fail and the number of active servers reaches three, the group's status changes to **down**.

- If the group is **down**, if the number of active real servers only goes up to four, the status remains **down**.
- When the number of active real servers is five or more, the group status changes to **up**.

These values are set using the `/cfg/slb/group/minthrsh` and `/cfg/slb/group/maxthrsh` commands. For more information, see the *Alteon Application Switch Operating System Command Reference*.

Weights for Real Servers

Weights can be assigned to each real server. These weights can bias load balancing to give the fastest real servers a larger share of connections. Weight is specified as a number from 1 to 48. Each increment increases the number of connections the real server gets. By default, each real server is given a weight setting of 1. A setting of 10 would assign the server roughly 10 times the number of connections as a server with a weight of 1.

To set weights

`>> # /cfg/slb/real <real server number>`	(Select the real server)
`>> Real server# weight 10`	(10 times the number of connections)

The effects of the bandwidth weighting apply directly to the real servers and are not necessarily confined to the real server group. When bandwidth-metered real servers are also used in other real server groups that use the leastconns or roundrobin metrics, the bandwidth weights are applied on top of the leastconns or roundrobin calculations for the affected real servers. Since the bandwidth weight changes dynamically, this can produce fluctuations in traffic distribution for the real server groups that use the leastconns or roundrobin metrics.

Readjusting Server Weights Based on SNMP Health Check Response

Alteon can be configured to dynamically change weights of real servers based on a health check response using the Simple Network Management Protocol (SNMP). To enable dynamic assignment of weights based on the response to an SNMP health check, enter the following commands:

`>> # /cfg/slb/adv/snmphc < SNMP health script number>`	
`>> SNMP Health Check 1# weight e`	(Enable weighting via SNMP health check)

For more information on configuring SNMP health checks, see SNMP Health Check, page 488.

Connection Time-Outs for Real Servers

In some cases, open TCP/IP sessions might not be closed properly (for example, Alteon receives the SYN for the session, but no FIN is sent). If a session is inactive for 10 minutes (the default), it is removed from the session table.

To change the time-out period

`>> # /cfg/slb/real <real server number>`	(Select the real server)
`>> Real Server# tmout 4`	(Specify an even numbered interval)

This example changes the time-out period of all connections on the designated real server to 4 minutes.

Maximum Connections for Real Servers

You can set the number of open connections each real server is allowed to handle for SLB. To set the connection limit, enter the following:

`>> # /cfg/slb/real <real server number>`	(Select the real server)
`>> Real Server 1 # maxcon 1600`	(Allow 1600 connections maximum)

`Current max connections mode [logical]:`
`Enter new max connections mode`
`[physical/logical][logical]:`	

Values average from approximately 500 HTTP connections for slower servers to 1500 for quicker, multiprocessor servers. The appropriate value also depends on the duration of each session and how much CPU capacity is occupied by processing each session. Connections that use many Java or CGI scripts for forms or searches require more server resources and thus a lower maxcon limit. You may want to use a performance benchmark tool to determine how many connections your real servers can handle.

When a server reaches its maxcon limit, Alteon no longer sends new connections to the server. When the server drops back below the maxcon limit, new sessions are again allowed.

You can also set the max connections mode to **physical** (default) or **logical**. Real servers with the same IP address must be set to the same maxcon connection mode.

- Real servers with the same IP address set to maximum connection mode **physical** must all have the same maxcon value. The maxcon value is the maximum number of connections that the real servers both support.
- Real servers with the same IP address set to maximum connection mode **logical** can each have different maxcon values. The maxcon value is the maximum number of connections that each logical real server supports individually.

Unlimited Connections to Real Servers

This feature allows for an unlimited number of connections to be allocated to traffic accessing a real server. Alteon allows for a range of 0 to 200000 connections per real server. A maxcon value of 0 allows the specified real server to handle up to its (or Alteon's) maximum number of connections.

To configure unlimited connections

1. Set the real server maxcon value to zero.

```
>> # Main# /cfg/slb/Real Server 700 # maxcon
Current max connections: 200000, physical
Max connections 0 means unlimited connections
Enter new max connections (0-200000)[200000]:   0
Current max connections mode: logical
Enter new max connections mode [physical/logical][logical]:
```

2. Apply and save the configuration.

```
>> # apply
>> # save
```

Backup/Overflow Servers

A real server can back up other real servers and can handle overflow traffic when the maximum connection limit is reached. Each backup real server must be assigned a real server number and real server IP address. It must then be enabled. Finally, the backup server must be assigned to each real server that it will back up.

Example Define Real Server 4 as a backup/overflow for Real Servers 1 and 2

`>> # /cfg/slb/real 4`	(Select Real Server 4 as backup)
`>> Real server 4# rip 200.200.200.5`	(Assign backup IP address)
`>> Real server 4# ena`	(Enable Real Server 4)
`>> Real server 4# /cfg/slb/real 1`	(Select Real Server 1)
`>> Real server 1# backup 4`	(Real Server 4 is the backup for 1)
`>> Real server 1# /cfg/slb/real 2`	(Select Real Server 2)
`>> Real server 2# backup 4`	(Real Server 4 is the backup for 2)
`>> Real server 2# overflow enabled`	(Overflow enabled)

Example Assign a backup/overflow server to a real server group

Similarly, a backup/overflow server can be assigned to a real server group. If all real servers in a real server group fail or overflow, the backup comes online.

>> # /cfg/slb/group <real server group number>	(Select Real Server group)
>> Real server group# backup r4	(Assign Real Server 4 as backup)

Example Real server groups using another real server group for backup/overflow

>> # /cfg/slb/group <real server group number>	(Select Real Server group)
>> Real server group# backup g2	(Assign Real Server Group 2 as backup)

Backup Only Server

Unlike a Backup/Overflow server, a Backup Only server is used to *only* backup real servers, and not provide an overflow capability. This enforces maximum session capacity while still providing resiliency. In this configuration, if the primary server reaches its maximum session capacity, the backup server does not take over sessions from the primary server. The backup server only comes into play if the primary server fails.

Example Define Real Server 4 as a backup only server for Real Servers 1 and 2

>> # /cfg/slb/real 4	(Select Real Server 4 as backup)
>> Real server 4# rip 200.200.200.5	(Assign backup IP address)
>> Real server 4# ena	(Enable Real Server 4)
>> Real server 4# /cfg/slb/real 1	(Select Real Server 1)
>> Real server 1# backup 4	(Real Server 4 is backup for 1)
>> Real server 1# /cfg/slb/real 2	(Select Real Server 2)
>> Real server 2# backup 4	(Real Server 4 is backup for 2)

Example Assign a backup/overflow server to a real server group

Similarly, a backup/overflow server can be assigned to a real server group. If all real servers in a real server group fail the backup comes online.

>> # /cfg/slb/group <real server group number>	(Select real server group)
>> Real server group# backup r4	(Assign Real Server 4 as backup)

Example Real server groups using another real server group for backup

`>> # /cfg/slb/group <real server group number>`	(Select real server group)
`>> Real server group# backup g2`	(Assign Real Server Group 2 as backup)

Backup Preemption

Alteon supports control preemption of backup when a primary server becomes active.

To enable or disable backup preemption

```
/cfg/slb/real <server number>/preempt e|d
```

By default, preempt is enabled. When the primary server becomes active, it displaces the backup server and takes control. When preempt is disabled, the backup server continues processing requests sent by Alteon even if the primary server becomes active. During this process, the primary server is operationally disabled and becomes active only if the backup server goes down.

In the following example, Real Server 4 is configured as backup for Real Server 1, and preemption is disabled in Real Server 1:

`>> # /cfg/slb/real 4`	(Select Real Server 4 as backup)
`>> Real server 4# rip 200.200.200.5`	(Assign backup IP address)
`>> Real server 4# ena`	(Enable Real Server 4)
`>> Real server 4# /cfg/slb/real 1`	(Select Real Server 1)
`>> Real server 1# backup 4`	(Real Server 4 is backup for 1)
`>> Real server 1# preempt dis`	(Disable preemption ability of real 1))

Server Slow Start

Server slow start is an optional service that can be implemented on new real servers. The primary purpose of this service is to avoid sending a high rate of new connections to a new server. When the slow start begins, traffic is throttled and increased gradually until server initialization is complete. Server slow start is controlled by setting a time limit that determines the length of the slow start period. Server slow start begins when any of the following occur:

- Server comes online
- A new real server is added and comes online
- Multiple real servers are in a slow start mode

Server slow start ends when any of the following occur:

- The server slow start time limit expires
- The new real server metric weight reaches its target value

Server slow start is enabled on a group level to all real servers in that group.

Example Enable slow start for Real Server Group 1 with a setting of 10 seconds

```
>> Main# /cfg/slb/group 1/slowstr 10
```

Example Disable slow start for Real Server Group 1

```
>> Main# /cfg/slb/group 1/slowstr 0
```

Extending Server Load Balancing Topologies

For standard SLB, all client-to-server requests to a particular virtual server and all related server-to-client responses must pass through the same Alteon. In complex network topologies, routers and other devices can create alternate paths around Alteon managing SLB functions. Under such conditions, the Alteon provides the following solutions:

- Virtual Matrix Architecture, page 189
- Client Network Address Translation (Proxy IP), page 190
- Mapping Ports, page 194
- Direct Server Return, page 197
- Direct Access Mode, page 200
- Delayed Binding, page 203

Virtual Matrix Architecture

Virtual Matrix Architecture (VMA) is a hybrid architecture that takes full advantage of the distributed processing capability in Alteon. With VMA, Alteon makes optimal use of system resources by distributing the workload to multiple processors, thereby improving performance and increasing session capacity. VMA also removes the topology constraints introduced by using Direct Access Mode (DAM). By default, VMA is enabled (**`/cfg/slb/adv/matrix`**).

To improve the distribution, there are two VMA configurable options, as shown in Table 22:

Table 22: VMA Configurable Options

Option	Description
VMA with source port: `>> /cfg/slb/adv/vmasport`	Source IP and source port are used to determine the processor.
VMA with destination IP: `>> /cfg/slb/adv/vmadip`	Source IP and destination IP are used to determine the processor. Both options can be enabled together, where source IP, source port, and destination IP are used to determine the processor.

Note: Radware recommends not changing VMA option while Alteon is in operation, as that may result in temporary disconnection of clients.

The maintenance mode command **`/maint/debug/vmasp`** can be used to find the processor for any combination of source IP, source port (if VMA with source port is enabled), and destination IP (if VMA with destination IP is enabled).

Miscellaneous Debug

When VMA with destination IP is enabled, the following message displays:

```
>> /cfg/slb/adv/vmadip ena
Current VMA with destination IP: disabled
New VMA with destination IP: enabled
WARNING!! Changing VMA option may result in temporary disconnection of clients.
Do you want to continue? [y/n] [n]
```

Client Network Address Translation (Proxy IP)

Network address translation (NAT) is the process of modifying IP address information in IP packet headers while in transit across a traffic-routing device.

There are several types of NAT mechanisms, but the most common method is to hide an entire IP address space behind a single IP address, or a small group of IP addresses. To enable correct handling of returned packets, a many-to-one NAT mechanism must modify higher-level information such as TCP or UDP ports in outgoing communications.

Alteon uses the many-to-one NAT mechanism to translate client IP address and port information. Client NAT can serve several purposes, including:

- Hiding the client IP address from the servers for increased security.
- Solving routing issues when clients and servers belong to the same IP address space (subnet). By using NAT on the the client IP address , traffic returning from the server is forced to pass via Alteon.
- Support for non-transparent proxy functionality. Alteon works as a non-transparent proxy in the following cases:
  — When performing connection management (multiplexing).
  — When performing as an IPv4/IPv6 gateway.

Note: Client IP address translation is mandatory for non-transparent proxy capabilities.

This section includes the following topics:

Client NAT for Virtual Services

You can perform client NAT per virtual service based on one of the following options:

- NAT using a proxy IP address configured on an ingress port or VLAN. For more information, see Port or VLAN-based Proxy IP Addresses, page 191.
- NAT using a proxy IP address configured on an egress port or VLAN. For more information, see Port or VLAN-based Proxy IP Addresses, page 191.
- NAT using a specific proxy IP address or subnet. For more information, see Specific Proxy IP Address for Virtual Service, page 192.
- NAT using a specific network class. For more information, see Specific Proxy IP Address for Virtual Service, page 192.

When client NAT is enabled for a virtual service, you can disable NAT or specify a different proxy IP address for any real server connected to that service. For more information, see Proxy IP Address for Real Servers, page 194.

Additional NAT capabilities on virtual services include:

- Client IP persistency in selecting a proxy IP address—The same proxy IP address is used to redirect all connections from a specific client using the same proxy IP address. Available when a proxy IP subnet or network class is configured per virtual service or real server.
- Host Preservation—Preserves the host bits of an IP address, and translates only the network prefix bits of the IP address. This is useful when the host number is used to identify users uniquely. For more information, see Host Preservation, page 192.

Note: Enable proxy processing on the client ports to perform client NAT on a virtual service.

Port or VLAN-based Proxy IP Addresses

Proxy IP addresses can be associated with physical ports or VLANs. You define the proxy IP mode per virtual service, and determine whether to perform client NAT using the proxy addresses configured on the ingress interface (port or VLAN), or on the egress interface. By default, ingress interface addresses are used.

You must define whether Alteon uses port-based or VLAN-based proxy IP addresses; they cannot both be active on the same Alteon.

When multiple addresses are configured per port or VLAN interface, the proxy IP for each connection is selected in round-robin mode.

Notes

- WAN Link Load Balancing (see WAN Link Load Balancing, page 631) requires port-based proxy IP addresses.
- Use an egress port or a VLAN-based proxy IP address for Web Cache Redirection (WCR) filtering.

You can configure up to 1024 port or VLAN-based proxy IP addresses (IPv4 or IPv6) per Alteon, and up to 32 per single port or VLAN interface.

To configure a virtual service to use ingress port-based proxy IP addresses

1. Configure proxy IP addresses on client ports.

`>> # /cfg/slb/pip/type port`	(Select a port-based PIP address)
`>> # /cfg/slb/pip/add`	(Add an IPv4 PIP address; use **add6** for an IPv6 address)
`Enter Proxy IP address: 10.10.10.1`	
`Enter port <1 to 28> or block <first-last>: e.g. 1 2 3-10`	(Add PIP address to ports 1 to 3)
`New pending: 1: 10.10.10.1 port 1-3`	

2. Enable proxy capability on the client ports.
3. Configure real servers, groups, and a virtual service.

The default value for the virtual service client NAT capability (Proxy IP Mode) is **ingress**, so no special configuration is required on the virtual service in this case. To use egress port-based proxy IPs:

`>> Virtual Server 1 80 http Service # pip/mode`	(Select egress port or VLAN-based proxy IP mode)
`Current pip mode: ingress`	
`Enter new pip mode [disable|ingress|egress|address|nwclss]: egress`	

Specific Proxy IP Address for Virtual Service

You can configure a specific proxy IP address (or entire subnet) per virtual service.

When you configure a specific IP subnet as a proxy IP pool for a virtual service, you can also define whether to select the proxy IP address for each connection in round-robin mode with no persistency, or to ensure client IP persistency (translate all connections from a certain client IP using the same proxy IP).

For a virtual service, you can configure an IPv4 and/or an IPv6 proxy IP address (both could be needed in a mixed IPv4/IPv6 environment).

You can configure up to 1024 IPv4 subnets, and up to 1024 IPv6 addresses per Alteon, as specific proxy IP addresses or as part of proxy IP network class.

Host Preservation

You can choose to translate only the network prefix portion of the client IP address, and to preserve the host portion.

For example, if the proxy IP address is set to 20.12.32.0/255.255.255.0, client IP 133.14.15.29 is translated to 20.12.32.29, client IP 145.11.23.67 is translated to 20.12.32.67, and so on.

This capability requires configuring a proxy IP subnet for the virtual service.

To configure a specific proxy IP address for a virtual service

1. Configure real servers, groups, and a virtual service.
2. Configure a proxy IP address for the virtual service.
   a. Configure a single proxy IP address:

`>> Virtual Server 1 80 http Service # pip/mode`	(Select PIP Mode Address/Subnet)
`Current pip mode: ingress`	
`Enter new pip mode [disable|ingress|egress|address|nwclss]: address`	
`>> Proxy IP# addr`	(Define proxy IP address)
`Current PIP addresses:`	
`v4 none`	
`v6 none`	
`persist disable`	
`Enter new IPv4 PIP address or none []: 2.2.2.35`	
`Enter new IPv4 PIP mask [255.255.255.255]:`	

```
Enter new IPv6 PIP address or none []:
Enter new IPv6 PIP prefix [128]:
```

b. Configure multiple proxy IP addresses (subnet):

`>> Virtual Server 1 80 http Service # pip/mode`	(Select PIP Mode Address/Subnet)
`Current pip mode: ingress`	
`Enter new pip mode [disable|ingress|egress|address|nwclss]: address`	
`>> Proxy IP# addr`	(Define proxy IP subnet)
`Current PIP addresses:`	
`v4 none`	
`v6 none`	
`persist disable`	
`Enter new IPv4 PIP address or none []: 2.2.2.0`	
`Enter new IPv4 PIP mask [255.255.255.255]: 255.255.255.252`	(Available PIP addresses are 2.2.2.1-2.2.2.5)
`Enter new IPv6 PIP address or none []:`	
`Enter new IPv6 PIP prefix [128]:`	
`Enter PIP persistency [disable|client|host][disable]: client`	(Set PIP persistency for the client IP address)

Proxy IP Network Class per Virtual Service

You can use the network class object to configure a pool of proxy IP addresses per virtual service. This is useful when you require a pool of discrete IP addresses or ranges.

For a virtual service, you can configure an IPv4 and/or an IPv6 network class (both could be needed in a mixed IPv4/IPv6 environment).

You can configure up to 1024 IPv4 subnets, and up to 1024 IPv6 addresses per Alteon, as specific proxy IP addresses or as part of proxy IP network class.

To configure a specific proxy IP address for a virtual service

1. Configure real servers, groups, and a virtual service.
2. Configure a network class:

```
>>Layer 4 # nwclss net1
>>Network Class net1 # network
Enter network element id: range1
>> Network Class net1 Network range1 # net
Current network:
Enter network type [range|subnet] [subnet]: range
Enter from IP address []: 2.2.2.10
Enter to IP address []: 2.2.2.20
```

3. Configure a proxy IP address for the virtual service:

```
>> Virtual Server 1 80 http Service # pip/mode
Current pip mode: ingress
Enter new pip mode [disable|ingress|egress|address|nwclss]: nwclss
>> Proxy IP# nwclss
Current PIP network class:
  v4 none
  v6 none
Select new IPv4 PIP network class or none: net1
Select new IPv6 PIP network class or none:
Enter PIP persistency [disable|client][disable]:client
```

Proxy IP Address for Real Servers

For virtual service traffic forwarded to a specific real server, you can choose to disable client IP translation, or to specify a different proxy IP address (address/subnet or network class) to the address configured at virtual service level.

Notes

- Real server proxy IP address configuration is ignored if the client NAT is disabled at the level of the virtual service.
- Real server-level proxy IP address configuration is ignored for traffic that arrives at the real server via a redirect filter. Instead, NAT is performed using proxy IP/NAT addresses defined at filter level.

Client NAT for Filters

Alteon supports translation of client IP addresses for traffic processed by NAT or redirect filters. You can choose to use ingress or egress port or VLAN-based proxy IP addresses, or you can configure a specific proxy IP address for a filter. For more information, see Filtering and Traffic Manipulation.

Using a Virtual Server IP Address to NAT outbound traffic

When internal servers initiate requests to the external network, they require a public IP address for their source IP address. When the real servers initiate traffic flows, Alteon can mask real IP addresses of the servers in the server farm with a virtual server IP address configured in Alteon. Using a virtual server IP address as the PIP address enables conservation of public IP addresses.

This behavior can be achieved by configuring a NAT filter that intercepts outbound traffic initiated by servers, and uses a virtual server IP address as a proxy IP. For more information, see Filtering and Traffic Manipulation.

Mapping Ports

For security, Alteon lets you hide the identity of a port by mapping a virtual server port to a different real server port. This section includes the following topics:

Mapping a Virtual Server Port to a Real Server Port

In addition to providing direct real server access in certain situations (see Mapping Ports for Multiple IP Addresses, page 202), mapping is required when administrators choose to execute their real server processes on different ports than the well-known TCP/UDP ports. Otherwise, virtual server ports are mapped directly to real server ports by default and require no mapping configuration.

Port mapping is configured from the *Virtual Server Services* menu. For example, to map the virtual server TCP/UDP port 80 to real server TCP/UDP port 8004:

```
>> # /cfg/slb/virt 1/service 80                (Select virtual server port 80)
>> Virtual Server 1 http Service# rport 8004   (Map to real port 8004)
```

Mapping a Virtual Server Port to Multiple Real Server Ports

To take advantage of multi-CPU or multi-process servers, Alteon can be configured to map a single virtual port to multiple real ports. This lets site managers, for example, differentiate users of a service by using multiple service ports to process client requests.

Alteon supports up to 64 real ports per server when multiple rports are enabled. This feature allows the network administrator to configure up to 64 real ports for a single service port. It is supported in Layer 4 and Layer 7 and in cookie-based and SSL persistence switching environments.

When multiple real ports on each real server are mapped to a virtual port, Alteon treats the real server IP address/port mapping combination as a distinct real server.

Note: For each real server, you can only configure one service with multiple real ports.

Figure 31 - Basic Virtual Port-to-Real Port Mapping Configuration, page 195 illustrates an example virtual port-to-real port mapping configuration:

Figure 31: Basic Virtual Port-to-Real Port Mapping Configuration

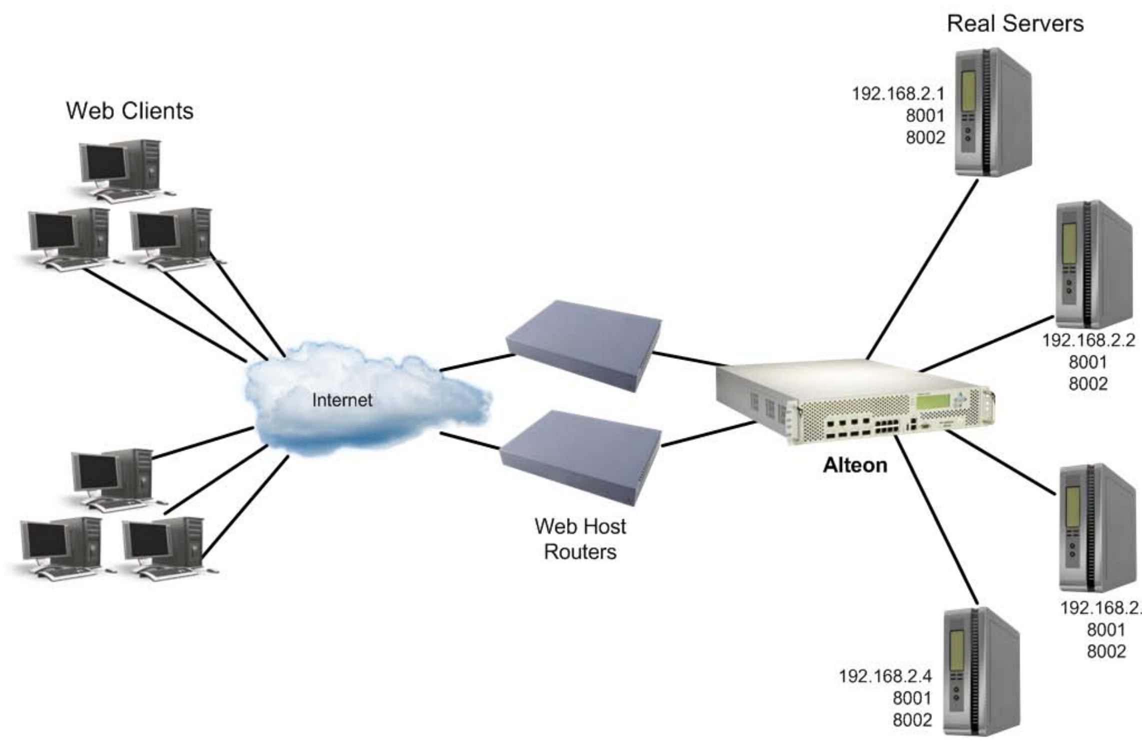


Table 23 - Basic Virtual Port-to-Real Port Mapping Configuration Example, page 196 further illustrates this example:

Table 23: Basic Virtual Port-to-Real Port Mapping Configuration Example

Domain Name	Virtual Server IP Address	Ports Activated	Port Mapping	Real Server IP Address
www.abcxyz.com	192.168.2.100	80 (HTTP)	8001 (rport 1) 8002 (rport 2)	192.168.2.1 (RIP 1) 192.168.2.2 (RIP 2) 192.168.2.3 (RIP 3 192.168.2.4 (RIP 4)

In the example, four real servers are used to support a single service (HTTP). Clients access this service through a virtual server with IP address 192.168.2.100 on virtual port 80. Since each real server uses two ports (8001 and 8002) for HTTP services, the logical real servers are:

- 192.168.2.1/8001
- 192.168.2.1/8002
- 192.168.2.2/8001
- 192.168.2.2/8002
- 192.168.2.3/8001
- 192.168.2.3/8002
- 192.168.2.4/8001
- 192.168.2.4/8002

Load-Balancing Metric

For each service, a real server is selected using the configured load-balancing metric (hash, leastconns, minmisses, or roundrobin). To ensure even distribution, once an available server is selected, Alteon uses the roundrobin metric to choose a real port to receive the incoming connection.

If the algorithm is leastconns, Alteon sends the incoming connections to the logical real server (real server IP address/port combination) with the least number of connections.

The **`/cfg/slb/virt`** command defines the real server TCP or UDP port assigned to a service. By default, this is the same as the virtual port (service virtual port). If rport is configured to be different from the virtual port defined in **`/cfg/slb/virt <virtual server number> / service <virtual port>`**, Alteon maps the virtual port to the real port.

Note: To use the single virtual port to multiple rports feature, set this real server port option to 0. However, you *cannot* configure multiple services with multiple rports in the same server if the multiple rport feature is enabled.

Configuring Multiple Service Ports

To configure multiple serve ports

Two commands, addport and remport, under the *Real Server* menu, let users add or remove multiple service ports associated with a particular server. A service port is a TCP or UDP port number. For example: **`addport 8001`** and **`remport 8001`**.

1. Configure the real servers.

```
>> # /cfg/slb/real 1/rip 192.168.2.1/ena
>> # /cfg/slb/real 2/rip 192.168.2.2/ena
>> # /cfg/slb/real 3/rip 192.168.2.3/ena
>> # /cfg/slb/real 4/rip 192.168.2.4/ena
```

2. Add all four servers to a group.

```
>> # /cfg/slb/group 1
>> Real server Group 1# add 1
>> Real server Group 1# add 2
>> Real server Group 1# add 3
>> Real server Group 1# add 4
```

3. Configure a virtual server IP address.

```
>> # /cfg/slb/virt 1/vip 192.168.2.100/ena
```

4. Turn on multiple rport for port 80.

```
>> # /cfg/slb/virt 1/service 80/rport 0
```

5. Add the ports to which the Web server listens.

```
>> # /cfg/slb/real 1/addport 8001          (Add port 8001 to Real Server 1)
>> # addport 8002                          (Add port 8002 to Real Server 1)
>> # /cfg/slb/real 2/addport 8001          (Add port 8001 to Real Server 2)
>> # addport 8002                          (Add port 8002 to Real Server 2)
>> # /cfg/slb/real 3/addport 8001          (Add port 8001 to Real Server 3)
>> # addport 8002                          (Add port 8002 to Real Server 3)
>> # /cfg/slb/real 4/addport 8001          (Add port 8001 to Real Server 4)
>> # addport 8002                          (Add port 8002 to Real Server 4)
```

Direct Server Return

The Direct Server Return (DSR) feature allows the server to respond directly to the client. This is useful for sites where large amounts of data flow from servers to clients, such as with content providers or portal sites that typically have asymmetric traffic patterns.

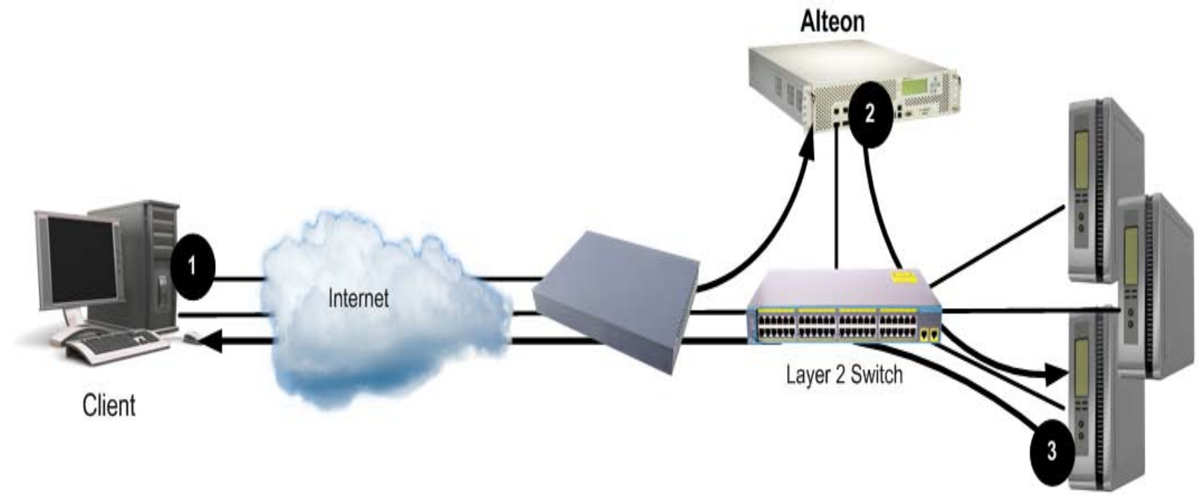
DSR and content-intelligent Layer 7 load balancing cannot be performed at the same time because content-intelligent load balancing requires that all frames go back to Alteon for connection splicing.

DSR requires that the server be set up to receive frames that have a destination IP address that is equal to the virtual server IP address.

How Direct Server Return Works

The sequence of steps that are executed in DSR are illustrated in Figure 32 - Direct Server Return, page 198:

Figure 32: Direct Server Return



1. A client request is forwarded to Alteon.
2. Because only MAC addresses are substituted, Alteon forwards the request to the best server, based on the configured load-balancing policy.
3. The server responds directly to the client, bypassing Alteon, and using the virtual server IP address as the source IP address.

To set up DSR

```
>> # /cfg/slb/real <real server number>/adv/submac ena
>> # /cfg/slb/virt <virtual server number>/service <service number>/nonat ena
```

One Arm Topology Application

Source MAC Address Substitution

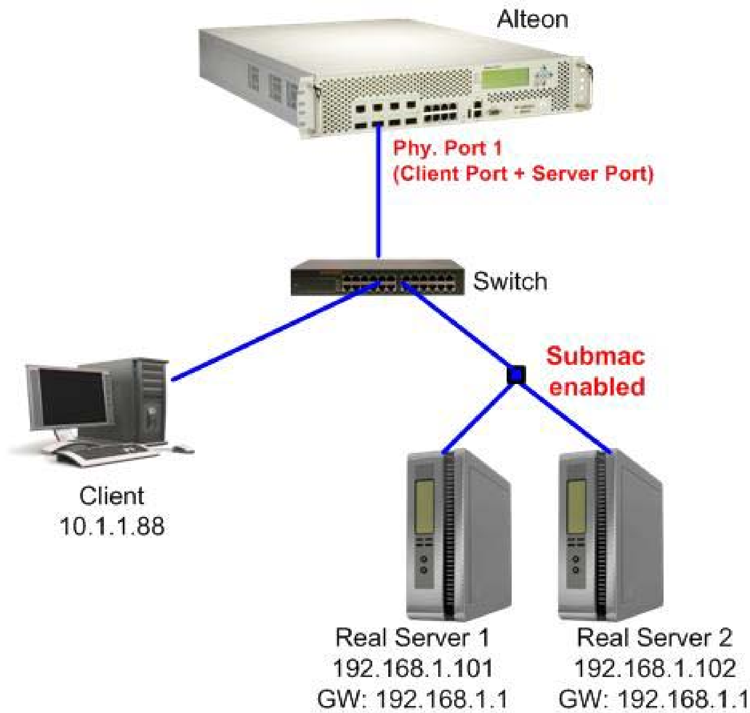
By default, in packets destined for servers in an SLB environment, the source MAC address is not modified and the client request is forwarded to the server with the MAC address of the client. You can substitute the client source MAC address for the packets going to the server with the Alteon MAC address using source MAC address substitution.

You can enable this feature globally (using the **`/cfg/slb/adv/submac enable`** command), or per-real service (using the **`/cfg/slb/real/adv/submac enable`** command). Global MAC address substitution supersedes per-real service MAC address substitution.

One Arm SLB Configuration

In a one-arm SLB configuration, you must enable MAC address substitution to avoid session failure. As illustrated in Figure 33 - One Arm Topology, page 199, in a one-arm configuration, the client and server are on same broadcast domain but have different IP address ranges.

Figure 33: One Arm Topology



Because in this configuration delayed binding (dbind) is enabled, you must force the reply traffic from the server to go back through Alteon for correct session conversion. This is performed through routing and not proxy IP (PIP), which forces the traffic to return though Alteon without making changes on the server.

In this configuration, everything works properly on the server side. The server receives packets with the client's source MAC address, and because it has a different IP range than the client, the server correctly returns the traffic to the client. However, the packets fail to reach the client because both Alteon and the Layer 2 switch are located on the same broadcast domain. This results in Alteon forwarding packets from the client on a different port on the Layer 2 switch, with the MAC address acting like a floating address, meaning that first the Layer 2 switch reads the client MAC address on the client's physical port, and then it reads it on the Alteon physical port.

When enabling source MAC substitution, the packets sent from an Alteon only use Alteon's MAC address, so the client MAC address "remains" on the client port of the switch.

Example - Enabling Source MAC Substitution for a One-Arm Topology

To set up submac for one arm topology

```
/cfg/l2/stg 1/off
/cfg/l3/if 1/addr 10.1.1.1/mask 255.255.255.0/en
/cfg/l3/if 2/addr 192.168.1.1/mask 255.255.255.0/en
/cfg/slb/on
/cfg/slb/adv/direct en
/cfg/slb/real 1/rip 192.168.1.101/en/submac en
/cfg/slb/real 2/rip 192.168.1.102/en/submac en
/cfg/slb/group 1/add 1/add 2
/cfg/slb/virt 1/vip 10.1.1.100/en
/cfg/slb/virt 1/service 80/group 1
/cfg/slb/port 1/client en
/cfg/slb/port 1/server en
```

Direct Access Mode

Direct Access Mode (DAM) allows any client to communicate with any real server's load-balanced service. Also, with DAM enabled, any number of virtual services can be configured to load balance a real service.

DAM enables both client and server processing on the same port to handle traffic that requires direct access to real servers.

The following topics are discussed in this section:

- Configuring Global Direct Access Mode, page 200
- Blocking Direct Access Mode on Selected Services, page 201
- Direct Access Mode Limitations, page 201

Configuring Global Direct Access Mode

To configure Direct Access Mode globally on Alteon

```
>> Main# /cfg/slb/adv/direct e
Current Direct Access Mode: disabled
New Direct Access Mode: enabled
```

When DAM (**/cfg/slb/adv/direct**) is enabled, any client can communicate with any real server's load-balanced service. Also, any number of virtual services can be configured to load balance a real service.

With DAM, traffic that is sent directly to real server IP addresses (instead of the virtual server IP address) is excluded from load-balancing decisions. The same clients may also communicate to the virtual server IP address for load-balanced requests.

DAM is necessary for applications such as:

- Direct access to real servers for management or administration.
- One real server serving multiple virtual server IP (VIP) addresses.

- Content-intelligent load balancing, which requires traffic to go to specific real servers based on the inspection of HTTP headers, content identifiers such as URLs and cookies, and the parsing of content requests.

  For more information see Content-Intelligent Server Load Balancing, page 219.

When DAM is enabled, port mapping and default gateway load balancing is supported only when filtering is enabled, a proxy IP address is configured, or URL parsing is enabled on any port.

Blocking Direct Access Mode on Selected Services

When Direct Access Mode (DAM) is enabled globally on Alteon, it can also be disabled on selected virtual servers and virtual services.

Example

You have enabled direct access mode on Alteon so that it can support content-intelligent load balancing applications such as those described in Content-Intelligent Server Load Balancing, page 219.

However, you also want to load balance a stateless protocol such as UDP, which by its nature cannot be recorded in a session entry in the session table.

To block use of DAM for the UDP protocol (service port 9200)

```
>> Main# /cfg/slb/adv/direct e                    (Enable DAM globally on Alteon)
>> /cfg/slb/virt 1/service 9200/direct disable
```

Notes

- The **/cfg/slb/virt <x> /service <y> /direct** command requires that DAM be enabled globally on Alteon. If DAM is not enabled globally on Alteon, the **direct disable** command has no effect. When DAM is enabled on Alteon and disabled on a virtual server/virtual port pair, direct access to other real servers (those servers that are not servicing a virtual server/virtual port pair with direct access mode disabled) is still allowed.
- DAM cannot be disabled for FTP and RTSP services.

Direct Access Mode Limitations

In the default SLB configuration with client and server on separate VLANs and Direct Access Mode disabled, a server response fails to reach the client if the server sends the response to the MAC address present in the client request.

The response does reach the client if one of the following conditions are met:

- The server sends the response to the MAC address of the default gateway (the Alteon interface or the virtual interface router).
- Direct Access Mode is enabled at **/cfg/slb/adv/direct**.
- MAC address substitution is enabled at **/cfg/slb/adv/submac**.

Assigning Multiple IP Addresses

One way to provide both SLB access and direct access to a real server is to assign multiple IP addresses to the real server. For example, one IP address could be established exclusively for SLB and another could be used for direct access needs.

Using Proxy IP Addresses

Proxy IP (PIP) addresses are used primarily to eliminate SLB topology restrictions in complex networks (see Client Network Address Translation (Proxy IP), page 190). PIP addresses can also provide direct access to real servers.

If Alteon is configured with proxy IP addresses and the client port is enabled for proxy, the client can access each real server directly using the real server's IP address. To directly access a real server, the port connected to the real server must have server processing disabled. However, if DAM is enabled (**`/cfg/slb/adv/direct ena`**), server processing must be enabled on the server port regardless of the proxy setting and SLB is still accessed using the virtual server IP address.

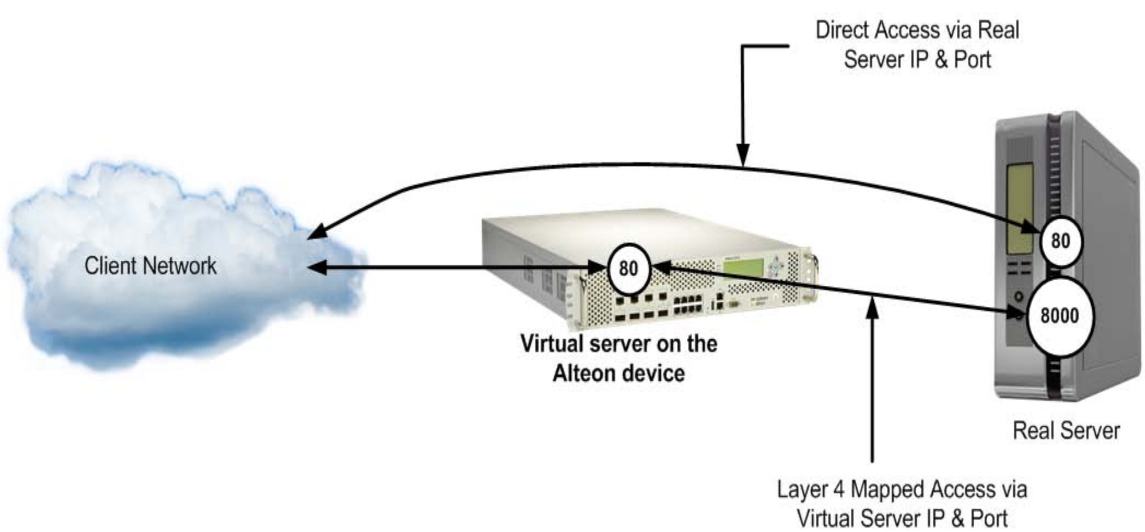
Mapping Ports for Multiple IP Addresses

When SLB is used without PIP addresses and without DAM, Alteon must process the server-to-client responses. If a client were to access the real server IP address and port directly, bypassing client processing, the server-to-client response could be mishandled by SLB processing as it returns through Alteon, with the real server IP address getting remapped back to the virtual server IP address on Alteon.

First, two port processes must be executed on the real server. One real server port handles the direct traffic, and the other handles SLB traffic. Then, the virtual server port on Alteon must be mapped to the proper real server port.

Figure 34 - Mapped and Non-Mapped Server Access, page 202 illustrates a topology where clients can access SLB services through well-known TCP port 80 at the virtual server's IP address. Alteon behaves like a virtual server that is mapped to TCP port 8000 on the real server. For direct access that bypasses the virtual server and SLB, clients can specify well-known TCP port 80 as the real server's IP address.

Figure 34: Mapped and Non-Mapped Server Access



Port mapping is supported with DAM when filtering is enabled, a proxy IP address is configured, or URL parsing is enabled on any port.

For more information on how to map a virtual server port to a real server port, see Mapping Ports, page 194.

Monitoring Real Servers

Typically, the management network is used by network administrators to monitor real servers and services. By configuring the mnet and mmask options of the *SLB Configuration* menu (**/cfg/slb/adv**), you can access the real services being load balanced.

Note: Clients on the management network do not have access to SLB services and cannot access the virtual services being load balanced.

The mnet and mmask options are described below:

- **mnet**—If defined, management traffic with this source IP address is allowed direct (non-SLB) access to the real servers. Only specify an IP address in dotted decimal notation. A range of IP addresses is produced when used with the mmask option.
- **mmask**—This IP address mask is used with mnet to select management traffic that is allowed direct real server access only.

Delayed Binding

Delayed binding can be used in several scenarios, for example Layer 7 matching, for which you need to accumulate information about the client connection on which a load-balancing decision is performed.

Delayed binding consists of the following statuses:

- Enabled— Performs SYN SYN denial-of-service Protection and enables some Alteon Layer 7 capabilities and SYN protection.
- Disabled— No delayed binding is performed.
- Force Proxy—Uses the Application Service Engine and enables TCP Optimization.

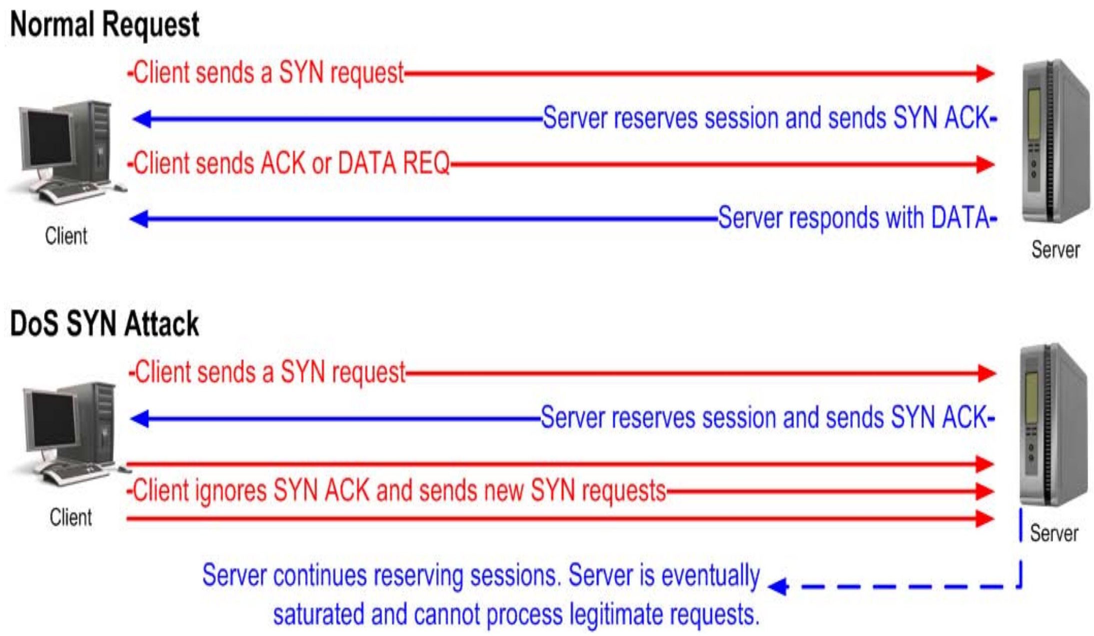
Delayed Binding Using Denial-of-service Protection

The delayed binding feature prevents SYN denial-of-service (DoS) attacks on the server. DoS occurs when the server or Alteon is denied servicing the client because it is saturated with invalid traffic.

Typically, a three-way handshake occurs before a client connects to a server. The client sends out a synchronization (SYN) request to the server. The server allocates an area to process the client requests, and acknowledges the client by sending a SYN ACK. The client then acknowledges the SYN ACK by sending an acknowledgement (ACK) back to the server, thus completing the three-way handshake.

Figure 35 - Mapped and Non-Mapped Server Access, page 204 illustrates a classic type of SYN DoS attack. If the client does not acknowledge the server's SYN ACK with a data request (REQ) and instead sends another SYN request, the server gets saturated with SYN requests. As a result, all of the servers resources are consumed and it can no longer service legitimate client requests.

Figure 35: Mapped and Non-Mapped Server Access

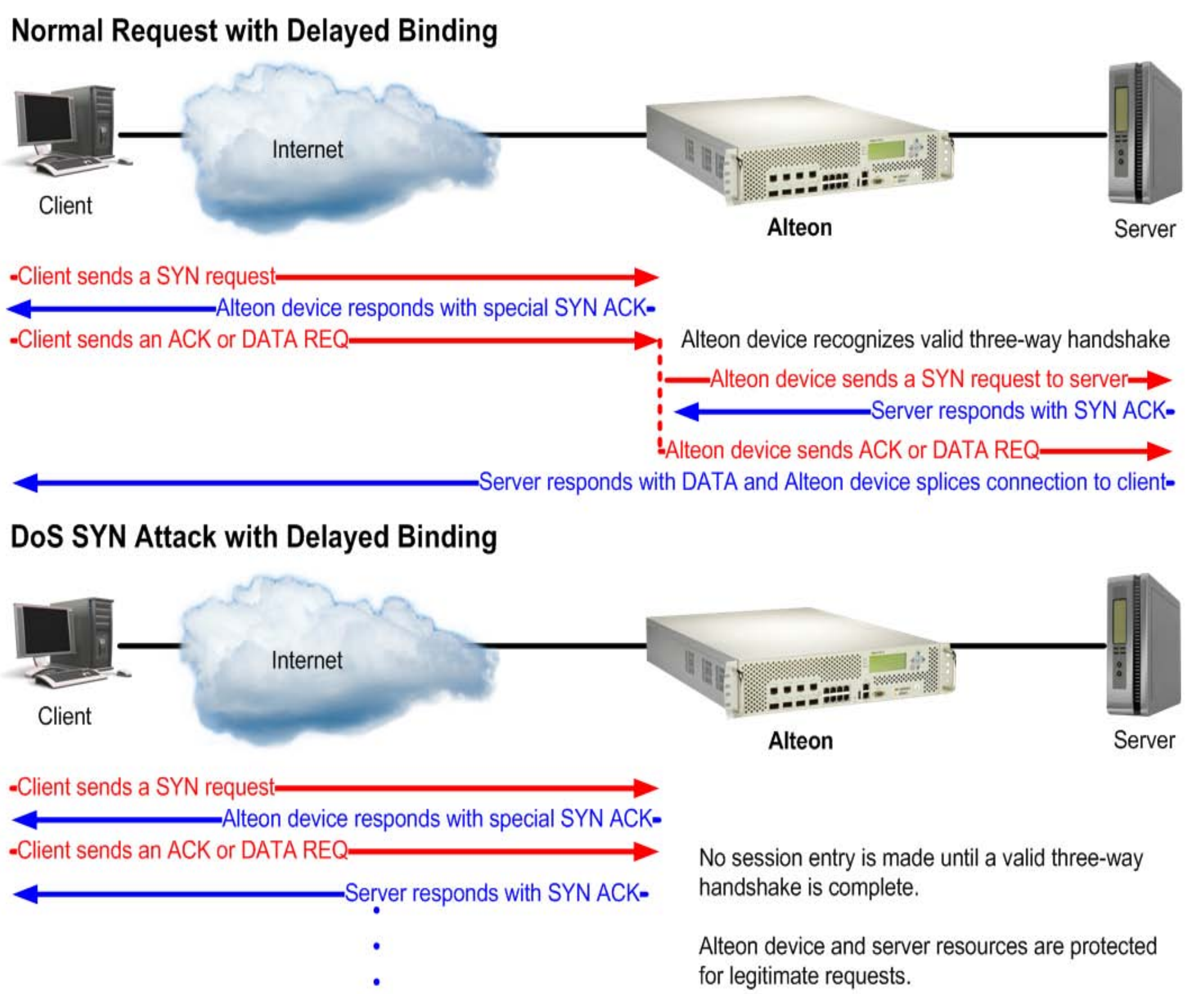


Using delayed binding, Alteon intercepts the client SYN request before it reaches the server. Alteon responds to the client with a SYN ACK that contains embedded client information. Alteon does not allocate a session until a valid SYN ACK is received from the client or the three-way handshake is complete.

Repelling DoS SYN Attacks With Delayed Binding

Figure 36 - Normal Request with Delayed Binding, page 205 is an illustration of a normal request with delated binding.

Figure 36: Normal Request with Delayed Binding



After Alteon receives a valid ACK or DATA REQ from the client, Alteon sends a SYN request to the server on behalf of the client, waits for the server to respond with a SYN ACK, and then forwards the clients DATA REQ to the server. This means that Alteon delays binding the client session to the server until the proper handshakes are complete.

As a result, two independent TCP connections span a session: one from the client to Alteon, and the second from Alteon to the selected server. Alteon temporarily terminates each TCP connection until content has been received, preventing the server from being inundated with SYN requests.

Note: Delayed binding is enabled when content-intelligent load balancing is used. However, if you are not parsing content, you must explicitly enable delayed binding if desired.

Configuring Delayed Binding

To configure delayed binding

```
>> # /cfg/slb/virt <virtual server number> /service <service type> /dbind
Current delayed binding: disabled
Enter new delayed binding [d/e/f]:e
```

Note: Enable delayed binding without configuring any HTTP SLB processing or persistent binding types.

To configure delayed binding for cache redirection, see Delayed Binding for Cache Redirection, page 461.

Detecting SYN Attacks

In Alteon, SYN attack detection is enabled by default whenever delayed binding is enabled. SYN attack detection includes the following capabilities:

- Provides a way to track half open connections
- Activates a trap notifying that the configured threshold has been exceeded
- Monitors DoS attacks and proactively signals alarm
- Provides enhanced security
- Improves visibility and protection for DoS attacks

The probability of a SYN attack is higher if excessive half-open sessions are generated on Alteon. Half-open sessions show an incomplete three-way handshake between the server and the client. You can view the total number of half-open sessions from the **`/stat/slb/layer7/maint`** menu.

To detect SYN attacks, Alteon keeps track of the number of new half-open sessions for a set period. If the value exceeds the threshold, then a syslog message and an SNMP trap are generated.

You can change the default parameters for detecting SYN attacks in the **`/cfg/slb/adv/synatk`** menu. You can specify how frequently you want to check for SYN attacks, from two seconds to one minute, and modify the default threshold representing the number of new half-open sessions per second.

Force Proxy Using the Application Service Engine

Alteon provides various application layer services which require a full TCP proxy behavior. Some of these capabilities include SSL offloading, HTTP caching and compression, HTTP modifications, TCP optimizations, and more. To facilitate these functionalities, Alteon includes a module named Application Service Engine.

The Application Service Engine is a full TCP proxy which performs delayed binding of connections, during which it can optimize TCP behavior, intercept client requests and server responses to modify them, and so on. In some cases, the proxy behavior itself may be required even without the use of any other application service. For this purpose, you can set delayed binding to *Force Proxy* mode. In this mode, the Application Service Engine will perform TCP optimizations without SYN attack protection (which is performed when the delayed binding mode is set to enabled), function as a full TCP proxy, reorder TCP packets, and so on.

The Application Service Engine can work in both Alteon delayed binding modes. In enabled delayed binding mode, the Application Service Engine only provides SYN attack protection. In force proxy mode, it only provides TCP optimizations.

Configuring Force Proxy

To configure force proxy

```
>> # /cfg/slb/virt <virtual server number> /service <service type> /dbind
Current delayed binding: disabled
Enter new delayed binding [d/e/f]:f
```

IP Address Ranges Using imask

The imask option lets you define a range of IP addresses for the real and virtual servers configured under SLB. By default, the imask setting is 255.255.255.255, which means that each real and virtual server represents a single IP address. An imask setting of 255.255.255.0 means that each real and virtual server represents 256 IP addresses.

Consider the following example:

- A virtual server is configured with an IP address of 172.16.10.1.
- Real servers 172.16.20.1 and 172.16.30.1 are assigned to service the virtual server.
- The imask is set to 255.255.255.0.

If the client request is set to virtual server IP address 172.16.10.45, the unmasked portion of the address (0.0.0.45) gets mapped directly to whichever real server IP address is selected by the SLB algorithm. This results in the request being sent to either 172.16.20.45 or 172.16.30.45.

Session Timeout Per Service

This feature allows for the configuration of session timeout based on a service timeout instead of the real server timeout. With this feature, by default the timeout value for the service is set to 0. When the value is 0, the service uses the real server timeout value. Once the timeout value for the service is configured, the new configuration is used instead.

The timeout for aging of persistent sessions is prioritized. According to the priority, persistent timeout is the highest followed by virtual service and real server timeout.

Note: Persistent timeout must be greater than the virtual service and real server timeout.

This is useful when sessions need to be kept alive after their real server configured timeout expires. An FTP session could be kept alive after its server defined timeout period, for example.

Example Configure a timeout of 10 minutes for HTTP (service 80) on virtual server 1

1. Select service 80.

```
>> Main# /cfg/slb/virt 1/service 80
```

2. Set the service timeout value.

```
>> Virtual Server 1 http Service#  tmout 10
```

3. Save configuration.

```
>> Virtual Server 1 http Service# apply
>> Virtual Server 1 http Service# save
```

IPv6 and Server Load Balancing

Alteon provides a full range of SLB options for Internet Protocol version 6 (IPv6).

Pure IPv6 Environment

In this environment, IPv6 virtual address traffic is sent to IPv6 real servers, where Alteon supports

- Layer 4 and Layer 7 traffic processing for HTTP and HTTPS, including application acceleration, and Layer 7 traffic processing for DNS over UDP.
- Layer 4 SLB for all other applications.

Mixed IPv4 and IPv6 Environment (Gateway)

In this environment, IPv6 client traffic is sent to IPv4 real servers, or IPv4 client traffic is sent to IPv6 real servers. Real server groups can contain mixed IPv4 and IPv6 servers.

When the IP version of the server is different from the IP version of the client, Alteon converts the client packet to a packet of the server IP version before it is forwarded to the server. In this environment, Alteon supports

- Layer 4 and Layer 7 traffic processing for HTTP and HTTPS, including application acceleration.
- Layer 4 SLB and SSL offloading for SSL.
- Basic Layer 4 SLB for UDP and TCP.

Note: Since IPv6 does not allow intermediary routers or switches to fragment packets, internal translation of the maximum IPv4 packet (MTU of 1500) cannot be translated without fragmenting. Therefore, all IPv4 real servers must use IPv6 SLB to be configured with a maximum MTU less than or equal to 1480.

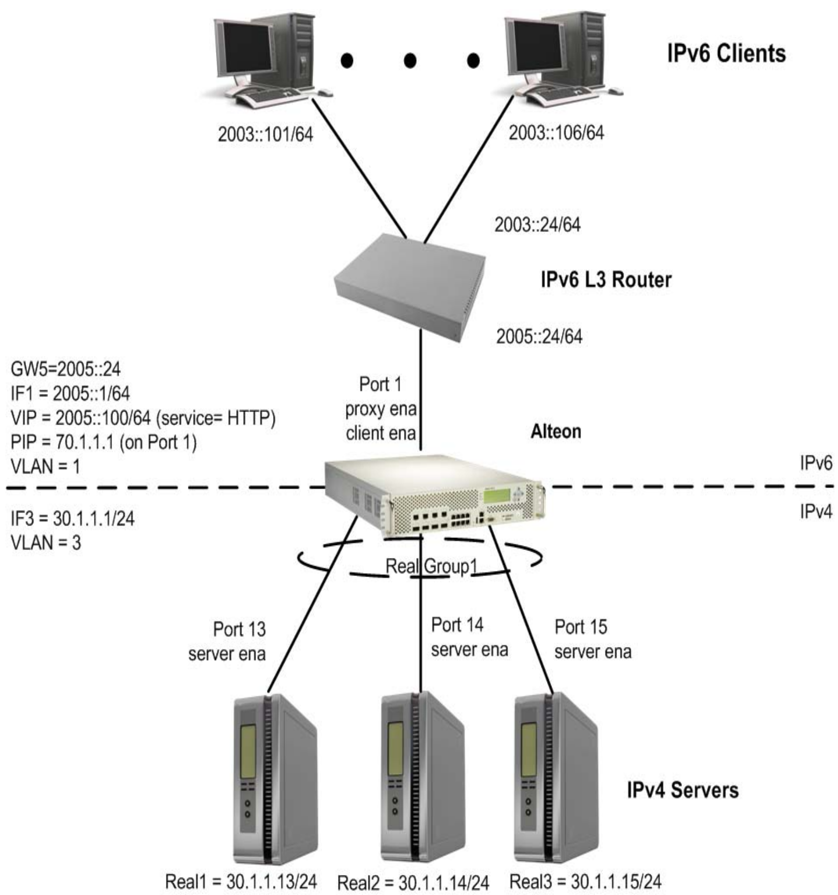
For example, in the Windows 2003 environment, run **REGEDIT** to add a new parameter to the registry in **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces*xx*** (where *xx* is the correct interface for the configured IP address), with the keyword **MTU**, using **REG_DWORD** with a decimal value of **1480**.

PIP addresses can be in either IPv4 or IPv6 format. Ports and VLANs can be assigned either one type or both. The appropriate PIP is used in load-balancing operations based on the IP version of the incoming packet.

IPv6 to IPv4 Server Load Balancing

Figure 37 - IPv6 to IPv4 Layer 4 SLB Example, page 209 illustrates SLB between IPv6 clients and IPv4 servers:

Figure 37: IPv6 to IPv4 Layer 4 SLB Example



To configure IPv6 support for load balancing IPv4 real servers

This procedure references Figure 37 - IPv6 to IPv4 Layer 4 SLB Example, page 209.

1. Configure the IPv6 network interface.

```
>> Main# /cfg/l3/if 1
>> IP Interface 1# ena
>> IP Interface 1# ipver v6
>> IP Interface 1# addr 2005:0:0:0:0:0:0:1
>> IP Interface 1# mask 64
>> IP Interface 1# apply
```

2. Configure VLAN for Interface 3.

```
>> Main# /cfg/l2/vlan 3
>> VLAN 3# ena
>> VLAN 3# add 13
Port 13 is an UNTAGGED port and its current PVID is 1.
Confirm changing PVID from 1 to 3 [y/n]: y
>> VLAN 3# add 14Port 14 is an UNTAGGED port and its current PVID is 1.
Confirm changing PVID from 1 to 3 [y/n]: y
>> VLAN 3# add 15Port 15 is an UNTAGGED port and its current PVID is 1.
Confirm changing PVID from 1 to 3 [y/n]: y
```

3. Configure the IPv4 network interface for the real servers.

```
>> Main# /cfg/l3/if 3
>> Interface 3# ena
>> Interface 3# ipver v4
>> Interface 3# addr 30.1.1.1
>> Interface 3# mask 255.255.255.0
>> Interface 3# broad 30.1.1.255
>> Interface 3# vlan 3
```

4. Configure the IPv6 default gateway.

```
>> Main# /cfg/l3/gw 5
>> Default gateway 5# ena
>> Default gateway 5# ipver v6
>> Default gateway 5# addr 2005:0:0:0:0:0:0:24
>> Default gateway 5# vlan 1
```

5. Configure the IPv6 virtual server IP address.

```
>> Main# /cfg/slb/virt 1
>> Virtual Server 1# ena
>> Virtual Server 1# ipver v6
>> Virtual Server 1# vip 2005:0:0:0:0:0:0:100
```

6. Assign the HTTP service to virtual server.

```
>> Main# /cfg/slb/virt 1/service http
>> Virtual Server 1 http Service# group 1
```

7. Enable SLB.

```
>> Main# /cfg/slb/on
```

8. Configure real servers and a real server group.

```
>> Main# /cfg/slb/real 1
>> Real Server 1# ena
>> Real Server 1# rip 30.1.1.13
>> Main# /cfg/slb/real 2
>> Real Server 2# ena
>> Real Server 2# rip 30.1.1.14
>> Main# /cfg/slb/real 3
>> Real Server 3# ena
>> Real Server 3# rip 30.1.1.15
>> Main# /cfg/slb/group 1
>> Real Server Group 1# ena
>> Real Server Group 1# health http
>> Real Server Group 1# add 1
>> Real Server Group 1# add 2
>> Real Server Group 1# add 3
```

9. Configure client and server processing on the client and server ports.

```
>> Main# /cfg/slb/port 1
>> SLB Port 1# client ena
>> Main# /cfg/slb/port 13
>> SLB Port 13# server ena
>> Main# /cfg/slb/port 14
>> SLB Port 14# server ena
>> Main# /cfg/slb/port 15
>> SLB Port 15# server ena
```

10. Configure a PIP and enable it on the client port.

    The PIP address is used to converge the IPv4 and IPv6 traffic. Optionally, the PIP address can be assigned to a VLAN instead of the port. To enable it on the VLAN, use the command `/cfg/slb/pip/type vlan`, instead of `/cfg/slb/pip/type port`.

```
>> Main# /cfg/slb/pip/type port
>> Proxy IP Address# add 70.1.1.1 1
>> Main# /cfg/slb/port 1
>> SLB Port 1# proxy ena
```

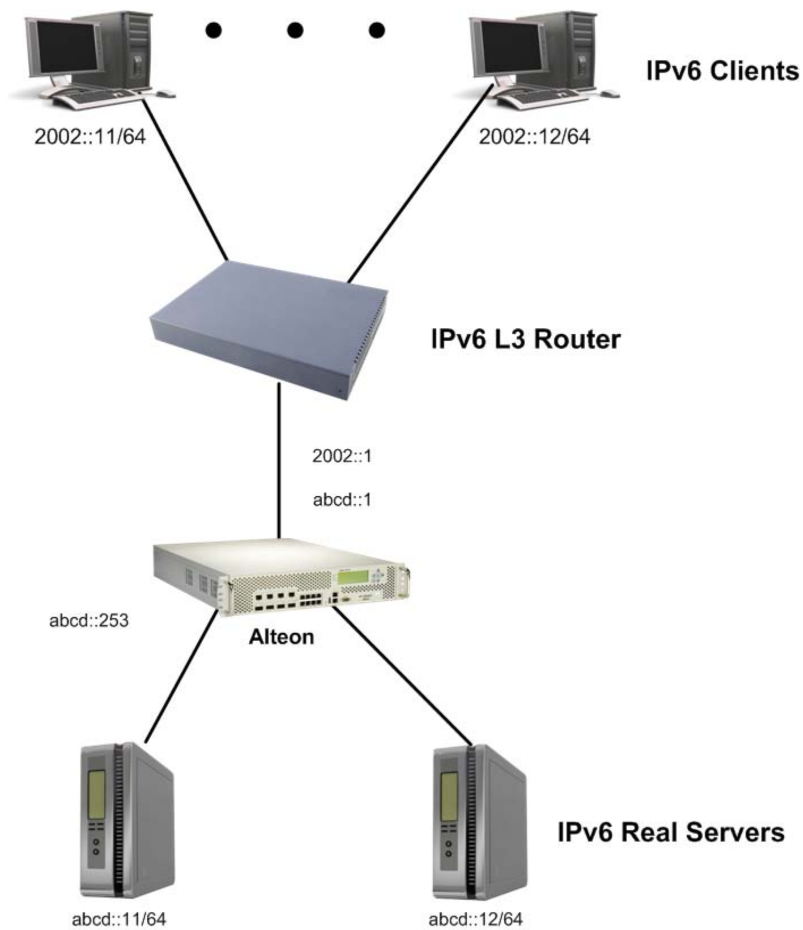
11. Apply and save the configuration.

```
>> Management Port# apply
>> Management Port# save
```

IPv6 to IPv6 Server Load Balancing

Figure 38 - IPv6 to IPv6 Layer 4 SLB Example, page 212 illustrates SLB between IPv6 clients and IPv6 servers:

Figure 38: IPv6 to IPv6 Layer 4 SLB Example



To configure IPv6 support for load balancing IPv6 real servers

This procedure references Figure 38 - IPv6 to IPv6 Layer 4 SLB Example, page 212.

1. Configure the IPv6 network interface.

```
>> Main# /cfg/l3/if 1>
> Interface 1# ena
>> Interface 1# ipver v6
>> Interface 1# addr abcd:0:0:0:0:0:0:253
>> Interface 1# mask 64
```

2. Globally enable load balancing.

```
>> Main# /cfg/slb
>> Layer 4# on
```

3. Configure the IPv6 real servers.

```
>> Main# /cfg/slb/real 1
>> Real Server 1# ena
>> Real Server 1# ipver v6
>> Real Server 1# rip abcd:0:0:0:0:0:0:11
>> Main# /cfg/slb/real 2
>> Real Server 2# ena
>> Real Server 2# ipver v6
>> Real Server 2# rip abcd:0:0:0:0:0:0:12
```

4. Configure the IPv6 real server groups.

```
>> Main# /cfg/slb/group 1
>> Real Server Group 1# ipver v6
>> Real Server Group 1# add 1
>> Real Server Group 1# add 2
```

5. Enable client processing on the SLB ports.

```
>> Main# /cfg/slb/port 1
>> SLB Port 1# client ena
>> Main# /cfg/slb/port 2
>> SLB Port 2# client ena
```

6. Enable server processing on the SLB ports.

```
>> Main# /cfg/slb/port 21
>> SLB Port 21# server ena
>> Main# /cfg/slb/port 22
>> SLB Port 22# server ena
```

7. Create the IPv6 virtual servers.

```
>> Main# /cfg/slb/virt 1
>> Virtual Server 1# ena
>> Virtual Server 1# ipver v6
>> Virtual Server 1# vip abcd:0:0:0:0:0:0:100
```

8. Assign the desired service to the IPv6 virtual server group.

```
>> Main# /cfg/slb/virt 1/service http
>> Virtual Server 1 http Service# group 1
```

IPv6 Layer 4 SLB Information

The following commands are used to display IPv6 Layer 4 session information:

To display IPv6-related items in the SLB session dump

```
>> Main# /info/slb/sess/dump
```

To display IPv6 client IP addresses in the SLB session dump

```
>> Main# /info/slb/sess/cip6
```

To display IPv6 destination IP addresses in the SLB session dump

```
>> Main# /info/slb/sess/dip6
```

IPv6 Real Server Health Checks

Health checking is supported for IPv6 real servers. For information on the configuration and management of health checking, refer to Health Checking, page 481.

Source Network-Based Server Load Balancing

Alteon lets you provide differentiated services for specific client groups, including different types of services, different levels of service, and different service access rights. This can be achieved by adding source IP classification to a virtual server or filter using network classes.

A network class is a configuration object that can include multiple IP ranges and/or IP subnets and can be used for traffic classification.

- Configuring Network Classes, page 214
- Configuring Source Network-Based Server Load Balancing, page 216

Configuring Network Classes

A network class contains multiple network elements, with each element defining a specific range, a specific IP subnet, or a specific IP address that is either included in the network class or excluded from the network class. Using network classes for traffic classification, you can add or remove IP addresses without changing the entire traffic classification configuration.

You can configure up to 1024 network classes, 4096 subnets or IP ranges, and 8192 single IPs.

To configure a network class

1. Access the *Network Class* menu.

```
>> # /cfg/slb/nwclss
```

2. At the prompt, enter the network class ID you want to configure. The *Network Class* menu displays.

```
[Network Class NWC1 Menu]
     name      - Set network class name
     network   - Network Element Menu
     ipver     - Set IP version
     copy      - Copy network class
     del       - Delete network class
     cur       - Display current network class
```

3. To define the network class name for that ID, enter **name**. At the prompt, enter the name you want to define.
4. To set a network element for the network class, enter **network.** At the prompt, enter the network element ID you want to set. The *Network Element* menu displays.

```
[Network Class NWC1 Network 123 Menu]
     net       - Set network element
     del       - Delete network element
     cur       - Display current network element
```

5. Enter **net** to define the network element. At the prompt, do one of the following:
   — Enter **range** to define a range of IP addresses, and the network match type.

```
Enter network type [range|subnet] [subnet]: range
Enter from IP address []:
Enter to IP address []:
Enter network match type [exclude|include] [include]:
```

   — Enter **subnet** to define an IP address, a subnet mask, and the network match type.

```
Enter network type [range|subnet] [range]: subnet
Enter IP address []:
Enter subnet mask []:
Enter network match type [exclude|include] [include]:
```

For a description of all of the `/cfg/slb/nwclss` commands, refer to the *Alteon Application Switch Operating System Command Reference*.

Configuring Source Network-Based Server Load Balancing

To configure differentiated service for a specific source network, you can configure network classes that define the required source network for specific virtual servers.

The configuration described in this example procedure is defined with the following service differentiation requirements:

- Accelerate applications for external service users. Caching and compression are applied to external client traffic.
- Regular application delivery for internal service customers.

To configure source network-based SLB

1. Before you can configure SLB string-based load balancing, ensure that Alteon is configured for basic SLB with the following tasks:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface.
   - Define each real server.
   - Assign servers to real server group 1.
   - Define caching policy **cache_ext**.
   - Define compression policy **compress_ext**.
   - Enable SLB
   - Enable client processing on the port connected to the clients.

   For information on how to configure your network for SLB, see Server Load Balancing, page 165.
2. Define network classes for the type of differentiated services you want to configure.

`>> # /cfg/slb/nwclss internal`	(Create a network class called **internal**.)
`>> Network Classifier internal# network 1/net range 10.201.1.1 10.205.255.255 include`	(Define network element 1 for this network class to include an IP address range.)
`>> # /cfg/slb/nwclss external`	(Create a network class called **external**.)
`>> Network Classifier external# network 1/net range 10.201.1.1 10.205.255.255 exclude`	(Specify a network element 1 for this network class to exclude an IP address range.)

3. Define virtual servers for internal and external customers, and assign the network classes you defined for each virtual server accordingly. Define an HTTP service for each of the virtual servers.

`>> # /cfg/slb/virt 1/vip 128.100.100.100`	(Define VIP for Virtual Server 1)
`>> Virtual 1 # srcnet internal`	(Assign the network class **internal** to Virtual Server 1)
`>> Virtual Server 1# service HTTP`	(Define the HTTP service for Virtual Server 1)
`>> Virtual Server 1 80 http Service# group 1`	(Set the group to Group 1)
`>> # /cfg/slb/virt 2/vip 128.100.100.100`	(Define the same VIP for Virtual Server 2)

`>> Virtual 2 # /cfg/slb/virt 2/srcnet external`	(Assign the network class **external** to Virtual Server 2)
`>> Virtual Server 2# service HTTP`	(Define the HTTP service for Virtual Server 2)
`>> Virtual Server 2 80 http Service# group 1`	(Set the group to Group 1)
`>> Virtual Server 2 80 http Service#cachepol cache_ext`	(Set the cache policy for the external customers)
`>> Virtual Server 2 80 http Service#comppol compress_ext`	(Set the compression policy for external customers)

HTTP/HTTPS Server Load Balancing

The Hypertext Transfer Protocol (HTTP) is a Layer 7 request-response protocol standard that lets you communicate between the client and the server. The client sends HTTP requests to the server, which sends messages, or responses, back to the client. The default port used for HTTP is 80, but it also can be used with other non-standard ports.

HTTPS, or HTTP Secure, combines HTTP with the SSL/TLS protocol, thereby enabling data encryption and secure server identification. The default port used for HTTPS is 443 but it also can be used with other non-standard ports.

Alteon enables you to load balance HTTP/HTTPS traffic.

Note: For a list of well-known ports identified by Alteon, see Supported Services and Applications, page 175.

This section describes the following topics:

Implementing HTTP/HTTPS Server Load Balancing

This section includes the following topics:

Procedures for Common HTTP and HTTPS Implementations

Use the following commands for common HTTP and HTTPS implementations.

To configure Alteon for HTTP load balancing on its well-known port (80)

Access the virtual server, and set the **HTTP virtual service.**

```
>> /cfg/slb/virt 1/service http
```

To configure Alteon for HTTPS load balancing on its well-known port (443)

Access the virtual server, and set the **HTTPS virtual service.**

```
>> /cfg/slb/virt 1/service https
```

To configure HTTP or HTTPS on a non-standard port

Use the same command with the requested port number. Alteon prompts you for the application for which you want to use this port (assuming it is not the well-known port of another application).

To configure Alteon for HTTP load balancing on port 88 (non-standard port)

Access the virtual server, and set the HTTP virtual service.

```
>> /cfg/slb/virt 1/service 88 http
```

To configure Alteon for HTTPS load balancing on port 444 (non-standard port)

Access the virtual server, and set the HTTP virtual service.

```
>> /cfg/slb/virt 1/service 444 https
```

HTTP/S Services Features

For HTTP/S services, you can use the following features:

- Content-Intelligent Server Load Balancing, page 219
- Content-Intelligent Application Services, page 237
- Advanced Content Modifications, page 244
- Content-Intelligent Connection Management, page 277

For HTTPS load balancing with SSL offloading, you must also supply a server certificate and associate an SSL policy. For more information, see Offloading SSL Encryption and Authentication, page 337.

Content-Intelligent Server Load Balancing

Alteon lets you load balance HTTP requests based on different HTTP header information, such as the "Cookie:" header for persistent load balancing, the "Host:" header for virtual hosting, or the "User-Agent" for browser-smart load balancing.

Alteon lets you load balance HTTP requests based on different HTTP protocol element information, such as headers, text, and XML.

Content-intelligent SLB uses Layer 7 content switching rules, which are defined per virtual service. These rules consist of a protocol-specific matching content class and an action, and are evaluated by priority based on their ID number. When Alteon matches a rule, the defined action is performed, and stops searching for matches. If no matching rule is found, Alteon performs the default service action configured at the service level itself.

Various actions are available per rule to provide further configuration granularity. For example, the actions for the HTTP rule include selecting a server group for load balancing (default), redirecting to an alterative location, or discarding the HTTP request altogether. Similarly, the default action configured at the service level can be any available action.

The content class is a matching object used for Layer 7 content switching rules. You can define a set of matching criteria that are based on the application type. For example, with an HTTP class, you can define matching criteria based on HTTP protocol elements such as URL, HTTP headers and so on. Each element can have multiple matching values, enabling advanced matching decisions to be evaluated. For example, "if (URL=my-site.com OR URL=my-site2.com) AND (Header=User-Agent: Internet-Explorer)".

Content classes can be nested using logical expressions. This enables you to use one class as part of the matching criteria for another class. For example, Class A includes a list of 100 mobile phone browser types. Classes B, C, and D need to match specific URLs for all the mobile phones from Class A. To configure this, Class A is defined as a logical expression matching the criteria of Classes B, C, and D. When you need to add additional mobile phone browsers to the list, you add them to Class A, and they are then propagated to Classes B, C, and D.

Notes

- Alteon supports Layer 7 content switching using an additional legacy configuration model that is based on Layer 7 strings. For related examples based on using Layer 7 strings see Appendix B - Content-Intelligent Server Load Balancing Not Using Layer 7 Content Switching Rules, page 809.
- To support IP fragment traffic when Layer 7 content switching is defined based on strings, use the **forceproxy** command under `/cfg/slb/virt/service/dbind` to force traffic through the Application Services Engine.

  For more information, see `/cfg/slb/virt<server number>/service <virtual port or application name>/dbind/forceproxy` option in the *Alteon Application Switch Operating System Command Reference*.

HTTP Layer 7 Content Switching

HTTP content switching uses HTTP content classes to match protocol element values. The HTTP content class enables matching with the following protocol elements: URL hostname, URL path, URL page name, URL page type, HTTP headers, cookies, text, and XML tags. Each value defined for the elements can be a simple text match or a regular expression (regex) match. When using text match, you can define the match for the exact string (equal), or for partial matching (contain, prefix, suffix). When using regex, the expression is always matched with contain logic, meaning that it can to appear anywhere in the matched element.

Alteon supports both HTTP1.0 and HTTP1.1 for Layer7 content switching.

Note: Alteon performs HTTP Layer 7 content switching before applying any modifications and is based on the original requests.

The following sample use cases illustrate the feature range of Layer 7 content switching:

- URL-Based Server Load Balancing, page 220
- Virtual Hosting, page 226
- Cookie-Based Preferential Load Balancing, page 227
- Browser-Smart Load Balancing, page 231
- XML/SOAP-Based Server Load Balancing, page 234
- URL Hashing for Server Load Balancing, page 236

URL-Based Server Load Balancing

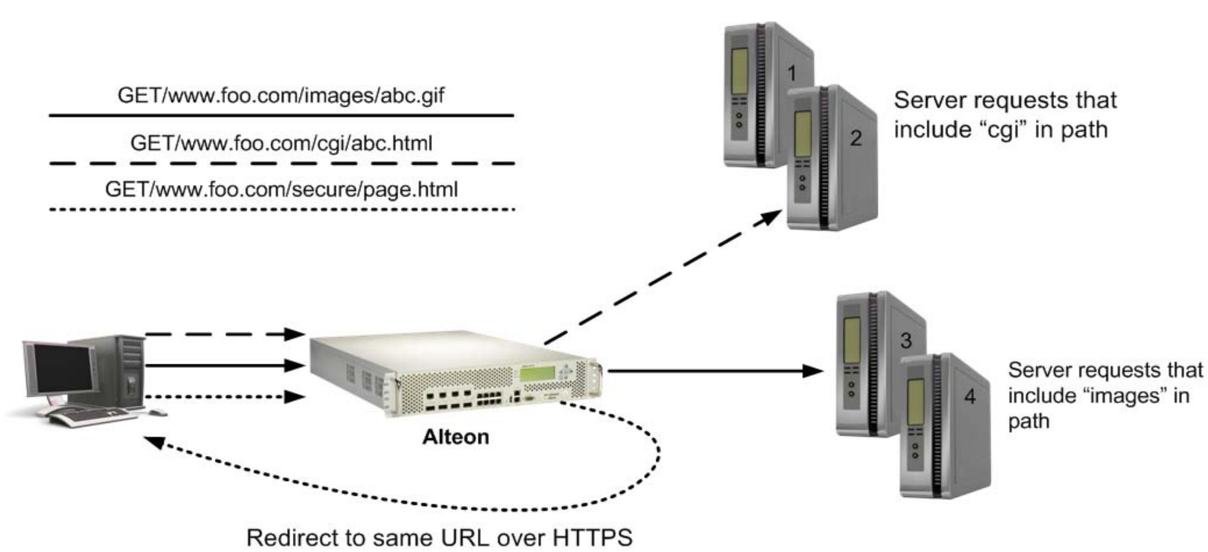
URL-based SLB enables you to optimize resource access and server performance. Content dispersion can be optimized by making load-balancing decisions on the entire path and filename of each URL.

Consider an example where the following criteria are specified for Layer 7 content switching:

- Requests with ".cgi" in the URL path are load balanced between Real Servers 1 and 2.
- Requests with "images" in the URL path are load balanced between Real Servers 3 and 4.
- Requests with "secure" in the URL path are redirected to same URL over secure HTTP (HTTPS).

Requests containing URLs with anything else are load balanced between Real Servers 1 through 4.

Figure 39: URL-Based SLB Scenario



To configure URL-based SLB

1. Before you can configure SLB string-based load balancing, ensure that Alteon is configured for basic SLB with the following tasks:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface.
   - Define each real server.
   - Define a real server group containing all servers (1 through 4), and set up health checks for the group.
   - Define a virtual server with a virtual service on port 80 (HTTP), and assign the real server group to service it. This will be the group servicing all "other" requests (not "cgi" or "images") containing Real Servers 1 through 4.
   - Enable SLB.
   - Enable client processing on the port connected to the clients.

   For information on how to configure your network for SLB, see Server Load Balancing, page 165.

2. Define the HTTP classes to be used for URL load balancing.
   — For an HTTP class to match a path that includes "cgi", do the following:

```
>> Server Load balance Resource# /cfg/slb/layer7/slb

>> Server Load balance Resource# cntclss
Enter Class id: cgi
------------------------------------------------------------
[HTTP Content Class cgi Menu]
      name     - Set the Descriptive HTTP content class name
      hostname - URL Hostname lookup Menu
      path     - URL Path lookup Menu
      filename - URL File Name lookup Menu
      filetype - URL File Type lookup Menu
      header   - Header lookup Menu
      cookie   - Cookie lookup Menu
      text     - Text lookup Menu
      xmltag   - XML tag lookup Menu
      logexp   - Set logical expression between classes
      copy     - Copy HTTP content class
      del      - Delete HTTP content class
      cur      - Display current HTTP content class

>> HTTP Content Class cgi# path
Enter path id: 1
------------------------------------------------------------
[Path 1 Menu]
      path     - Set path to match
      match    - Set match type
      case     - Enable/disable case sensitive for string matching
      copy     - Copy path
      del      - Delete path
      cur      - Display current path configuration
>> Path 1# path
Current path to match:
Enter new path to match: cgi

>> Path 1# match
Current matching type: include
Enter new matching type [sufx|prefx|equal|include|regex]: include

>> Path 1# case
Current Case sensitive matching: disabled
Enter new Case sensitive matching [d/e]: d
```

   — For an HTTP class to match a path that includes "images", perform the same procedure and specify "images" in the path parameter.
   — For an HTTP class to match a path that includes "secure", perform the same procedure and specify "secure" in the path parameter.
3. Create two additional server groups containing the real servers that only serve "cgi" (Real Servers 1 and 2), and the real servers that only serve "images" (Real Servers 3 and 4), and assign health checks to the groups.
4. Create Layer 7 content switching rules on the HTTP virtual service, including matching and traffic redirection.

— The following rule defines matching the "cgi" class and redirecting traffic to the group of Real Servers 1 and 2 for load balancing:

```
>> HTTP Load Balancing# /cfg/slb/virt 10/service http
------------------------------------------------------------
[Virtual Server 10 80 http Service Menu]
      name     - Set descriptive virtual service name
      http     - HTTP Load Balancing Menu
      cntrules - Content Based Services Rules Menu
      action   - Set action type of this service
      group    - Set real server group number
      redirect - Set application redirection location
      rport    - Set real port
      hname    - Set hostname
      cont     - Set BW contract for this virtual service
      pbind    - Set persistent binding type
      thash    - Set hash parameter
      tmout    - Set minutes inactive connection remains open
      ptmout   - Set in minutes for inactive persistent connection
      dbind    - Enable/disable/forceproxy delayed binding
      nonat    - Enable/disable only substituting MAC addresses
      direct   - Enable/disable direct access mode
      mirror   - Enable/disable session mirroring
      epip     - Enable/disable pip selection based on egress port/vlan
      winsize0 - Enable/disable using window size zero in SYN+ACK
      ckrebind - Enable/disable server rebalancing when cookie is absent
      del      - Delete virtual service
      cur      - Display current virtual service configuration
```

```
>> Virtual Server 10 80 http Service# cntrules
Enter Content Based Services Rule number (1-12800):     5
------------------------------------------------------------
[HTTP Content Rule 5 Menu]
      name     - Set descriptive content rule name
      cntclss - Set content class for this rule
      action   - Set action type for this rule
      group    - Set real server group number for this rule
      redirect - Set application redirection location for this rule
      copy     - Copy rule
      ena      - Enable rule
      dis      - Disable rule
      del      - Delete rule
      cur      - Display current rule configuration

>> HTTP Content Rule 5# name
     Current descriptive content rule name:
     Enter new descriptive content rule name: cgi rule

>> HTTP Content Rule 5# cntclss
Current content class:
Enter new content class or none: <<< Hit TAB to get list of all HTTP content
classes
a  cgi  images  my-class  secure
Enter new content class or none: cgi
For content class updates use /cfg/slb/layer7/slb
>> HTTP Content Rule 5# action
Current action type:    group
Enter new action type [group|redirect|discard] : group

>> HTTP Content Rule 5# group
Current real server group: 1
Enter new real server group [1-1024]: 2
```

— Define a similar content switching rule to match the "image" class, and redirect traffic to the group of Real Servers 3 and 4.

Tip: Because the content switching rule ID serves as rule matching priority, Radware recommends that you leave a gap between rule numbers that you create so you can easily place future rules within the current hierarchy. For example, create rules 1, 5, and 10 in the event that new rule 3 should be placed between rules 1 and 5, or new rule 7 should be placed between rules 5 and 10. If you need to move a rule to a different ID, use the copy command. This creates a copy of the rule from within the command that was used with a new ID, after which you can delete the original rule ID.

— The following rule defines matching the "secure" class and redirecting traffic to a secure site:

```
>> Virtual Server 10 80 http Service# /cfg/slb/virt/service

>> Virtual Server 10 80 http Service# cntrules
Enter Content Based Services Rule number (1-12800):     15
------------------------------------------------------------
[HTTP Content Rule 15 Menu]
      name     - Set descriptive content rule name
      cntclss - Set content class for this rule
      action   - Set action type for this rule
      group    - Set real server group number for this rule
      redirect - Set application redirection location for this rule
      copy     - Copy rule
      ena      - Enable rule
      dis      - Disable rule
      del      - Delete rule
      cur      - Display current rule configuration

>> HTTP Content Rule 15# name
Current descriptive content rule name:
Enter new descriptive content rule name: redirect secure request

>> HTTP Content Rule 15# cntclss
Current content class:
Enter new content class or none: secure
For content class updates use /cfg/slb/layer7/slb

>> HTTP Content Rule 15# action
Current action type:    group
Enter new action type [group|redirect|discard] : redirect

>> HTTP Content Rule 15# redirect ?
Usage: redirect <"redirection location"> |none
To use the same value as in the request, use:
        $PROTOCOL, $PORT, $HOST, $PATH, $QUERY
        Examples:
        http://www.mysite.com:8080/mypath
        https://$HOST/new/$PATH

>> HTTP Content Rule 15# redirect
Enter new redirect location: https://$HOST/$PATH?$QUERY
```

Note: The redirection location must consist of a full URL (including protocol, hostname, and path). The optional tokens enable dynamic copying of URL parts from the request to the redirect location, as a result preserving original client requests.

Virtual Hosting

Alteon enables individuals and companies to have a presence on the Internet in the form of a dedicated Web site address. For example, you can have a "www.site-a.com" and "www.site-b.com" instead of "www.hostsite.com/site-a" and "www.hostsite.com/site-b."

Service providers, on the other hand, do not want to deplete the pool of unique IP addresses by dedicating an individual IP address for each home page they host. By supporting an extension in HTTP 1.1 to include the host header, Alteon enables service providers to create a single virtual server IP address to host multiple Web sites per customer, each with their own hostname.

The following list provides more detail on virtual hosting with configuration information:

- An HTTP/1.0 request sent to an origin server (*not* a proxy server) is a partial URL instead of a full URL.

  The following is an example of the request that the origin server receives:

  ```
  GET /products/Alteon/ HTTP/1.0
  User-agent: Mozilla/3.0
  Accept: text/html, image/gif, image/jpeg
  ```

  The GET request does not include the hostname. From the TCP/IP headers, the origin server recognizes the hostname, port number, and protocol of the request.
- With the extension to HTTP/1.1 to include the HTTP Host: header, the above request to retrieve the URL www.radware.com/products/Alteon would look like this:

  ```
  GET /products/Alteon/ HTTP/1.1
  Host: www.radware.com

  User-agent: Mozilla/3.0
  Accept: text/html, image/gif, image/jpeg
  ```

  The Host: header carries the hostname used to generate the IP address of the site.
- Based on the Host: header, Alteon forwards the request to servers representing different customer Web sites.
- The network administrator needs to define a domain name as part of the 128 supported URL strings.

Note: It is also possible to provide virtual hosting for SSL encrypted sites (HTTPS), using the SSL protocol Server Name Indication (SNI) extension.

To configure virtual hosting based on HTTP Host: headers

1. Define the hostnames as HTTP content classes. If needed, associate multiple hostnames to the same HTTP content class. For an example of creating a content class, see URL-Based Server Load Balancing, page 220.

   Both domain names "www.company-a.com" and "www.company-b.com" resolve to the same IP address. In this example, the IP address is for a virtual server on Alteon.
2. Define dedicated real server groups for each of the customer's servers.

   Servers 1 through 4 belong to "www.company-a.com" and are defined as Group 1. Servers 5 through 8 belong to “www.company-b.com” and are defined as Group 2.
3. Create Layer 7 content switching rules on the virtual server's HTTP service, assigning HTTP content classes and groups to each rule. For an example of creating a content class, see URL Hashing for Server Load Balancing, page 236.
4. Alteon inspects the HTTP host header in requests received from the client.

— If the host header is "www.company-a.com," Alteon directs requests to the server group containing one of the Servers 1 through 4.

— If the host header is "www.company-b.com," Alteon directs requests to the server group containing one of the Servers 5 through 8.

Cookie-Based Preferential Load Balancing

Cookies can be used to provide preferential services for customers, ensuring that certain users are offered better access to resources than other users when site resources are scarce. For example, a Web server could authenticate a user via a password and then set cookies to identify them as "Gold," "Silver," or "Bronze" customers. Using cookies, you can distinguish individuals or groups of users and place them into groups or communities that get redirected to better resources and receive better services than all other users.

Note: Cookie-based persistent load balancing is described in Persistence, page 583.

Cookie-based preferential services enables, among others, the following supported use cases:

- Redirect higher priority users to a larger server or server group.
- Identify a user group and redirect them to a particular server group.
- Serve content based on user identity.
- Prioritize access to scarce resources on a Web site.
- Provide better services to repeat customers, based on access count.

Clients that receive preferential service can be distinguished from other users by one of the following methods:

- **Individual User**—A specific individual user can be distinguished by IP address, login authentication, or permanent HTTP cookie.
- **User Communities**—A set of users, such as "Premium Users" for service providers who pay higher membership fees than "Normal Users", can be identified by source address range, login authentication, or permanent HTTP cookie.
- **Applications**—Users can be identified by the specific application they are using. For example, priority can be given to HTTPS traffic that is performing credit card transactions versus HTTP browsing traffic.
- **Content**—Users can be identified by the specific content they are accessing.

Based on one or more of these criteria you can load balance requests to different server groups.

To configure cookie-based preferential load balancing

1. Before you can configure header-based load balancing, ensure that Alteon is configured for basic SLB with the following tasks:
   — Assign an IP address to each of the real servers in the server pool.
   — Define an IP interface.
   — Define each real server.
   — Assign servers to real server groups.
   — Define virtual servers and services.

   For information on how to configure your network for SLB, see Server Load Balancing, page 165.
2. Configure the Layer 7 content classes to match the various cookie values by which you need to load balance.

For example, to configure the cookie name session-id with the value gold:

```
>> Main# /cfg/slb/layer7/slb/cntclss/
Enter Class id: cookie-gold
------------------------------------------------------------
[HTTP Content Class cookie-gold Menu]
     name     - Set the Descriptive HTTP content class name
     hostname - URL Hostname lookup Menu
     path     - URL Path lookup Menu
     filename - URL File Name lookup Menu
     filetype - URL File Type lookup Menu
     header   - Header lookup Menu
     cookie   - Cookie lookup Menu
     text     - Text lookup Menu
     xmltag   - XML tag lookup Menu
     logexp   - Set logical expression between classes
     copy     - Copy HTTP content class
     del      - Delete HTTP content class
     cur      - Display current HTTP content class


>> HTTP Content Class cookie-gold# cookie/
Enter cookie id: 1
------------------------------------------------------------
[Cookie 1 Menu]
     cookie   - Set cookie to match
     match    - Set match type
     case     - Enable/disable case sensitive for string matching
     copy     - Copy cookie
     del      - Delete cookie
     cur      - Display current cookie configuration

>> Cookie 1# cookie
Current cookie to match: key= value=
Enter new cookie key to match or none []:session-id
Enter new cookie value to match or none []:gold
```

3. Repeat step 2 to define HTTP content classes to match the values silver and bronze.
4. Define real server groups to serve each client group according to their cookie value.

   For example, Gold clients are served by Real Servers 1 through 4 (Group 1), Silver clients are served by Real Servers 5 through 8 (Group 2), Bronze clients are served by Real server 9 through 10 (Group 3).

5. Define Layer 7 content switching rules in the HTTP virtual service to match each cookie value and redirect to the respective server group:

```
>> Main# /cfg/slb/virt 10/service http
------------------------------------------------------------
[Virtual Server 10 80 http Service Menu]
      name     - Set descriptive virtual service name
      http     - HTTP Load Balancing Menu
      cntrules - Content Based Services Rules Menu
      action   - Set action type of this service
      group    - Set real server group number
      redirect - Set application redirection location
      rport    - Set real port
      hname    - Set hostname
      cont     - Set BW contract for this virtual service
      pbind    - Set persistent binding type
      thash    - Set hash parameter
      tmout    - Set minutes inactive connection remains open
      ptmout   - Set in minutes for inactive persistent connection
      dbind    - Enable/disable/forceproxy delayed binding
      nonat    - Enable/disable only substituting MAC addresses
      direct   - Enable/disable direct access mode
      mirror   - Enable/disable session mirroring
      epip     - Enable/disable pip selection based on egress port/vlan
      winsize0 - Enable/disable using window size zero in SYN+ACK
      ckrebind - Enable/disable server rebalancing when cookie is absent
      del      - Delete virtual service
      cur      - Display current virtual service configuration

>> Virtual Server 10 80 http Service# cntrules
Enter Content Based Services Rule number (1-12800):     10
------------------------------------------------------------
[HTTP Content Rule 10 Menu]
      name     - Set descriptive content rule name
      cntclss - Set content class for this rule
      action   - Set action type for this rule
      group    - Set real server group number for this rule
      redirect - Set application redirection location for this rule
      copy     - Copy rule
      ena      - Enable rule
      dis      - Disable rule
      del      - Delete rule
      cur      - Display current rule configuration
>> HTTP Content Rule 10# name
Current descriptive content rule name:
Enter new descriptive content rule name: gold users

>> HTTP Content Rule 10# cntclss
Current content class:
Enter new content class or none: cookie-gold
For content class updates use /cfg/slb/layer7/slb


>> HTTP Content Rule 10# action
Current action type:    group
Enter new action type [group|redirect|discard] : group
```

```
>> HTTP Content Rule 10# group
Current real server group: 1
Enter new real server group [1-1024]: 10
```

6. Because a session cookie does not exist in the first request of an HTTP session, a default server group is needed to assign cookies to a None cookie HTTP request. Create a server group containing designated servers for example servers 1 through 10, and associate it to the HTTP virtual service as the fallback group.

```
>> Main# /cfg/slb/virt 10/service http
------------------------------------------------------------
[Virtual Server 10 80 http Service Menu]
      name     - Set descriptive virtual service name
      http     - HTTP Load Balancing Menu
      cntrules - Content Based Services Rules Menu
      action   - Set action type of this service
      group    - Set real server group number
      redirect - Set application redirection location
      rport    - Set real port
      hname    - Set hostname
      cont     - Set BW contract for this virtual service
      pbind    - Set persistent binding type
      thash    - Set hash parameter
      tmout    - Set minutes inactive connection remains open
      ptmout   - Set in minutes for inactive persistent connection
      dbind    - Enable/disable/forceproxy delayed binding
      nonat    - Enable/disable only substituting MAC addresses
      direct   - Enable/disable direct access mode
      mirror   - Enable/disable session mirroring
      epip     - Enable/disable pip selection based on egress port/vlan
      winsize0 - Enable/disable using window size zero in SYN+ACK
      ckrebind - Enable/disable server rebalancing when cookie is absent
      del      - Delete virtual service
      cur      - Display current virtual service configuration

>> Virtual Server 10 80 http Service# action
Current action type of this service: group
Enter new action type of this service [group|redirect|discard]: group
For load balancing group updates use /cfg/slb/virt/service/group
>> Virtual Server 10 80 http Service# group
Current real server group: 1
Enter new real server group [1-1024]: 15
```

Using this example, the following results:

— Request 1 comes in with no cookie. It is load balanced between servers in Group 15 (Real Servers 1 through 10) to receive a response and a cookie assigned.

— Request 2 comes in with a "Gold" cookie; it is load balanced between servers in Group 10 (Real Servers 1 through 4).

— Request 3 comes in with a "Silver" cookie; it is load balanced between servers in Group 11 (Real Servers 5 through 8).

— Request 4 comes in with a "Bronze" cookie; it is load balanced between servers in Group 12 (Real Servers 9 through 10).

— Request 5 comes in with a "Titanium" cookie; it is load balanced between servers in Group 15 (Real Servers 1 through 10), and because it does not contain an exact cookie match, it uses the fallback action.

Browser-Smart Load Balancing

HTTP requests can be directed to different servers based on browser type by inspecting the "User-Agent" header. For example:

```
GET /products/Alteon/ HTTP/1.0
User-agent: Mozilla/3.0
Accept: text/html, image/gif, image/jpeg
```

This also enables content-based load balancing based on device type (for example, laptop versus mobile phones), as each device type uses unique browser types. Since the list of browser user agents is quite extensive, it might be hard to manage and update them. To facilitate this kind of list referencing, using a content class enables nesting classes in a logical expression as part of the class.

Example Browser-Smart Load Balancing

- HTTP Class1—Includes a list of user-agents to match laptops and desktops.
- HTTP Class2—Includes a list of user agents to match mobile phones.
- HTTP Class3—Matched with URL my-site.com AND class1 and performs SLB using Server Group 1, providing regular web site content.
- HTTP Class4—Matched with URL my-site.com *and* class2 and redirects request to the mobile-phone specific version of the Web site located at mobile.my-site.com.
- HTTP Class5—Matched with URL mobile.my-site.com and performs SLB using Server Group 2 which contains the optimized "mobile" version of the web site.

To enable Alteon to perform browser-smart load balancing

This procedure is based on Example Browser-Smart Load Balancing, page 231.

1. Before you can configure browser-based load balancing, ensure that Alteon is configured for basic SLB with the following tasks:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface.
   - Define each real server.
   - Assign servers to real server groups (Group 1 and Group 2).
   - Define virtual servers and HTTP services.
2. Configure Class1 called "desktop-browsers" to match laptop or desktop browsers. In this example, Internet Explorer version 7 and later and Firefox are matched:

```
>> Main# /cfg/slb/layer7/slb/cntclss/
Enter Class id: desktop-browsers
------------------------------------------------------------
[HTTP Content Class desktop-browsers Menu]
      name     - Set the Descriptive HTTP content class name
      hostname - URL Hostname lookup Menu
      path     - URL Path lookup Menu
      filename - URL File Name lookup Menu
      filetype - URL File Type lookup Menu
      header   - Header lookup Menu
      cookie   - Cookie lookup Menu
      text     - Text lookup Menu
      xmltag   - XML tag lookup Menu
      logexp   - Set logical expression between classes
      copy     - Copy HTTP content class
      del      - Delete HTTP content class
      cur      - Display current HTTP content class

>> HTTP Content Class desktop-browsers# header
Enter header id: internet-explorer
------------------------------------------------------------
[Header internet-explorer Menu]
      header   - Set header to match
      match    - Set match type
      case     - Enable/disable case sensitive for string matching
      copy     - Copy header
      del      - Delete header
      cur      - Display current header configuration

>> Header internet-explorer# match
Current matching type for Header: name=include, value=include
Enter new matching type for Header name [eq|incl|regex][regex]:eq
Enter new matching type for Header value [eq|incl|regex][regex]:regex

>> Header internet-explorer# header
Current header to match: name= value=
Enter new header name to match or none []:User-agent
Enter new header value to match or none []:MSIE ([789].[0-9]+|1[01].[0-9]+)

>> Header internet-explorer# ..
HTTP Content Class desktop-browsers# header
Enter header id: firefox

[Header firefox Menu]

      header   - Set header to match
      match    - Set match type
      case     - Enable/disable case sensitive for string matching
      copy     - Copy header
      del      - Delete header
      cur      - Display current header configuration

>> Header firefox# header
Current header to match: name= value=
Enter new header name to match or none []:User-agent
Enter new header value to match or none []:Firefox
```

Regular expressions (regex) can be used to match multiple browser user agents with a single value. Additional desktop or laptop browser user agents can be added to this class.

3. Configure Class2 to match mobile browsers user-agent header values using the same procedure as Class1 in step 2.
4. Configure Class3 to match URL my-site.com *and* class1 (desktop-browsers) by using the logical expression option in the *Class* menu:

```
>> Server Load balance Resource# cntclss
Enter Class id: class3
------------------------------------------------------------
[HTTP Content Class class3 Menu]
      name     - Set the Descriptive HTTP content class name
      hostname - URL Hostname lookup Menu
      path     - URL Path lookup Menu
      filename - URL File Name lookup Menu
      filetype - URL File Type lookup Menu
      header   - Header lookup Menu
      cookie   - Cookie lookup Menu
      text     - Text lookup Menu
      xmltag   - XML tag lookup Menu
      logexp   - Set logical expression between classes
      copy     - Copy HTTP content class
      del      - Delete HTTP content class
      cur      - Display current HTTP content class

>> HTTP Content Class class3# hostname
Enter hostname id: 1
------------------------------------------------------------
[Hostname 1 Menu]
      hostname - Set hostname to match
      match    - Set match type
      copy     - Copy hostname
      del      - Delete hostname
      cur      - Display current hostname configuration

>> Hostname 1# hostname
Current hostname to match:
Enter new hostname to match: my-site.com

>> Hostname 1# ..

>> HTTP Content Class class3# logexp
Current logical expression:
Enter new logical expression:
Enter logical expression: desktop-browsers
```

5. Configure Class4 to match URL my-site.com *and* Class2 (mobile-browsers) using the procedure in step 4.
6. Configure Class5 matched with URL mobile.my-site.com using the same procedure in the URL-based content load balancing example (URL Hashing for Server Load Balancing, page 236).
7. Configure an HTTP Layer7 content switching rule in the HTTP virtual service to match Class3 (with URL my-site.com *and* desktop-browsers), and perform load balancing using Server Group 1.

8. Configure an HTTP Layer7 content switching rule in the HTTP virtual service to match Class4 (with URL my-site.com *and* mobile-browsers), and perform HTTP redirection to http://mobile.my-site.com.
9. Configure an HTTP Layer7 content switching rule in the HTTP virtual service to match Class5 (with URL mobile.my-site.com), and perform load balancing using Server Group2.

XML/SOAP-Based Server Load Balancing

With the evolution of Web applications, much of HTTP traffic is based on SOAP messages or other XML formatted data transfer. Alteon can perform content switching based on specific XML tag attributes or tag values. The following is a SOAP message written in XML format and sent over HTTP protocol:

Example XML/SOAP-Based Message

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.example.org/stock">
<m:GetStockPrice   StockEx=NASDAQ>
<m:StockName>IBM</m:StockName>
</m:GetStockPrice>
</soap:Body>

</soap:Envelope>
```

In this message, Alteon performs content switching based on a tag attribute such as the tag GetStockPrice with the attribute StockEx, which has the value **NASDAQ**. Alternatively, Alteon can perform content switching based on a tag value like the tag StockName with the value **IBM**.

To configure XML-based load balancing

1. Before you can configure XML-based load balancing, ensure that Alteon is configured for basic SLB with the following tasks:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface.
   - Define each real server.
   - Assign servers to real server groups.
   - Define virtual servers and services.

   For information on how to configure your network for SLB, see How Server Load Balancing Works, page 166.

2. Configure the Layer 7 content classes to match the XML tags values you need to load balance by. For example, configuring the XML tag StockName from Example XML/SOAP-Based Message, page 234:

```
>> Main# /cfg/slb/layer7/slb/cntclss/
Enter Class id: StockName-IBM
------------------------------------------------------------
[HTTP Content Class StockName-IBM Menu]
      name     - Set the Descriptive HTTP content class name
      hostname - URL Hostname lookup Menu
      path     - URL Path lookup Menu
      filename - URL File Name lookup Menu
      filetype - URL File Type lookup Menu
      header   - Header lookup Menu
      cookie   - Cookie lookup Menu
      text     - Text lookup Menu
      xmltag   - XML tag lookup Menu
      logexp   - Set logical expression between classes
      copy     - Copy HTTP content class
      del      - Delete HTTP content class
      cur      - Display current HTTP content class

>> HTTP Content Class StockName-IBM# xmltag/
Enter xmltag id: ibm
------------------------------------------------------------
[XML tag ibm Menu]
      xmltag   - Set XML tag to match
      match    - Set match type
      case     - Enable/disable case sensitive for string matching
      copy     - Copy XML tag
      del      - Delete XML tag
      cur      - Display current XML tag configuration

>> XML tag ibm# xmltag
Current XML tag to match: pathtag= value=
Enter new XML path and tag name to match or none:\GetStockPrice\StockName
Enter new value to match or none []:IBM
```

Note: To reference a tag attribute, use the @ sign in the tag path before the tag attribute name.

3. Configure additional Layer 7 content classes with different match values (for example Microsoft, Goggle, and so on). You can also include multiple match values in each class (for example, IBM or HP).
4. Configure server groups with the real servers that will serve each of the XML tag values, and assign health checks to them.
5. Configure a Layer 7 content rule in the HTTP virtual service, using the defined XML-based content classes and groups. For more information on how to configure content switching rules, see URL-Based Server Load Balancing, page 220.

URL Hashing for Server Load Balancing

By default, hashing algorithms use the IP source address and/or IP destination address (depending on the application area) to determine content location. The default hashing algorithm for SLB is the IP source address. By enabling URL hashing, requests going to the same page of an origin server are redirected to the same real server or cache server.

Load Balancing Non-transparent Caches

You can deploy a cluster of non-transparent caches and use the virtual server to load balance requests to the cache servers. The client's browser is configured to send Web requests to a non-transparent cache (the IP address of the configured virtual server).

If hash is selected as the load-balancing algorithm, Alteon hashes the source IP address to select the server for SLB. Under this condition, Alteon may not send requests for the same origin server to the same proxy cache server. For example, requests made from a client to http://radwarealteon.com from different clients may get sent to different caches.

Figure 40: Load Balancing Non-transparent Caches

Configuring URL Hashing

You can direct the same URL request to the same cache or proxy server by using a virtual server IP address to load balance proxy requests. By configuring hash or minmisses as the metric, Alteon uses the number of bytes in the URI to calculate the hash key.

If the host field exists and Alteon is configured to look into the Host: header, Alteon uses the Host: header field to calculate the hash key.

To configure URL hashing

1. Before you can configure URL hashing, ensure that Alteon is configured for basic SLB with the following tasks:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface.
   - Define each real server.
   - Assign servers to real server groups.
   - Define virtual servers and services.
   - Configure load-balancing algorithm for hash or minmiss.
   - Enable SLB.
   - Define server port and client port.

   For information on how to configure your network for SLB, see Server Load Balancing, page 165.

2. Enable URL hashing.

```
>> # /cfg/slb/virt 1
>> Virtual Server 1 # service 80
>> Virtual Server 1 http Service # http/httpslb urlhash
Enter new hash length [1-255]: 25
```

Hashing is based on the URL, including the HTTP `Host:` header (if present), up to a maximum of 255 bytes.

3. Set the metric for the real server group to `minmisses` or `hash`.

```
>> # /cfg/slb/group 1/metric <hash|minmisses>
```

HTTP Normalization

Alteon normalizes characters in the HTTP strings that are encoded to real characters and performs URL path traversal reversals before performing rule matching for HTTP Layer 7 content switching and HTTP modifications. After matching the content, it is sent back to the real servers in its original format.

You can enable or disable HTTP normalization via the HTTP Virtual Service menu. For more information, see the *Alteon Application Switch Operating System Command Reference*.

Content-Intelligent Application Services

Alteon lets you modify HTTP responses and requests to achieve the following purposes:

- Sending Original Client IPs to Servers, page 237
- Controlling Server Response Codes
- Changing URLs in Server Responses, page 239
- Enhancing Server Security by Hiding Server Identity, page 241
- Enhancing Security by Hiding Page Locations, page 241
- Replacing Free Text in Server Responses, page 243

Sending Original Client IPs to Servers

Alteon can insert the inclusion of the X-Forwarded-For header in client HTTP requests in order to preserve client IP information. This feature is useful in proxy mode, where the client source IP information is replaced with the proxy IP address. However, it may also be used for all Layer 4 load balancing in both proxy and non-proxy mode, if there is a need to include the X-Forwarded-For header. This feature is supported for Layer 4 and Layer 7.

Note: To enable X-Forwarded-For, you need to either set delayed binding to full proxy mode and configure a PIP or enable DAM.

To configure Alteon to insert the X-Forwarded-For header

1. Ensure that Alteon is configured for basic SLB:
   - — Assign an IP address to each of the real servers in the server pool.
   - — Define an IP interface.
   - — Define each real server.

— Assign servers to real server groups.

— Define virtual servers and services.

2. Enable client proxy operation mode on the real servers used in load balancing.

```
>> # /cfg/slb/real 1/adv/proxy ena
```

3. On the virtual server attached to the real servers, enable the X-Forwarded-For header:

```
>> # /cfg/slb/virt 1/service 80/http/xforward ena
```

Note: Session mirroring is not supported when X-Forward-For is enabled.

4. Apply and save the configuration.

```
>> # apply
>> # save
```

Controlling Server Response Codes

Alteon can intercept server responses and update the HTTP error messages sent to the user by the server.

You can change the error code generated by the server, edit the error reason, or redirect to a different HTTP location. When redirecting, the hostname specified should include the protocol. For example: HTTP://www.a.com and not www.a.com.

You can define multiple error codes per service if all use the same behavior. When editing the *errcode* configuration, type all the relevant codes. To configure multiple error codes, type the codes separated with a comma. For example: **403, 504**.

Make sure that you define whether the new values are added to or replace the existing values. For example, if the current configuration is for X and you update the code to Y, then X is removed. To configure both X and Y, type both ports separated with a comma. For example: **X, Y**.

When editing the existing configuration, the current configuration is displayed in square brackets [ ] to facilitate the update. To clear the existing configuration of the page name and page type, enter **None**.

To configure server response code control

1. Ensure that Alteon is configured for basic SLB:

— Assign an IP address to each of the real servers in the server pool.

— Define an IP interface.

— Define each real server.

— Assign servers to real server groups.

— Define virtual servers and services.

2. Access error code handling, enable it and then enter the error codes to be changed.

```
>> Main# /cfg/slb/virt 1/service 80/http/errcode
>> Enter status enabled/disabled [e:d:c] [c]: e
>> Enter match error code(s), e.g 203, 204 []: 504
```

3. Enable or disable HTTP redirection, and then enter a new error code and a new error reason.

```
>> Use http redirection? [y:n]: y
>> Enter URL for redirection: http://www.changesite.com
>> Enter new error code []: 302
```

Example

To change server responses with error code 333 or 444 to a redirection to **www.alternatesite.com/trythis**, use the following configuration:

```
>> HTTP Load Balancing# errcode
Current error code configuration:
Disabled << It should work only if it's Enabled

Enter enabled/disabled or clear [e|d|c] [c]: e
Enter match error code(s), e.g 203,204 []: 333,444
Use http redirection [y/n] [y]:
Enter URL for redirection []: http://www.alternatesite.com/trythis
```

Changing URLs in Server Responses

Alteon lets you update the links within the server responses that do not match the actual object location on the servers. By changing the URL, the server responses are updated with the correct URLs. This can be used when the content of the servers has been moved, but the links have not yet been updated. You can match the hostname, URL, page and page type within the server responses, and update the URL, page and page type within the server responses.

When editing the existing configuration, the current configuration is displayed in square brackets [ ] to facilitate the update. To clear the existing configuration of the page name and page type, enter **None**.

By default, URL path change modification is disabled.

Note: Using these commands results in path modifications only. The protocol (HTTP or HTTPS) and the port (when specified) are not modified.

To change URLs in server responses

1. Ensure that Alteon is configured for basic SLB:
   — Assign an IP address to each of the real servers in the server pool.
   — Define an IP interface.
   — Define each real server.
   — Assign servers to real server groups.
   — Define virtual servers and services.

2. Access and then enable URL path change.

```
>> Main# /cfg/slb/virt 1/service 80/http/urlchang
>> Enter enabled/disabled or clear [e|d|c] [d]: e
>> Enter hostname match type [sufx|prefx|eq|incl|any] [any]: eq
>> Enter hostname to match: www.a.com
>> Enter path match type [sufx|prefx|eq|incl|any] [any]: eq
>> Enter path to match: www.path.com
>> Enter page name to match or none []: test
>> Enter page type to match or none: html
>> Enter path action type [insert:replace:remove:none]:
```

3. Depending on the action type, enter the required parameters.

Table 24: URL In Server Responses Action Parameters

Action	Action Parameters
None	No action is taken. Continue to the next step
Remove	The matched path section is removed. Continue to the next step

Insert	The following path section is inserted.
	`>> Enter path to insert []:`
	`>> Insert the specified path before or after the matched section? [b/a]:`

Replace	The following path section is removed. `>> Enter new path to replace the matched section:`

4. Enter the page name and path type to be used for the path change.

```
>> Enter new page name or none []: newpagename
>> Enter new page type or none []: html
```

Example

To change links in server responses with paths starting with “abcd” to start with “aaabcd”, use the following configuration:

```
>> HTTP Load Balancing# urlchang

Note: The match condition applies to the response.

Current URL Change configuration disabled

Enter enabled/disabled or clear [e|d|c] [c]: e
Enter hostname match type [sufx|prefx|eq|incl|any] [any]:
Enter path match type [sufx|prefx|eq|incl|any] []: prefx
Enter path to match []: abcd
Enter page name to match or none []:
Enter page type to match or none []:
Enter path action type [insert|replace|remove|none] []: insert
Enter path to insert []: aa
Insert the specified path before or after the matched section? [b|a] []: b
Enter new page name or none []:
Enter new page type or none []:
```

Enhancing Server Security by Hiding Server Identity

Alteon lets you modify server responses by replacing HTTP headers that include information about the server computer and operating system. By default modifying server responses is disabled.

To hide the server identity

1. Ensure that Alteon is configured for basic SLB:
   — Assign an IP address to each of the real servers in the server pool.
   — Define an IP interface.
   — Define each real server.
   — Assign servers to real server groups.
   — Define virtual servers and services.
2. Access and enable server resource cloaking.

```
>>Main# /cfg/slb/virt 1/service 80/http/cloaksrv ena
```

Enhancing Security by Hiding Page Locations

Alteon enables you to hide links within the server responses to avoid exposing the internal data structure on the server. When hiding path locations, specified URLs within the server responses are removed and added back to the client requests.

For example, if the user wants to hide a path with "newsite", all links such as **www.site.com/newsite/page.htm** appear to the user as **www.site.com/page.htm**. Therefore, **newsite** will be added at the beginning of the path to all requests to **www.site.com**.

You can enable, disable, or clear the path obfuscation configuration.

When editing the existing configuration, the current configuration is displayed in square brackets [ ] to facilitate the update. To clear the existing configuration of the page name and page type, enter "None".

Note: Using these commands results in path modifications only. The protocol (HTTP or HTTPS) and the port (when specified) are not modified.

To hide page locations

1. Ensure that Alteon is configured for basic SLB:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface.
   - Define each real server.
   - Assign servers to real server groups.
   - Define virtual servers and services.
2. Access and then enable URL path change:

```
>> Main# /cfg/slb/virt 1/service 80/http/pathhide
>> Enter enabled/disabled or clear [e:d:c] [e]: e
```

3. Enter the hostname type and path type to be matched.

```
>> Enter hostname match type [sufx:prefx:eq:incl:any] [any]:
>> Enter hostname to match:
>> Enter path match type [sufx:prefx:eq:incl:any] [any]:
>> Enter path to remove:
```

Example

In all URLs in the server responses that use **www.site.com/test/**, "test" should be removed from the path. For example, when **www.site.com/test/a/page.html** appears in the response, it is translated to **www.site.com/a/page.html**.

Client requests are modified the opposite way. For example, a request from the user to **www.site.com** is modified and sent to the server as **www.site.com/test**. A request to **www.site.com/my.page** is modified to **www.site.com/test/my.page**.

To perform this action, use following configuration:

```
>> HTTP Load Balancing# pathhide

Note: The match condition applies to the response.

Current path hide (obfuscate) configuration: disabled

Enter enabled/disabled or clear [e|d|c] [c]: e
Enter hostname match type [sufx|prefx|eq|incl|any] [any]: eq
Enter hostname to match []: www. site.com
Enter path match type [sufx|prefx|eq|incl|any] []: prefx
Enter path to remove []: test
```

Replacing Free Text in Server Responses

Alteon lets you remove or replace free text in server responses.

To replace free text in server responses

1. Ensure that Alteon is been configured for basic SLB:
   — Assign an IP address to each of the real servers in the server pool.
   — Define an IP interface.
   — Define each real server.
   — Assign servers to real server groups.
   — Define virtual servers and services.
2. Access and enable URL path change, and define the action type.

```
>> Main# /cfg/slb/virt 1/service 80/http/textrep
>> Enter status enabled/disabled or clear [e:d:c] [d]: e
>> Enter action [replace:remove] []:
```

3. Depending on the action type, enter the required parameters.

Table 25: Replacing Free Text in Server Responses Action Parameters

Action	Action Parameters
Remove	The matched text to be removed: `>> Enter text to remove []:`

Replace	The matched text to be replaced:
	`>> Enter text to be replaced []:`
	`>> Enter new text[]:`

Example

To remove the text "this is a dummy line" from server responses, use the following configuration:

```
>> HTTP Load Balancing# textrep

Current text replace configuration: disabled

Enter enabled/disabled or clear [e|d|c] [c]: e
Enter action [replace|remove] []: remove
Enter text to remove []: this is a dummy line
```

Advanced Content Modifications

In various cases there is a need to control the content returned by a Web application or sent to the Web application. This can include modifying URLs of objects, modifying cookies or other HTTP headers or modifying any text in the HTTP or HTML.

Alteon lets you modify different types of HTTP elements. Following are the HTTP elements that can be modified:

- **HTTP Headers**—Can be inserted, replaced, or removed. See Configuring HTTP Modification for HTTP Headers, page 246.
- **Cookies**—Can be replaced or removed. See Configuring HTTP Modification for Cookies, page 249.
- **File type**—File type elements within the HTTP requests can be replaced. See Configuring HTTP Modifications for the HTTP File Type, page 254.
- **Status Line**—Status line elements within the HTTP responses can be replaced. See Configuring HTTP Modification for HTTP Status Line, page 255.
- **URL**—Within requests or responses, headers or entire message body can be replaced. See Configuring HTTP Modification for URL Elements, page 256.
- **Text**—Any text elements can be replaced in HTTP headers or the entire message body. See Configuring HTTP Modification for Text Elements, page 265.

Depending on the element type, these modifications are applied to the header only or both header and body of the HTTP responses or requests.

About Rule Lists

You can configure lists of HTTP modification rules (rule lists), and then associate a rule list to services. The same HTTP modification rule list can be reused across virtual services. The rule-list identifier is a name. Within each rule list, you create rules for each HTTP element type.

For more information on associating rule lists to services, see Associating HTTP Modification Rules to a Service, page 267.

About Rules

HTTP Modification rules are based on different types of HTTP elements. A rule can be added, removed, or copied. The rules are evaluated according to their priority, with the lowest number getting evaluated first. The maximum number of rules in a rule list is 128.

When defining a rule, you first set the rule ID, and then select the desired element on which the rule will be based on. You cannot update a rule after setting its rule ID and element. To change the element, the rule must be deleted and a new rule created.

Once a rule is matched and acted upon, the rest of the rules in the list are not evaluated for that object. Rules are displayed in numerical order.

Tip: Radware recommends that you leave a gap between rule numbers that you create so you can easily place future rules within the current hierarchy. For example, create rules 1, 5, and 10 in the event that new rule 3 should be placed between rules 1 and 5, or new rule 7 should be placed between rules 5 and 10.

If more than one rule match the same element, only the first modification will take place, that is, you cannot match and modify an element that has already been modified.

Note: You have to enable the desired rule list and rule, and apply the changes for the modifications to take effect.

For information on how to associate rules to a virtual service, see Associating HTTP Modification Rules to a Service, page 267.

The following is a list of all HTTP elements and their supported actions:

Table 26: HTTP Elements and Their Supported Actions

Element	Action
Header	• Configuring the Replace Action for HTTP Headers, page 246 • To configure the remove action for HTTP Headers, page 247 • To configure the insert action for HTTP headers, page 248
Cookie	• To configure the replace action for cookies, page 250 • To configure the remove action for cookies, page 251 • To configure the insert action for cookies, page 252
File type	• To configure HTTP modification for the HTTP file type, page 254
Status line	• To configure the replace action for the HTTP status line, page 255
URL	• Configuring Modification for HTTP URL Elements, page 257
Text	• To configure the replace action for an HTTP text element, page 265 • To configure the remove action for the HTTP text element, page 266

Configuring HTTP Modification for HTTP Headers

When creating a rule for a HTTP header element, the following actions can be defined:

- Configuring the Replace Action for HTTP Headers, page 246
- To configure the remove action for HTTP Headers, page 247
- To configure the insert action for HTTP headers, page 248

Configuring the Replace Action for HTTP Headers

This action replaces the matched header name and value with the new header name and value specified. Only the first encountered matching header field of the original string in the message is replaced. A value match means a complete word within the value of the header.

To configure the replace action for header elements

Note: The numbers and names in this procedure are examples only.

1. Access HTTP Modification rule list configuration via the *Layer 7* menu, enter a rule list ID, and enable the rule list.

```
>>Main# /cfg/slb/layer7/httpmod
>>Enter HTTP Modification rule-list id (alphanumeric): http-mod-list
>>HTTP Modification rule-list http-mod-list# ena
```

2. Enter **rule**, the rule ID number, and then enter the desired element type.

```
>>HTTP Modification rule-list http-mod-list# rule
>>Enter HTTP Modification rule number (1-128):5
>>Element can be one of: url, header, cookie, filetype, statusline, text
>>Enter element to be modified: header
>>header Modification http-mod-list Rule 5
```

3. Enter **action** to access the *Rule Action* menu, and then enter **replace** to set the new rule replace action.

```
>>header Modification http-mod-list Rule 5 # action
>>Enter rule action [insert|replace|remove]: replace
>>Enter header field to replace:
>>Enter value to replace or none:
>>Enter new header field or none:
>>Enter new value or none:
```

Note: To replace only the content of the header field (the value) and not the header field name, enter the same header field name in **new header field** prompt.

4. Enter **directn** to set the rule direction, and then enter the rule direction: request or response.

```
>>header Modification http-mod-list Rule 5 # directn
>>Enter new rule direction [req:resp] [req]:
```

Example

To replace the value of the HTTP Header "My-Header" in all client requests, so that the first match of the string "ABC" is replaced with "XYZ", use the following configuration:

```
>>HTTP Modification http_mod Rule 2# cur
Current rule: 2
    enabled, name My_list
    action replace header
        from: HEADER=My-Header, VALUE=ABC
        to: HEADER=My-Header, VALUE=XYZ
    direction request
```

The header value is only replaced if the original string is an exact match of the complete replacement value. In this example, if the value is “ABCABC”, it is not replaced since it is not an exact match.

To configure the remove action for HTTP Headers

With this action, the entire matching header field is removed. The value specified is used to decide whether the header should be removed. Only the first encountered matching header field of the original string in the message is removed. A value match means a complete word within the value of the header.

Note: The numbers and names in this procedure are examples only.

1. Access HTTP Modification rule list configuration via the *Layer 7* menu, enter a rule list ID, and enable the rule list.

```
>>Main# /cfg/slb/layer7/httpmod
>>Enter HTTP Modification rule-list id (alphanumeric): http-mod-list
>>HTTP Modification rule-list http-mod-list# ena
```

2. Enter **rule**, the rule ID number, and then enter the desired element type.

```
>>HTTP Modification rule-list http-mod-list# rule
>>Enter HTTP Modification rule number (1-128):5
>>Element can be one of: url, header, cookie, filetype, statusline, text
>>Enter element to be modified: header
>>header Modification http-mod-list Rule 5
```

3. Enter **action** to access the *Rule Action* menu, and then enter **remove** to set the new rule remove action.

```
>>header Modification http-mod-list Rule 5 # action
>>Current rule action:
>>Enter new rule action [insert|replace|remove]: remove
>>Enter header field to remove:
>>Enter value to remove:
```

4. Enter **directn** to set the rule direction, and then enter the rule direction: request or response.

```
>>header Modification http-mod-list Rule 5 # directn
>>Enter new rule direction [req:resp] [req]:
```

Example

To remove HTTP Header "Test-Header" from all server responses, use the following configuration:

```
 >> HTTP Modification http_mod Rule 2# cur
Current rule: 2
    enabled, name My_list
    action remove header
        HEADER=Test_Header
    direction request
```

If you leave the value empty, the complete header is removed, regardless of the value of the header. If you set the cookie value, the cookie is only removed when both the key and value match.

To configure the insert action for HTTP headers

This action inserts the header field and value at the beginning of the header area. A value match means a complete word within the value of the header.

Note: The numbers and names in this procedure are examples only.

1. Access HTTP Modification rule list configuration via the *Layer 7* menu, enter a rule list ID, and enable the rule list.

```
>>Main# /cfg/slb/layer7/httpmod
>>Enter HTTP Modification rule-list id (alphanumeric): http-mod-list
>>HTTP Modification rule-list http-mod-list# ena
```

2. Enter **rule**, the rule ID number, and then enter the desired element type.

```
>>HTTP Modification rule-list http-mod-list# rule
>>Enter HTTP Modification rule number (1-128):5
>>Element can be one of: url, header, cookie, filetype, statusline, text
>>Enter element to be modified: header
>>header Modification http-mod-list Rule 5
```

3. Enter **action** to access the *Rule Action* menu, and then enter **insert** to set the new rule insert action.

```
>>header Modification http-mod-list Rule 5 # action
>>Current rule action:
>>Enter new rule action [insert|replace|remove]: insert
>>Enter header field to insert:
>>Enter value to insert:
```

4. For the insert action, you can define a match criteria. If you define a match criteria, the insert is performed only if the match is met.

   Enter the element to be matched for insertion.

```
>>Element to match can be one of url, header, cookie, filetype, statusline,
>>text, regex, none
>>Enter element to match []:
```

5. Based on the selected match element, enter the required parameters. For more information, refer to the *Alteon Application Switch Operating System Command Reference*.
6. Enter **directn** to set the rule direction, and then enter the rule direction: request or response.

```
>>header Modification http-mod-list Rule 5 # directn
>>Enter new rule direction [req:resp] [req]:
```

Example

To insert the HTTP Header "New-Header" with value of "VALUE" to all client requests for www.site.com/path/new, use the following configuration:

```
 >> HTTP Modification http_mod Rule 2# cur
Current rule: 2
    enabled, name My_list
    action insert header
  HEADER=New-Header, VALUE=VALUE
  MATCH=url, URL=www.site.com, PATH=/path/new
    direction request
```

Configuring HTTP Modification for Cookies

When using cookies for request, the Cookies HTTP header is updated. When using cookies for responses, the Set-Cookie header is updated.

When creating a rule for a cookie element, the following actions can be defined:

- To configure the replace action for cookies, page 250
- To configure the remove action for cookies, page 251
- To configure the insert action for cookies, page 252

Note: When both cookie-based pbind is used and HTTP modifications on the same cookie header are defined, Alteon performs both. This may lead to various application behaviors and should be done with caution.

To configure the replace action for cookies

This action replaces the matched cookie key and value with the new specified key and value. When the direction is set to request, the cookie header is modified. When the direction is set to response, the Set-Cookie header is modified.

Note: The numbers and names in this procedure are examples only.

1. Access HTTP Modification rule list configuration via the *Layer 7* menu, enter a rule list ID, and enable the rule list.

```
>>Main# /cfg/slb/layer7/httpmod
>>Enter HTTP Modification rule-list id (alphanumeric): http-mod-list
>>HTTP Modification rule-list http-mod-list# ena
```

2. Enter **rule**, the rule ID number, and then enter the desired element type.

```
>>HTTP Modification rule-list http-mod-list# rule
>>Enter HTTP Modification rule number (1-128):5
>>Element can be one of: url, header, cookie, filetype, statusline, text
>>Enter element to be modified: cookie
>>cookie Modification http-mod-list Rule 5
```

3. Enter **action** to access the *Rule Action* menu, and then enter **replace** to set the new rule replace action.

```
>>cookie Modification http-mod-list Rule 5 # action
>>Current rule action:
>>Enter new rule action [insert|replace|remove]: replace
>>Enter cookie key to replace or none:
>>Enter cookie value to replace or none:
>>Enter new cookie key or none:
>>Enter new cookie value or none:
```

4. Enter **directn** to set the rule direction, and then enter the rule direction: request or response.

```
>>cookie Modification http-mod-list Rule 5 # directn
>>Enter new rule direction [req:resp] [req]:
```

Example

To change the value of the cookie "User-Type" from "Gold" to "Premium" in all client requests, use the following configuration:

```
>>HTTP Modification rule-list mylist# cur
Current rule-list: mylist enabled
  10:
    enabled
    action replace cookie
        from: KEY=User-Type, VALUE=Gold
        to: KEY=User-Type, VALUE=Premium
    direction request
```

To configure the remove action for cookies

With this action, the entire key=value pair is removed from the header. The value specified is used to decide whether the header should be removed. When the direction is set to request, the cookie header is modified. When the direction is set to response, the Set-Cookie header is modified.

Note: The numbers and names in this procedure are examples only.

1. Access HTTP Modification rule list configuration via the *Layer 7* menu, enter a rule list ID, and enable the rule list.

```
>>Main# /cfg/slb/layer7/httpmod
>>Enter HTTP Modification rule-list id (alphanumeric): http-mod-list
>>HTTP Modification rule-list http-mod-list# ena
```

2. Enter **rule**, the rule ID number, and then enter the desired element type.

```
>>HTTP Modification rule-list http-mod-list# rule
>>Enter HTTP Modification rule number (1-128):5
>>Element can be one of: url, header, cookie, filetype, statusline, text
>>Enter element to be modified: cookie
>>cookie Modification http-mod-list Rule 5
```

3. Enter **action** to access the *Rule Action* menu, and then enter **remove** to set the new rule remove action.

```
>>cookie Modification http-mod-list Rule 5 # action
>>Current rule action:
>>Enter new rule action [insert|replace|remove]: remove
>>Enter cookie key to remove:
>>Enter cookie value to remove:
```

4. Enter **directn** to set the rule direction, and then enter the rule direction: request or response.

```
>>cookie Modification http-mod-list Rule 5 # directn
>>Enter new rule direction [req:resp] [req]:
```

Example

To remove the Set-Cookie for a cookie named "Old-Cookie" from all server responses, use the following configuration:

```
>>URL Modification rule-list mylist# cur
Current rule-list: mylist enabled
  10:
    enabled
    action remove cookie
        KEY=Old-Cookie
    direction response
```

When you leave the cookie value empty, the cookie is removed.

If you set the cookie value, the cookie is removed only when both the key and value match.

To configure the insert action for cookies

This action inserts the cookie header at the beginning of the header area, after the request line. When the direction is set to request, the cookie header is modified. When the direction is set to response, the Set-Cookie header is modified.

Note: The numbers and names in this procedure are examples only.

1. Access HTTP Modification rule list configuration via the *Layer 7* menu, enter a rule list ID, and enable the rule list.

```
>>Main# /cfg/slb/layer7/httpmod
>>Enter HTTP Modification rule-list id (alphanumeric): http-mod-list
>>HTTP Modification rule-list http-mod-list# ena
```

2. Enter **rule**, the rule ID number, and then enter the desired element type.

```
>>HTTP Modification rule-list http-mod-list# rule
>>Enter HTTP Modification rule number (1-128):5
>>Element can be one of: url, header, cookie, filetype, statusline, text
>>Enter element to be modified: cookie
>>cookie Modification http-mod-list Rule 5
```

3. Enter **action** to access the *Rule Action* menu, and then enter **insert** to set the new rule insert action.

```
>>cookie Modification http-mod-list Rule 5 # action
>>Current rule action:
>>Enter new rule action [insert|replace|remove]: insert
>>Enter cookie key to insert:
>>Enter cookie value to insert:
>>Enter cookie path or none:
>>Enter cookie domain name or none:
>>Enter insert-cookie expiration as either :
>>... a date     <MM/dd/yy[@hh:mm]> (e.g. 12/31/01@23:59)
>>... a duration <days[:hours[:minutes]]> (e.g. 45:30:90)
>>... or none    <return>
>>Enter cookie expiration:
```

4. For the insert action, you can define a match criteria. If you define a match criteria, the insertion is performed only if the match is met.

   Enter the element to be matched for insertion. For more information, see the *Alteon Application Switch Operating System Command Reference*.

```
>>Element to match can be one of url, header, cookie, filetype, statusline,
>>text, regex, none
>>Enter element to match []:
```

5. Based on the selected match element, enter the required parameters.
6. Enter **directn** to set the rule direction, and then enter the rule direction: request or response.

```
>>cookie Modification http-mod-list Rule 5 # directn
>>Enter new rule direction [req:resp] [req]:
```

Examples

A To insert the Set-Cookie for a cookie named "Device-ID" with the value "Alteon123" to all server responses, use the following configuration:

```
>>HTTP Modification rule-list mylist# cur
Current rule: mylist enabled
  10:
    enabled
    action insert cookie
        KEY=Device-ID, VALUE=Alteon123
    direction response
```

B To insert the Set-Cookie for a cookie named "Device-ID" with the value "Alteon123" to server responses where a cookie named "GSLB" with the value "On" exists, use the following configuration:

```
>> HTTP Modification http-mod-list Rule 1# cur
Current rule: 1
    enabled, name My_list
    action insert cookie
        KEY=Device_ID, VALUE=Alteon123
        MATCH=cookie, KEY=GSLB, VALUE=On
    direction response
```

The header is only inserted if the response contains the header Set-Cookie: GSLB=On.

Configuring HTTP Modifications for the HTTP File Type

When creating a rule for an HTTP file type element, only the replace action can be defined. Only the request direction is supported.

In the response, the file type may appear in different locations. If such file type elements need to be modified, the modification depends on the location, as follows:

- HTTP Headers in the server response—Location and Content-Type
  - The **Content type** field indicates the media type of the entity-body sent to the recipient
  - The **Location** is used to redirect the recipient to a location other than the Request-URL for completion of the request
  - If you want to modify these headers, use HTTP modification for headers and specify header name as Location or Content-Type accordingly.
- Links that appear in the HTML within the server response—If you want to modify all file types of other objects referenced in the server's response (for example, links in the HTML), then use URL modification and select Header and Body.

To configure HTTP modification for the HTTP file type

Note: The numbers and names in this procedure are examples only.

1. Access HTTP Modification rule list configuration via the *Layer 7* menu, enter a rule list ID, and enable the rule list.

```
>>Main# /cfg/slb/layer7/httpmod
>>Enter HTTP Modification rule-list id (alphanumeric): http-mod-list
>>HTTP Modification rule-list http-mod-list# ena
```

2. Enter **rule**, the rule ID number, and then enter the desired element type.

```
>>HTTP Modification rule-list http-mod-list# rule
>>Enter HTTP Modification rule number (1-128):5
>>Element can be one of: url, header, cookie, filetype, statusline, text
>>Enter element to be modified: filetype
>>filetype Modification http-mod-list Rule 5
```

3. Enter **action** to access the *Rule Action* menu, and then enter **replace** to set the new rule replace action.

```
>>filetype Modification http-mod-list Rule 5 # action
>>Current rule action:
>>filetype supports only action replace
>>Enter file type to replace:
>>Enter new file type:
```

Example

To replace all requests for ".jpeg" files to use ".jpg", use the following configuration:

```
>> HTTP Modification http-mod-list Rule 2# cur
Current rule: 2
    enabled, name My_list
    action replace filetype
        from: FILETYPE=jpeg
        to: FILETYPE=jpg
    direction request
```

Configuring HTTP Modification for HTTP Status Line

The status line is a mandatory part of an HTTP response. A single status line must appear in every HTTP response. Therefore, the status line cannot be inserted or removed. The only supported modification for status line is replace.

For elements of the status line type, the direction is set to response and cannot be changed.

When creating a rule for a HTTP status line element, only the replace action can be defined.

To configure the replace action for the HTTP status line

Note: The numbers and names in this procedure are examples only.

1. Access HTTP Modification rule list configuration via the *Layer 7* menu, enter a rule list ID, and enable the rule list.

```
>>Main# /cfg/slb/layer7/httpmod
>>Enter HTTP Modification rule-list id (alphanumeric): http-mod-list
>>HTTP Modification rule-list http-mod-list# ena
```

2. Enter **rule**, the rule ID number, and then enter the desired element type.

```
>>HTTP Modification rule-list http-mod-list# rule
>>Enter HTTP Modification rule number (1-128):5
>>Element can be one of: url, header, cookie, filetype, statusline, text
>>Enter element to be modified: statusline
>>statusline Modification http-mod-list Rule 5
```

3. Enter **action** to access the *Rule Action* menu, and then enter **replace** to set the new rule replace action.

```
>>statusline Modification http-mod-list Rule 5 # action
>>Current rule action:
>>Enter status code to replace: 333
>>Enter status line to replace or none:
>>Enter new status code or none: 444
>>Enter new status line or none:
```

Example

To replace responses with status code of 333 to 444 with text of "status is 444", use the following configuration:

```
>> HTTP Modification http-mod-list Rule 1# cur
Current rule: 1
    enabled
    action replace statusline
        from: STATUSCODE=333
        to: STATUSCODE=444, STATUSLINE=status is 444
    direction response
```

If you do not set the new status line, the previous text remains.

Configuring HTTP Modification for URL Elements

Modification for URL element s lets you perform complex operations. You can set actions for the protocol (HTTP or HTTPS), port, host, path, page name and page type in one rule.

For example, when the URL is as HTTP://www.site.com/a/b/c/index.html, the following results:

— The protocol is HTTP
— The port is 80 (default for HTTP)
— The host is www.site.com
— The path is a/b/c
— The page name is index
— The page type is html

All the components within this URL can be modified using a single HTTP Modification URL rule.

The following topics are discussed in this section:

- Configuring Modification for HTTP URL Elements, page 257
- Example 1: Update the Path, page 259
- Example 2: Force links to sensitive information to use HTTPS, page 261
- Example 3: Update Host and Path, page 262

Configuring Modification for HTTP URL Elements

The following procedure provides general background and parameter-level explanation for modifying HTTP URL elements.

To use HTTP content modifications for URL elements

Note: The numbers and names in this procedure are examples only.

1. Access HTTP Modification rule list configuration via the *Layer 7* menu, enter a rule list ID, and enable the rule list.

```
>>HTTP Modification rule-list http-mod-list#
>>Enter HTTP Modification rule-list id (alphanumeric): http-mod-list
>>HTTP Modification rule-list http-mod-list# ena
```

2. Enter **rule**, the rule ID number, and then enter the desired element type.

```
>>HTTP Modification rule-list http-mod-list# rule
>>Enter HTTP Modification rule number (1-128):5
>>Element can be one of: url, header, cookie, filetype, statusline, text
>>Enter element to be modified: URL
>>URL Modification http-mod-list Rule 5
```

3. Enter directn to set the rule direction, and then enter the desired rule direction:
   - Request—Only client requests are inspected for modification.
   - Response—Only server responses are inspected for modification.
   - Bidirectional—The modification is done on server response and the reverse modification is done on the subsequent client request. For example, you can remove the complete path from the response so that the same path is added to the subsequent request.

```
>>URL Modification http-mod-list Rule 5 # directn
>>Enter new rule modification direction [req:resp:bidirectional] [req]:
```

4. Enter **body** to enable URL modification in the body.

```
>>URL Modification http-mod-list Rule 5 # body
>>Current rule body: exclude
>>Enter new rule body [include:exclude] [exclude]:
```

   By default, only headers are modified (body exclude). To modify both header and body, set to **body include**.

5. Enter **match** to access the *Match* menu and define the match criteria.

   Set the match parameters according to the configured rule direction: request or response. When the direction is set to bidirectional, set the match parameters to match the server response.

   You can set match criteria for the following:
   - Protocol—HTTP or HTTPS. The default value is HTTP.

— Port—The port used in the URL. The default value is 0, implying a match for cases when the port is not explicitly specified in the URL. This means the default port for the specified protocol (80 for HTTP, 443 for HTTPS) is used. Another example is when the default port appears explicitly in the URL.

When the port is 0 for both match and action, this implies that the port parameter is not checked (the rule is matched regardless of the port that is used in the URL) and not changed.

— Host

- Host Match Type can be set to Suffix, Prefix, Equal, Include or Any. Any implies that any host will match.
- Host to Match indicates the value to be used for the match. This parameter is not required when Match Type is set to Any.
  For example: Host Match Type prefix and Host to Match www.a will match all hosts that start with **www.a**, such as **www.a.com**, **www.abc.com**, and so on.

— Path

- Path Match Type can be set to Suffix, Prefix, Equal, Include or Any. This parameter is not required when Match Type is set to Any. Any implies that any non-empty path will match.
- Path to Match indicates the value to be used for the match.

This parameter is not required when the Match Type is set to Any.
For example: Path Match Type **include**, and Path to Match **abc** match any path that has **abc** in it, such as **/abc/**, **/a/abc**, and so on.

— Page Name—Used for an exact match.

— Page Type—Used for an exact match.

Note: An AND operation is used between the configured match criteria. Therefore, only when all the configured match criteria are met in the request (or response), the action is performed.

6. Enter **action** to access the *Rule Action* menu, and define the action criteria.

   You can set actions for the following parameters:

— Protocol—HTTP or HTTPS. The default value is HTTP.

— Port—The port to be set in the URL. The default value is 0, which means:

- When the match port is not 0, the port is removed from the URL.
- When the port parameter is 0 for both match and action, the port in the URL remains unchanged. That is, if it was explicitly specified it remains as it is, if it was not specified it remains so.

— Host—The Host Action Type can be set to Insert, Replace, or Remove.

- Insert— Lets you insert additional text to the hostname, either before or after the matched text.
- Replace—Lets you replace the matched text in the hostname with another text.
- Remove—Lets you remove the matched text from the hostname.
- None—No action is taken.

Replace and Remove are not allowed when the Host Match Type is set to **Any**.

When a host match is set, an action must be specified. To leave the same host, use **action replace** with the same text string used in the match.

For example: Host Match Type prefix and Host to Match **www.a** match all hosts that start with **www.a**. Using Host Action Insert After with Host to Insert **bbb** results in the following: host **www.a.com** is modified to **www.abbb.com**. Host **www.az.com** is modified to **www.abbbz.com**.

— Path—Path Action Type can be set to Insert, Replace, or Remove.
  - Insert—Lets you insert additional text to the path, either before or after the matched text.
  - Replace—Lets you replace the matched text in the path with another text.
  - Remove—Lets you remove the matched text from the path.
  - None—No action is taken.

  Replace and Remove are not allowed when the Path Match Type is set to **Any**.

  When using a path match, an action must be specified. To use path match as match criteria only and leave the same path, use the replace action with the same text string used in the match.

  For example: Path Match Type include, and Path to Match **abc** match any path that contains **abc**, such as **/abc/**, **/a/abc**, and so on. Using Path Action Remove results in the following: path **abc** is removed, path **de/abc/xyz** is modified to **de/xyz**.

— Page Name—A new page name. Leave this action empty to remove the matched page name. When both match and action are empty, no operation is performed.

— Page Type— A new page type. Leave this action empty to remove the matched page type. When both match and action are empty, no operation is performed.

Example 1: Update the Path

The web site links should be updated as follows:

Every link that ends with **cars** should now be updated to end with **new-cars**. For example, the URL **HTTP://www.site.com/vehicles/offer-cars/details.html** should now be **HTTP://www.site.com/vehicles/offer-new-cars/details.html**.

Note: The numbers and names in this procedure are examples only.

1. Configure the required real servers, group, virtual server and service. The service is HTTP or HTTPS, according to the site. Associate an HTTP modification policy to achieve the HTTPS link updates.

```
>> HTTP Load Balancing Menu # httpmod
Current HTTP modifications rule-list:
Enter new HTTP modifications rule-list or none: add-new
>>For HTTP Modification rule-list configuration use /cfg/slb/layer7/httpmod
```

2. Create the HTTP modifications rule list:

```
>> Main # /cfg/slb/layer7/httpmod
>> Enter HTTP Modification rule-list id (alphanumeric): add-new
```

3. One rule is required. In this example, Rule 10 is added:

```
>>URL Modification rule-list add-new#
>>Enter HTTP Modification rule number (1-128): 10
>>Element can be one of: url, header, cookie, filetype, statusline, text
>>Enter element to be modified: URL
>>URL Modification add-new Rule 10#
```

4. It is required to modify URLs in the body of the response, so set the body to **include**.

```
>>URL Modification add-new Rule 10#body
Current rule body: exclude
Enter new rule body [include|exclude] [exclude]:include
>>URL Modification add-new Rule 10#
```

5. Set match criteria:

```
>>URL Modification add-new Rule 10#match
>>URL Match#path
Current path match configuration:
Enter path match-type [sufx|prefx|eq|incl|any] [any]:sufx
Enter path to match:cars
```

6. Set the required action. Path match was set, so an action also must be specified. In order not to change the path, use replace with the same path string.

```
>>URL Modification add-new Rule 10#action
>>URL Match#path
Current path action configuration: none
Enter path action-type [insert|replace|remove|none] [none]: insert
Enter path to insert: new-
Insert the specified path before or after the matched section? [b|a]: b
```

7. Enable the rule and the rule list.

```
>>URL Modification add-new Rule 10#ena
>>URL Modification add-new Rule 10#..
>>URL Modification rule-list add-new#ena
```

8. Apply and save. You can use **cur** to see the complete rule list configuration:

```
>>HTTP Modification rule-list add-new# apply
>>HTTP Modification rule-list add-new# save
>> HTTP Modification rule-list add-new# cur
Current Httpmod Rule-List add-new:
    enabled
    Rules:
        1: enabled
           element url
                      match:
                          protocol http, port 0
                          path suffix cars
                      action:
                          protocol http, port 0
                          path insert new- before
                      direction response
                      body include
```

Example 2: Force links to sensitive information to use HTTPS

A Web site includes sensitive information. However, the links in the Web site were not designed to use HTTPS for the sensitive information, and so some links refer to HTTP.

Alteon needs to modify URLs that appear in the response, where the path includes "/sensitive/", to use HTTPS rather than HTTP.

Note: The numbers and names in this procedure are examples only.

1. Configure the required real servers, group, virtual server and service. The service is HTTP or HTTPS, according to the site. Associate an HTTP Modification Policy to achieve the HTTPS links.

```
>> HTTP Load Balancing Menu # httpmod
Current HTTP modifications rule-list:
Enter new HTTP modifications rule-list or none: force-https
For HTTP Modification rule-list configuration use /cfg/slb/layer7/httpmod
```

2. Create the HTTP modifications rule list:

```
>> Main # /cfg/slb/layer7/httpmod
Enter HTTP Modification rule-list id (alphanumeric): force-https
```

3. One rule is required. In this example, Rule 10 is added:

```
>>URL Modification rule-list force-https#
>>Enter HTTP Modification rule number (1-128): 10
>>Element can be one of: url, header, cookie, filetype, statusline, text
>>Enter element to be modified: URL
>>URL Modification force-https Rule 10#
```

4. It is required to modify URLs in the body of the response, so set the body to **include**.

```
>>URL Modification force-https Rule 10#body
Current rule body: exclude
Enter new rule body [include|exclude] [exclude]:include
>>URL Modification force-https Rule 10#
```

5. Set the match criteria:

```
>>URL Modification force-https Rule 10#match
>>URL Match#protocol http
>>URL Match#path
Current path match configuration:
Enter path match-type [sufx|prefx|eq|incl|any] [any]:incl
Enter path to match:/sensitive/
```

6. Set the required action. Since a path match was set, an action also must be specified. To leave the path unchanged, use replace with the same path string.

```
>>URL Modification force-https Rule 10#action
>>URL Match#protocol https
>>URL Match#path
Current path action configuration: none
Enter path action-type [insert|replace|remove|none] [none]: replace
Enter new path to replace the matched section: /sensitive/
```

7. Enable the rule and the rule list.

```
>>URL Modification force-https Rule 10#ena
>>URL Modification force-https Rule 10#..
>>URL Modification rule-list force-https#ena
```

8. Apply and save. In addition, you can use **cur** to see the complete rule list configuration:

```
>>URL Modification rule-list force-https# apply
>>URL Modification rule-list force-https# save
>>URL Modification rule-list force-https# cur
Current rule-list: force-https enabled
  10:
    enabled
    element url
    match:
        protocol http, port 80
        path incl /sensitive/
    action:
        protocol https, port 443
        path replace /sensitive/
    direction response
    body include
```

Example 3: Update Host and Path

All links to **HTTP://www.site2.com/anypath** should be updated to point to **HTTP://www.site1.com/site2/anypath**.

1. Configure the required real servers, group, virtual server and HTTP service. Associate an HTTP modification policy to achieve the HTTPS links.

```
>> HTTP Load Balancing Menu # httpmod
Current HTTP modifications rule-list:
Enter new HTTP modifications rule-list or none: move-site2
For HTTP Modification rule-list configuration use /cfg/slb/layer7/httpmod
```

2. Create the HTTP modifications rule list.

```
>> Main # /cfg/slb/layer7/httpmod
Enter HTTP Modification rule-list id (alphanumeric): move-site2
```

3. One rule is required. In this example, Rule 20 is added:

```
>>URL Modification rule-list move-site2#
>>Enter HTTP Modification rule number (1-128): 20
>>Element can be one of: url, header, cookie, filetype, statusline, text
>>Enter element to be modified: URL
>>URL Modification move-site2 Rule 20#
```

4. It is required to modify URLs in the body of the response, so set the body to **include**

```
>>URL Modification move-site2 Rule 20#body
Current rule body: exclude
Enter new rule body [include|exclude] [exclude]:include
>>URL Modification move-site2 Rule 20#
```

5. Set the match criteria:

```
>>URL Modification move-site2 Rule 20#match
>>URL Match#host
Current host match configuration:
Enter host match-type [sufx|prefx|eq|incl|any] [any]:eq
Enter host to match: www.site2.com
```

6. Set the required action.

```
>>URL Modification move-site2 Rule 20#action
>>URL Match#host
Current host action configuration: none
Enter host action-type [insert|replace|remove|none] [none]: replace
Enter new path to replace the matched section: www.site1.com
>>URL Match#path
Current path action configuration: none
Enter path action-type [insert|replace|remove|none] [none]: insert
Enter path to insert: site2/
Insert the specified path before or after the matched section? [b|a]: b
```

7. Enable the rule and the rule list.

```
>>URL Modification move-site2 Rule 20#ena
>>URL Modification move-site2 Rule 20#..
>> HTTP Modification rule-list force-https#ena
```

8. Apply and save. In addition, you can use **cur** to see the complete rule list configuration:

```
>>URL Modification rule-list move-site2# apply
>>URL Modification rule-list move-site2# save
>>URL Modification rule-list move-site2# cur
Current rule-list: move-site2 enabled
  20:
    enabled
    element url
    match:
        protocol http, port 80
        host eq www.site2.com
        path any
    action:
        protocol http, port 80
        host replace www.site1.com
        path insert site2/ before
```

Note: The current rule matches any link that includes any path at **www.site2.com**. To modify the URL HTTP://www.site2.com itself (with no path), a different rule is required. The path match is set to equal empty (leave the value empty), so that the rule list looks as follows:

```
>>URL Modification rule-list move-site2# cur
Current rule-list: move-site2 enabled
  20:
    enabled
    element url
    match:
        protocol http, port 80
        host eq www.site2.com
        path any
    action:
        protocol http, port 80
        host replace www.site1.com
        path insert site2/ before
    direction response
    body include
  30:
    enabled
    element url
    match:
        protocol http, port 80
        host eq www.site2.com
        path eq
    action:
        protocol http, port 80
        host replace www.site1.com
        path insert site2/ before
    direction response
    body include
```

Configuring HTTP Modification for Text Elements

When configuring actions for text elements, these modifications are applied to the header only (default), or to both the header and body, of the HTTP responses or requests.

When creating a rule for a HTTP text element, the following actions can be defined:

- To configure the replace action for an HTTP text element, page 265
- To configure the remove action for the HTTP text element, page 266

To configure the replace action for an HTTP text element

This action replaces the matched string with the new text specified.

Note: The numbers and names in this procedure are examples only.

1. Access HTTP Modification rule list configuration via the *Layer 7* menu, enter a rule list ID, and enable the rule list.

```
>>Main# /cfg/slb/layer7/httpmod
>>Enter HTTP Modification rule-list id (alphanumeric): http-mod-list
>>HTTP Modification rule-list http-mod-list# ena
```

2. Enter **rule**, the rule ID number, and then enter the desired element type.

```
>>HTTP Modification rule-list http-mod-list# rule
>>Enter HTTP Modification rule number (1-128):5
>>Element can be one of: url, header, cookie, filetype, statusline, text
>>Enter element to be modified: text
>>text Modification http-mod-list Rule 5
```

3. Enter **action** to access the *Rule Action* menu, and then enter **replace** to set the new rule replace action.

```
>>text Modification http-mod-list Rule 5 # action
>>Enter rule action [replace:remove]: replace
>>Enter text to replace: Copyright 2013
>>Enter new text: All rights reserved
```

4. Enter **directn** to set the rule direction, and then enter the desired rule direction.

```
>>text Modification http-mod-list Rule 5 # directn
>>Enter new rule modification direction [req:resp] [req]: resp
```

5. Enter **body** to enable text modification in the body.

```
>>text Modification http-mod-list Rule 5 # body
>>Current rule body: exclude
>>Enter new rule body [include:exclude] [exclude]: include
```

Example

To replace responses that include the text "Copyright 2013" to "All rights reserved", use the following configuration:

```
>>URL Modification rule-list mylist# cur
Current rule-list: mylist enabled
  10:
    enabled
    action replace text
from: TEXT=Copyright 2013
to: TEXT=All rights reserved
    direction response
    body include
```

To configure the remove action for the HTTP text element

With this action, the string matching the condition is removed.

Note: The numbers and names in this procedure are examples only.

1. Access HTTP Modification rule list configuration via the *Layer 7* menu, enter a rule list ID, and enable the rule list.

```
>>Main# /cfg/slb/layer7/httpmod
>>Enter HTTP Modification rule-list id (alphanumeric): http-mod-list
>>HTTP Modification rule-list http-mod-list# ena
```

2. Enter **rule**, the rule ID number, and then enter the desired element type.

```
>>HTTP Modification rule-list http-mod-list# rule
>>Enter HTTP Modification rule number (1-128):5
>>Element can be one of: url, header, cookie, filetype, statusline, text
>>Enter element to be modified: text
>>text Modification http-mod-list Rule 5
```

3. Enter **action** to access the *Rule Action* menu, and then enter **remove** to set the new rule remove action.

```
>>text Modification http-mod-list Rule 5 # action
>>Enter rule action [replace:remove]: remove
>>Enter text to remove: test test test
```

4. Enter **directn** to set the rule direction, and then enter the desired rule direction.

```
>>text Modification http-mod-list Rule 5 # directn
>>Enter new rule modification direction [req:resp] [req]:
```

5. Enter **body** to enable text modification in the body.

```
>>text Modification http-mod-list Rule 5 # body
>>Current rule body: exclude
>>Enter new rule body [include:exclude] [exclude]:
```

Example

To remove the text "test test test" wherever it appears in the response, use the following configuration:

```
>>URL Modification rule-list mylist# cur
Current rule-list: mylist enabled
  10:
    enabled
    action remove text
TEXT=test test test
    direction response
    body include
```

Associating HTTP Modification Rules to a Service

After defining HTTP modification rule lists, you can associate them to one or multiple services. The following procedure applies to all types of elements.

To associate HTTP modification rules to a service

Access the desired service and enable the desired rule list for the selected service.

```
>>Main# /cfg/slb/virt 1/service 80/http/httpmod
>>Enter new HTTP Modification rule list or none:
```

Content-Intelligent Caching and Compression Overview (FastView™)

Application acceleration helps to speed up the performance of Web applications for remote employees, customers or partners who access these applications over a network.

An Application Delivery Controller (ADC) that accelerates Web traffic addresses the two main factors that impede performance: latency (the time delay between two computers communicating with each other over a network), and bandwidth (the amount of network capacity available to applications) using the following techniques:

- **Content caching**—This technique stores data that is likely to be used again and is unlikely to change, instead of requiring servers to retrieve or generate it every time. For more details, see Content-Intelligent Caching, page 268.
- **Compression**—This technique reduces the amount of data crossing the link (squeezing it into smaller amounts) making it faster and more efficient to send across a network. For more details, see Content-Intelligent Compression, page 272.
- **Connection management**—Connection management uses the optimizations to the standard TCP protocol to gain better performance of transporting the data over the network and multiplexing of HTTP requests from multiple clients over a much smaller number of server connections. For more information about the specific TCP optimization see Content-Intelligent Connection Management, page 277.

Content-Intelligent Caching

Web pages are composed of a series of objects. Many of these objects are static objects that are used repeatedly from page to page. Alteon caching can recognize requests for such objects and retrieve them directly from Alteon's local cache without fetching them from the Web server. This relieves the server of dealing with repetitive requests for the same content and at the same time accelerates objects delivery to the end-user.

Alteon caching support is compliant with RFC 2616 of HTTP 1.1. It respects relevant HTTP headers (such as Cache-control, Expires, Authorization, Pragma, and so on) which are the Web Application means of dictating which content is to be cached and when it should be refreshed.

Alteon caching has options to determine its cache behavior, both in terms of which content to cache, and in terms of which content to serve to clients from cache. Caching support includes the option to define per-URL caching behavior, cache expiration time, and includes an option to optimize a client browser's caching to improve response time and Quality of Experience (QoE).

Alteon caching is based on Alteon's RAM to ensure fast retrieval of content and delivery to clients. You can configure the amount of RAM dedicated for the caching Web object. However, the more cache space you allocate, the fewer the number of concurrent connections that can be handled by Alteon.

Caching occurs at the client side of the flow. This means that when a request comes, it is considered higher priority for serving from cache before all other application services (for example, HTTP modifications). On the other hand, when a server response arrives at the Application Services Engine, it goes through all required treatments, such as compression and HTTP modification, before being cached. Therefore the next serving of that response from cache also includes them.

Caching configuration includes a FastView policy and a cache URL exceptions rule list that is optionally associated to that policy. FastView policies are, in turn, associated with an HTTP virtual service.

This section describes the following procedures:

- Configuring the Caching Virtual Service, page 268
- Configuring the FastView Policy, page 269

Configuring the Caching Virtual Service

For Alteon to perform caching, you must define an HTTP virtual service and associate a FastView policy to it. As with other Alteon capabilities, the virtual service is assigned to an application, in this case HTTP, or HTTPS with SSL offloading.

To associate a caching policy to a virtual service

1. Access the *Virtual Server Service* menu for the virtual service to which you want to associate the FastView policy. In the following example, Virtual Server 1 is associated with the HTTP application.

   Note: When indicating the virtual service, you can use either the virtual port number or a name. In this example, instead of the HTTP, you can enter 80 for the standard HTTP port number.

```
>> Main# /cfg/slb/virt 1/service 80/http/cachepol
```

2. Enter a new FastView policy name, or one that already exists.

```
Current FastView policy name:
Enter new FastView policy name or none: Caching1
```

The FastView policy name you entered is now associated with virtual service HTTP.

3. Configure the FastView policy, as described in Configuring the FastView Policy, page 269.

Configuring the FastView Policy

The FastView policy defines the caching behavior required for the virtual service. A single FastView policy can be associated to multiple virtual services if they share the same caching configuration. Caching parameters include:

- Policy name
- Maximum expiration time
- Minimum object size to be stored
- Maximum object size to be stored
- Cache URL exceptions rule list
- Behavior for storing new object in cache
- Behavior when serving client with object
- Inclusion of query parameter
- Enable or disable optimize browser cache

For details on configuring the FastView policy parameters, see the section on the **/cfg/slb/accel/fastview/cachepol** menu in the *Alteon Application Switch Operating System Command Reference*.

Cache Content Management

You can manage the content of the cache using Alteon configuration or Alteon operations.

- Alteon configuration—Use Caching rule lists (see Caching Rule Lists, page 270) to define which objects do not go into the cache.
- Alteon operations—Use a cache purge (see Purging Cached Content, page 270) to specify services and virtual service, and URLs including a wildcard (*). The cahe purge is an operational command. The object is removed immediately from the cache, but it may be cached again later.

Alteon automatically removes from its cache objects that have been changed by users. HTTP POST, PUT, or DELETE requests for an object clear that object from the cache, in accordance with RFC 2616.

The exact behavior is determined by the configuration of the Query parameter in the FastView policy, as follows:

- When query is set to *ignore*—Objects matching the URL only, not including the query parameters, are removed from cache.
- When query is set to *consider*—Objects matching the URL, including its query parameters, are removed from the cache.

This section describes the following procedures:

- Caching Rule Lists, page 270
- Purging Cached Content, page 270
- Common FastView Policy Use Cases, page 270

Caching Rule Lists

Associating caching rule lists to a FastView policy enables you to skip caching certain types of traffic that either require too many resources or provide little benefit in caching them.

A rule list is an ordered list of rules that specifies which URLs to cache or not cache. You can create multiple rule lists and change the lists associated with a FastView policy as needed.

Rule list logic is first-match, meaning once a rule within the list is matched, the remaining rules in the list are not evaluated. You can duplicate an entire rule list using the Copy Rule-List option.

Rules are ordered in the rule list according to their index number. Radware recommends putting rules that are matched often at the top of the list to optimize performance. See the cache URL rule list statistics per rule to determine how often rules are matched.

Notes

- When the URL ends with an asterisk (*) it is interpreted as a wildcard, and causes the entire objects "tree" under the specified URL to be invalidated. The wildcard is interpreted in a wide sense; meaning anything that appears in URL after that point is removed including multiple page instances differentiated by query parameters.
- If the URL includes a page name and/or page suffix and then an asterisk (for example, http://mycompany.com/path/page.type*), only various instances of the specific page with different query parameters (specified after the question mark sign) are removed.

Purging Cached Content

In some cases you may want to purge the cached content of HTTP responses. The cache is purged for the specified virtual server and virtual service. For a granular purge, you can also specify the object URL, including a wildcard (*).

Notes

- When the URL ends with an asterisk (*) it is interpreted as a wildcard, and causes the entire objects "tree" under the specified URL to be invalidated. The wildcard is interpreted in a wide sense; meaning anything that appears in URL after that point is removed including multiple page instances differentiated by query parameters.
- If the URL includes a page name and/or page suffix and then an asterisk (for example, http://mycompany.com/path/page.type*), only various instances of the specific page with different query parameters (specified after the question mark sign) are removed.

For more information, see the section on the **`/oper/slb/cachpurg`** command in the *Alteon Application Switch Operating System Command Reference*.

Common FastView Policy Use Cases

Example 1: Configuring a Basic FastView Service

1. Before you can configure a caching service, ensure that Alteon is configured for basic SLB:
   — Define an IP interface.
   — Enable SLB.
   — Assign an IP address to each of the real servers in the server pool.
   — Define each real server.
   — Assign servers to real server groups.

- Define server port and client port.
- Define virtual server

For more information on how to configure your network for SLB, see Server Load Balancing, page 165.

2. Define the caching policy which will govern the caching behavior, as follows:

`>> Main# /cfg/slb/accel/fastview/fastpol myPol`	(Define an ID to identify the FastView policy)
`>> FastView Policy myPol# ena`	(Enable the policy)

For details on defining additional caching policy parameters, see the section on the `/cfg/slb/accel/fastview/fastpol/caching` menu in the *Alteon Application Switch Operating System Command Reference*.

3. Globally enable FastView.

```
>> Main# /cfg/slb/accel/fastview/on
```

4. Set the HTTP virtual service to be used in the defined virtual server.

`>> Main# /cfg/slb/virt 1/service 80/http`	(Define HTTP service)
`>> HTTP Load Balancing# fastpol myPol`	(Associate the defined FastView policy)

5. Enable DAM or configure proxy IP addresses and enable proxy on the client port.

Example 2: Configuring a FastView Service with a Caching Exception Rule List

1. Before you can configure a FastView service, ensure that Alteon is configured for basic SLB:
   - Define an IP interface.
   - Enable SLB.
   - Assign an IP address to each of the real servers in the server pool.
   - Define each real server.
   - Assign servers to real server groups.
   - Define server port and client port.
   - Define virtual server

   For more information on how to configure your network for SLB, see Server Load Balancing, page 165.

2. Define the FastView policy which will govern the caching behavior, as follows:

`>> Main# /cfg/slb/accel/fastview/fastpol myPol`	(Define an ID to identify the FastView policy)
`>> FastView Policy myPol# urllist myurllist`	(Associate the Cache rule list name myurllist)
`>> FastView Policy myPol# ena`	(Enable the policy)

For details on defining additional caching policy parameters, see the section on the **`/cfg/slb/accel/fastview/fastpol/caching`** menu in the *Alteon Application Switch Operating System Command Reference*.

3. Define a caching rule list.

`>> Main# /cfg/slb/accel/fastview/cachlist myurllist`	(Define an ID to identify the Caching rule list)
`>> Rule-List myurllist#`	
`>> Rule-List myurllist# ena`	(Enable the Caching List)

4. Globally enable FastView.

```
>> Main# /cfg/slb/accel/fastview/on
```

5. Set the HTTP virtual service to used in the defined virtual server.

`>> Main# /cfg/slb/virt 1/service 80/http`	(Define HTTP service)
`>> HTTP Load Balancing# fastpol myPol`	(Associate the defined FastView policy)

6. Enable DAM or configure proxy IP addresses and enable proxy on the client port.

Content-Intelligent Compression

HTTP compression is built into Web servers and Web clients to make better use of available bandwidth, and provide faster perceivable transmission speeds between both, as less data is actually transferred. HTTP data is compressed before it is sent from the server as follows:

- Compliant browsers announce what methods are supported to the server before requesting each object. Commonly supported methods are the gzip and Deflate compression algorithms.
- Browsers that do not support compliant compression method download uncompressed data.

Alteon compression can ensure optimal application delivery and bandwidth savings through compression of Web pages such as HTML and JavaScript in real-time before transmission on the network. This is important especially for small remote offices and home office users where bandwidth may be limited. This dynamic HTML compression accelerates traffic by reducing the payload using an open compression standard (gzip and Deflate), providing a powerful performance boost. The support of the industry-standard gzip algorithm (as well the Deflate algorithm) ensures compatibility with virtually all popular Web browsers without requiring any special software installation on the end-user computer.

Alteon HTTP compression includes options to control compression behavior. These include the ability to define whether objects should be compressed for browser, content-type or URL specific behavior, as well as a set of predefined exceptions of the default compression behavior based on known browser limitations.

Compression configuration includes an compression policy and two types of compression rule lists (URL exceptions and browser exceptions) that are optionally associated to the policy. Compression policies are, in turn, associated with an HTTP virtual service.

This section describes the following procedures:

- Configuring the Compression Virtual Service, page 272
- Compression Policy, page 273
- Compression Exceptions Rule Lists, page 273
- Common Compression Policy Use Cases, page 274

Configuring the Compression Virtual Service

For Alteon to perform compression, you must define an HTTP virtual service and associate a compression policy to it. As with other Alteon capabilities, the virtual service is assigned to an application, in this case HTTP or HTTPS. HTTP is the only supported application type and is the only protocol that supports compression inherently.

To associate a compression policy to a virtual service

1. Access the *Virtual Server Service* menu for the virtual service to which you want to associate a compression policy. In the following example, Virtual Server 1 is associated with the HTTP application.

   Note: When indicating the virtual service, you can use either the virtual port number or a name. In this example, instead of the HTTP, you can enter 80 for the standard HTTP port number.

```
>> Main# /cfg/slb/virt 1/service 80/http/comppol
```

2. Enter a new compression policy name, or one that already exists.

```
Current compression policy:
Enter new compression policy or none: mycompression
```

   The compression policy name you entered is now associated with virtual service HTTP.

3. To configure the compression policy, see the section on the **`/cfg/slb/accel/compress`** menu in the *Alteon Application Switch Operating System Command Reference*.

Compression Policy

The compression policy defines the compression behavior required for the virtual service. A single compression policy can be associated to multiple virtual services if they share the same compression configuration. Compression parameters include:

- Policy name
- Compression algorithm
- Compression level
- Minimum file size to be compressed
- Maximum file size to be compressed
- Compression URL exceptions rule list
- Compression browser exceptions rule list
- Predefined browser exceptions rule list
- Compression by real server

For details on configuring the compression policy parameters, see the section on the **`/cfg/slb/accel/compress`** menu in the *Alteon Application Switch Operating System Command Reference*.

Compression Exceptions Rule Lists

Associating exceptions rule lists to a compression policy enables you to skip compressing certain types of traffic that either require too many resources or provide little benefit in compressing them.

A rule list is an ordered list of rules that specifies which URLs to compress or not compress. You can create multiple rule lists and change the lists associated with a compression policy as needed.

Rule list logic is first-match, meaning once a rule within the list is matched, the remaining rules in the list are not evaluated. You can duplicate an entire rule list using the Copy Rule-List option.

Rules are ordered in the rule list according to their index number. Radware recommends putting rules that are matched often at the top of the list to optimize performance. See the compression URL rule list statistics per rule to determine which rules are matched more or less often.

The following are the types of compression rule lists you can associate with a compression policy:

- **URL Exceptions Rule List**—This is a list of compression exceptions rules based on an object's URL (file/folder). These rules are the primary filter for evaluating exceptions. Browser exception and browser limitation rules are only evaluated after the URL exceptions.

  For example, the following rules result in all files in images folder being compressed except for **image1.jpg**:

  rule1: /images/image1.jpg, do not compress

  rule2: /images/, compress
- **Browser Exceptions Rule List**—This is a list of compression exception rules based on user-agent (browser type) and/or content-type (file type). These rules skip the compression of certain objects that create issues when uncompressed or that require too many resources with little benefit (for example, PDFs and PPT files). Browser exception rules are evaluated after the URL exception rules are evaluated, so they are more general than the URL exceptions.

  For example, the following rules result in all files in images folder being compressed except for **image1.jpg**:

  rule1: /images/image1.jpg, do not compress

  rule2: /images/,compress
- **Predefined Browser Exceptions Rule List**—This is a list of compression browser exception rules that address known issues in commonly used browsers which cause them to mishandle specific types of compressed content. The predefined browser limitation rule list cannot be modified or deleted. In order to customize it, you should first copy the rule list to a new browser exceptions rule list. This exception list is evaluated last, after the URL exception and browser exception lists, and therefore can be overridden by both the user-defined browser exception rule list and the URL rule list.

When there are both URL exception rule lists and browser exception rule lists associated with a compression policy, compression occurs only if both rule lists result in no exceptions.

Common Compression Policy Use Cases

Example 1: Configuring a Basic Compression Service

1. Before you can configure a compression service, ensure that Alteon is configured for basic SLB:
   — Define an IP interface.
   — Enable SLB.
   — Assign an IP address to each of the real servers in the server pool.
   — Define each real server.
   — Assign servers to real server groups.
   — Define server port and client port.
   — Define virtual server

   For more information on how to configure your network for SLB, see Server Load Balancing, page 165.
2. Define the compression policy which will govern the compression behavior:

`>> Main# /cfg/slb/accel/compress/comppol myPol`	(Define an ID to identify the compression policy)
`>> Compression Policy myPol#`	
`>> Compression Policy myPol# ena`	(Enable the policy)

For details on defining additional compression policy parameters, see the section on the **`/cfg/slb/accel/compress/comppol`** menu in the *Alteon Application Switch Operating System Command Reference*.

3. Globally enable compression.

```
>> Main# /cfg/slb/accel/compress/on
```

4. Set the HTTP virtual service to used in the defined virtual server.

`>> Main# /cfg/slb/virt 1/service 80/http`	(Define HTTP service)
`>> HTTP Load Balancing# comppol myPol`	(Associate the defined compression policy)

5. Enable DAM or configure proxy IP addresses and enable proxy on the client port.

Example 2: Configuring a Compression Service with a Compression URL Exception Rule List

1. Before you can configure a compression service, ensure that Alteon is configured for basic SLB:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface.
   - Define each real server.
   - Assign servers to real server groups.
   - Enable SLB.
   - Define server port and client port.
   - Define virtual server

   For more information on how to configure your network for SLB, see Server Load Balancing, page 165.
2. Define the compression policy which will govern the compression behavior:

`>> Main# /cfg/slb/accel/compress/comppol myPol`	(Define an ID to identify the compression policy)
`>> Compression Policy myPol# urllist myurllist`	(Associate a URL Rule List)
`>> Compression Policy myPol# ena`	(Enable the policy)

For details on defining additional compression policy parameters, see the section on the **`/cfg/slb/accel/compress/comppol`** menu in the *Alteon Application Switch Operating System Command Reference*.

3. Define a compression URL exception rule list.

`>> Main# /cfg/slb/accel/compress/urllist myurllist`	(Define an alphanumeric ID to identify the URL exception rule list)
`>> Compression URL Rule-List myurllist#`	(Add a rule to the rule list)
`>> Compression URL Rule-List myurllist# ena`	(Enable the URL List)

4. Globally enable compression.

```
>> Main# /cfg/slb/accel/compress/on
```

5. Set the HTTP virtual service to used in the defined virtual server.

`>> Main# /cfg/slb/virt 1/service 80/http/comppol`	(Define HTTP service)
`>> HTTP Load Balancing# comppol myPol`	(Associate the defined compression policy)

6. Enable DAM or configure proxy IP addresses and enable proxy on the client port.

Example 3: Configuring a Compression Service with a Compression Browser Exception Rule List

1. Before you can configure a compression service, ensure that Alteon is configured for basic SLB:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface.
   - Define each real server.
   - Assign servers to real server groups.
   - Enable SLB.
   - Define server port and client port.
   - Define virtual server

   For more information on how to configure your network for SLB, see Server Load Balancing, page 165.
2. Define the compression policy which will govern the compression behavior:

`>> Main# /cfg/slb/accel/compress/comppol myPol`	(Define an alphanumeric ID to identify the compression policy)
`>> Compression Policy myPol# brwslist mybrwslist`	(Associate a browser rule list)
`>> Compression Policy myPol# ena`	(Enable the policy)

   For details on defining additional compression policy parameters, see the section on the **`/cfg/slb/accel/compress/comppol`** menu in the *Alteon Application Switch Operating System Command Reference*.
3. Define a compression browser exception rule list.

`>> Main# /cfg/slb/accel/compress/brwslist mybrwslist`	(Define an alphanumeric ID to identify the URL exception rule list)

`>> Compression Browser Rule-List mybrwslist#`	(Add a rule to the rule list)
`>> Main# /cfg/slb/accel/compress/brwslist# ena`	(Enable the browser List)

4. Globally enable compression.

```
>> Main# /cfg/slb/accel/compress/on
```

5. Set the HTTP virtual service to used in the defined virtual server.

`>> Main# /cfg/slb/virt 1/service 80/http/comppol`	(Define HTTP service)
`>> HTTP Load Balancing# comppol myPol`	(Associate the defined compression policy)

TCP Congestion Avoidance

Alteon uses an improved TCP congestion avoidance algorithm for maximum throughput. The algorithm automatically adjusts the TCP congestion window size according to media and congestion conditions, so no configuration is required.

FastView Licensing

FastView licenses are not available for Alteon version 29.

Radware's FastView Advanced Web Performance Optimization solution is available as a standalone solution. For more information, see www.radware.com/Solutions/Enterprise/ApplicationNetworking/ApplicationAcceleration.aspx.

Content-Intelligent Connection Management

Alteon supports connection management, which multiplexes client and server connections and improves the throughput of SLB. It also helps the real server lower the need of establishing and tearing down TCP connections.

Since Alteon acts as a client for the back-end servers, Alteon always tries to reuse previously established SSL sessions. The SSL session reuse attempts are usually successful because the back-end server recognizes Alteon as a client that connects repeatedly. SSL session re-use between Alteon and the back-end servers helps lower the overhead involved in performing a full SSL handshake.

In a connection managed environment, a pool of server connections is maintained for servicing client connections. When a client sends an HTTP request, a server-side connection is selected from the server pool and used to service the request. When the client request is complete, the server connection is returned to the pool and the client connection dropped.

This feature only supports the HTTP and HTTPS protocols over TCP, and can work in conjunction with SSL, caching, and compression. When used with back-end SSL (where SSL is used between Alteon and the servers), it also reduces load on servers because fewer SSL handshakes are needed to be performed by them.

The following example enables connection management for the HTTP and HTTPS protocol on virtual Server 1:

```
>> Main# /cfg/slb/virt 1/service 80/http/connmgt
Current Connection management configuration: disabled
Enter new Connection management configuration [enabled|disabled|pooling] [d]:
ena
Enter server side connection idle timeout in minutes [0-32768] [10]:
Note: PIP must be set when connection management is enabled. It is recommended
to use egress PIP.
```

Connection management statistics can be displayed by issuing the following command:

```
>> Main# /stats/slb/http/connmng
```

Note: You must configure the Proxy IP (PIP) addresses to be used as source IP addresses for the server-side connections. Radware recommends using egress PIP, to ensure PIP is used only to the required servers and service. When using ingress PIP, all traffic coming via the specified port uses PIP, including traffic to other services.

Chapter 13 – Load Balancing Special Services

This chapter discusses Server Load Balancing (SLB) based on special services, such as HTTP, HTTPS, SSL, source IP addresses, FTP, LDAP, RTSP, DNS, WAP, IDS, and SIP, as well as basic SLB.

The following topics are discussed in this chapter:

For additional information on SLB commands, refer to the *Alteon Application Switch Operating System Command Reference*.

IP Server Load Balancing

IP SLB lets you perform server load balancing based on a client's IP address only. Typically, the client IP address is used with the client port number to produce a session identifier. When the Layer 3 option is enabled, Alteon uses only the client IP address as the session identifier.

To configure Alteon for IP load balancing

```
>> # /cfg/slb/virt <virtual server number>
>> Virtual Server 1# layr3 ena
>> Virtual Server 1# service ip
>> Virtual Server 1 IP Service# group <group number >
```

Note: The session that is created for the IP service ages based on setting for real server timeout.

FTP Server Load Balancing

As defined in RFC 959, FTP uses two connections: one for control information and another for data. Each connection is unique. Unless the client requests a change, the server always uses TCP port 21 (a well-known port) for control information, and TCP port 20 as the default data port.

FTP uses TCP for transport. After the initial three-way handshake, a connection is established. When the client requests any data information from the server, it issues a PORT command (such as ls, dir, get, put, mget, and mput) via the control port.

There are two FTP operation modes:

- In *Active FTP*, the FTP server initiates the data connection.
- In *Passive FTP*, the FTP client initiates the data connection. Because the client also initiates the connection to the control channel, passive FTP mode does not pose a problem with firewalls and is the most common mode of operation.

Alteon supports both active and passive FTP operation modes. You can switch from active to passive, or vice versa, in the same FTP session.

Active FTP Configuration

To create an Active FTP configuration, both the FTP and FTP-Data services must be enabled on the virtual server.

To create an Active FTP configuration

1. Add the FTP virtual service to the virtual server.

```
>> Main# /cfg/slb/virt 1/service 21
```

2. Add the FTP-Data virtual service to the virtual server.

```
>> Main# /cfg/slb/virt 1/service 20
```

3. Apply and save the configuration change.

```
>> Main# apply
>> Main# save
```

FTP Network Topology Restrictions

FTP network topology restrictions are:

- FTP control and data channels must be bound to the same real server.
- FTP with port mapping is not supported.

Configuring FTP Server Load Balancing

The following procedure is an example configuration for FTP SLB.

To configure FTP SLB

1. Ensure that a proxy IP address is enabled on the client ports, or DAM is enabled.
2. Ensure the virtual port for FTP is set up for the virtual server.

```
>> Main# /cfg/slb/virt 1/service ftp
```

3. Enable FTP parsing on the FTP service.

```
>> Main# /cfg/slb/virt 1/service 21/ftpp ena
```

4. To make your configuration changes active, apply your changes.

```
>> Virtual Server 1 ftp Service# apply
```

TFTP Server Load Balancing

As defined in RFC 1350, Trivial File Transfer Protocol (TFTP) can only read and write files from or to a remote server. TFTP uses UDP datagrams to transfer data. A transfer begins with a request to read or write a file, which also serves to request a connection. If the server grants the request, the connection is opened and the file is sent in fixed length blocks of 512 bytes.

Each data packet contains one block of data, and must be acknowledged by an acknowledgment packet before the next packet can be sent. A data packet of less than 512 bytes signals termination of a transfer.

TFTP SLB is similar to other types of server load balancing. It uses configured SLB metric to select a TFTP server. No additional commands are required to load balance to TFTP servers.

Requirements

You must select or enable the following:

- Load-balancing service port 69
- DAM

The following are not supported:

- PIP, because the server port is changed. PIP uses server port for allocating a pport.
- Multiple rports

Configuring TFTP Server Load Balancing

The following procedure is an example configuration for TFTP SLB.

To configure TFTP SLB

1. Ensure that Direct Access Mode (DAM) is enabled.
2. Ensure the virtual port for TFTP is set up for the virtual server.

```
>> # /cfg/slb/virt 1/service tftp
```

3. To make your configuration changes active, apply your changes.

```
>> Virtual Server 1 69 tftp Service# apply
```

Lightweight Directory Access Server SLB

As defined in RFC 2251, Lightweight Directory Access Protocol (LDAP) is an application-level protocol between LDAP clients and servers, which allows clients to retrieve LDAP directory entries via the Internet. The client sends a protocol operation request to the server and the server responds with a response. If it is based on TCP, port 389 is used. Once a connection is set up between the client and server, the client issues operations to the server, and the server sends responses back to the client. Before LDAP directory operations can be issued, usually a bind operation is first issued in which authorization is also sent.

LDAP Operations and Server Types

There are two kinds of LDAP operations: read and write. Clients use read operations to browse directories on servers, and use write operations to modify a directory on a server. There are two types of LDAP servers: read and write servers. Read servers only conduct read operations, and write servers perform both read and write operations.

How LDAP SLB Works

An LDAP connection is set up via Layer 4 load balancing and is bound to a read server. After that, operation frames received by Alteon are checked at Layer 7 to determine if there are any write operations. The bind and write operation data frames are stored for potential later use. When a write operation arrives, Alteon disconnects the connection to the read server and re-initiates another connection with the write server without the client's knowledge. Once the connection is set up with the write server, all the later requests go to the write server until an unbind request is received by Alteon. All these operations occur within one TCP connection.

After the reset is sent to the old server, a connection is set up to the new server. Stored data frames are forwarded to the server. After the write operation is forwarded to the server, the connection is spliced.

Selectively Resetting a Real Server

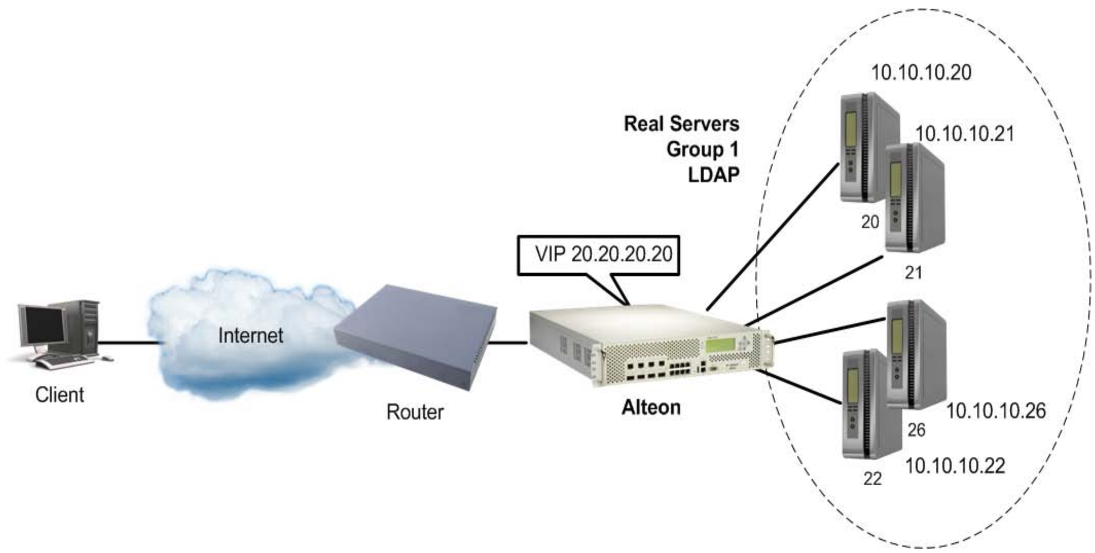
If a long-lived LDAP connection exceeds Alteon's maximum session length (32,768 minutes), the session ages out before the LDAP connection is closed. Alteon may then create another session to accept the same connection data. To prevent this, Alteon can be configured to send a reset to a real server whose session has timed out before the LDAP connection is closed.

To enable a session reset for a virtual server that is running the LDAP service

```
>> # /cfg/slb/virt 1/service ldap/reset enable
```

Figure 41 - LDAP Load Balancing, page 283 shows four LDAP servers load balancing LDAP queries:

Figure 41: LDAP Load Balancing



Configuring LDAP SLB

This procedure references Figure 41 - LDAP Load Balancing, page 283.

To configure LDAP SLB

1. Enable SLB.

```
>> # /cfg/slb/on
```

2. Configure the four real LDAP servers and their real IP addresses.

```
>> # /cfg/slb/real 20
>> Real server 20 # ena                          (Enable Real Server 20)
>> Real server 20 # rip 10.10.10.20              (Specify the IP address)
>> Real server 20 # layer7/                      (Select the Layer 7 menu)
>> Real Server 20 Layer 7 Commands# ldapwr e     (Enable LDAP read-write)
```

`/cfg/slb/real 21/ena/rip 10.10.10.21/layer7/ldapwr e`	
	(Configure and enable LDAP Write Server 21)
`/cfg/slb/real 22/ena/rip 10.10.10.22/layer7/ldapwr e`	
	(Configure and enable LDAP Write Server 21)
`/cfg/slb/real 26/ena/rip 10.10.10.26/layer7/ldapwr e`	
	(Configure and enable LDAP Write Server 21)

3. Configure Group 1 for LDAP.

`>> # /cfg/slb/group 1`	(Select real server Group 1)
`>> Real server group 1 # metric roundrobin`	(Specify the load-balancing metric for Group 1)
`>> Real server group 1 # add 20`	(Add Real Server 20)
`>> Real server group 1 # add 21`	(Add Real Server 21)
`>> Real server group 1 # add 22`	(Add Real Server 22)
`>> Real server group 1 # add 26`	(Add Real Server 26)

4. Configure and enable a virtual server IP address 1 on Alteon.

`>> # /cfg/slb/virt 1/vip 20.20.20.20`	(Specify the virtual server IP address)
`>> Virtual Server 1# ena`	(Enable the virtual server)

5. Set up the LDAP service for the virtual server, and add real server Group 1.

`>> Virtual Server 1# service ldap`	(Specify the LDAP service)
`>> Virtual Server 1 LDAP Service# group 1`	(Select the real server group)

6. Enable LDAP load balancing.

```
>> # /cfg/slb/virt 1/service ldap/ldapslb ena
```

7. Optionally, enable session reset for long LDAP connections.

```
>> # /cfg/slb/virt 1/service ldap/reset enable
```

8. Apply and save your configuration.

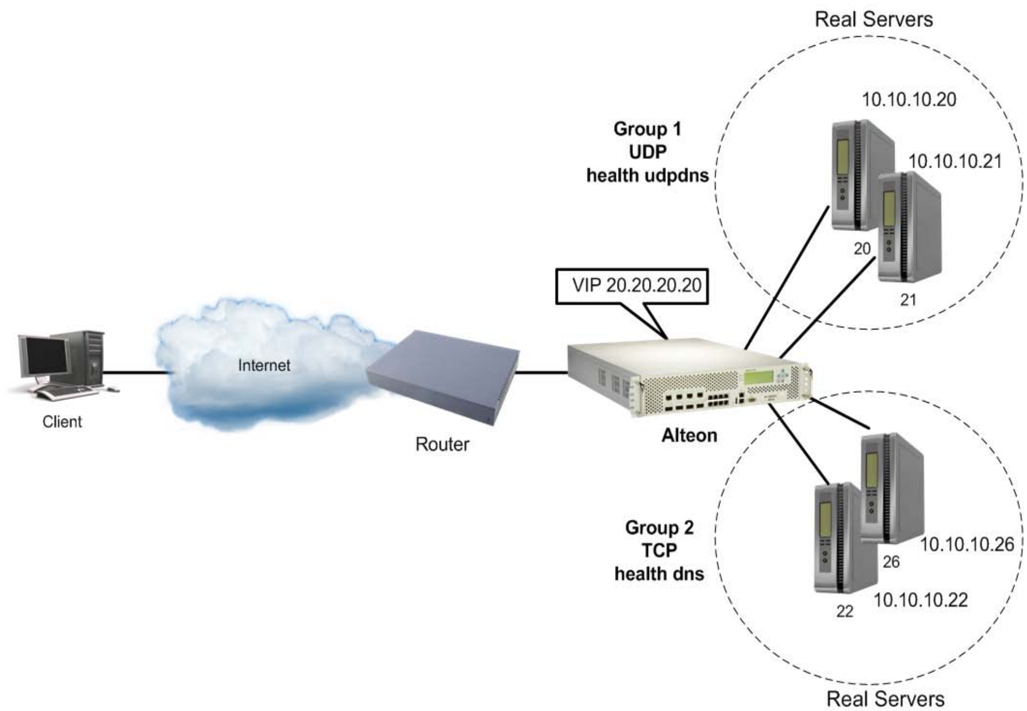
```
>> Virtual Server 1 LDAP Service# apply
>> Virtual Server 1 LDAP Service# save
```

Domain Name Server (DNS) SLB

In Alteon, DNS load balancing lets you choose the service based on the two forms of DNS queries: UDP and TCP. This enables Alteon to send TCP DNS queries to one group of real servers and UDP DNS queries to another group of real servers. The requests are then load balanced among the real servers in that group.

Figure 42 - Layer 4 DNS Load Balancing, page 285 shows four real servers load balancing UDP and TCP queries between two groups:

Figure 42: Layer 4 DNS Load Balancing



Note: You can configure both UDP and TCP DNS queries for the same virtual server IP address.

Pre-configuration Tasks

This procedure references Figure 42 - Layer 4 DNS Load Balancing, page 285.

To pre-configure Alteon for Layer 4 DNS load balancing

1. Enable SLB.

```
>> # /cfg/slb/on
```

2. Configure the four real servers and their real IP addresses.

`>> # /cfg/slb/real 20`	
`>> Real server 20 # ena`	(Enable Real Server 20)
`>> Real server 20 # rip 10.10.10.20`	(Specify the IP address)
`>> Real server 20 # /cfg/slb/real 21`	
`>> Real server 21 # ena`	(Enable Real Server 21)
`>> Real server 21 # rip 10.10.10.21`	(Specify the IP address)
`>> Real server 20 # /cfg/slb/real 22`	
`>> Real server 22 # ena`	(Enable Real Server 22)
`>> Real server 22 # rip 10.10.10.22`	(Specify the IP address)
`>> Real server 20 # /cfg/slb/real 26`	
`>> Real server 26 # ena`	(Enable Real Server 26)
`>> Real server 26 # rip 10.10.10.26`	(Specify the IP address)

3. Configure Group 1 for UDP and Group 2 for TCP.

`>> Main # /cfg/slb/group 1`	(Select Real Server Group 1)
`>> Real server group 1 # metric roundrobin`	(Specify the load-balancing metric for Group 1)
`>> Real server group 1 # health udpdns`	(Set the health check to UDP)
`>> Real server group 1 # add 20`	(Add Real Server 20)
`>> Real server group 1 # add 21`	(Add Real Server 21)
`>> Real server group 1 # /cfg/slb/group 2`	
`>> Real server group 2 # metric roundrobin`	(Specify the load-balancing metric for Group 2)
`>> Real server group 2 # health dns`	(Set the health check to TCP)
`>> Real server group 2 # add 22`	(Add Real Server 22)
`>> Real server group 2 # add 26`	(Add Real Server 26)

For more information on configuring health checks, see TCP and UDP-based DNS Health Checks, page 488.

4. Define and enable the server ports and the client ports.

   For more information, see Table 19 - Web Host Example: Port Usage, page 173. Some DNS servers initiate upstream requests and must be configured both as a server and a client.

Configuring UDP-Based DNS Load Balancing

The following procedure is an example configuration for UDP-Based DNS SLB.

To configure UDP-based DNS Load Balancing

1. Configure and enable a virtual server IP address 1 on Alteon.

```
>> # /cfg/slb/virt 1/vip 20.20.20.20          (Specify the virt server IP address)
>> Virtual Server 1# ena                      (Enable the virtual server)
```

2. Set up the DNS service for the virtual server, and add Real Server Group 1.

```
>> Virtual Server 1# service dns              (Specify the DNS service)
>> Virtual Server 1 DNS Service# group 1      (Select the real server group)
```

3. Disable delayed binding. Delayed binding is not required because UDP does not process session requests with a TCP three-way handshake.

```
>> Virtual Server 1 DNS Service# dbind dis
```

4. Enable UDP DNS queries.

```
>> Virtual Server 1 DNS Service# protocol udp
```

5. Apply and save your configuration.

```
>> Virtual Server 1 DNS Service# apply
>> Virtual Server 1 DNS Service# save
```

Configuring TCP-Based DNS Load Balancing

The following procedure is an example configuration for TCP-Based DNS SLB.

To configure TCP-based DNS load balancing

1. Configure and enable the virtual server IP address 2 on Alteon.

```
>> # /cfg/slb/virt 2/vip 20.20.20.20          (Specify the virt server IP address)
>> Virtual Server 2# ena                      (Enable the virtual server)
```

2. Set up the DNS service for virtual server, and select Real Server Group 2.

```
>> Virtual Server 2# service dns              (Specify the DNS service)
>> Virtual Server 2 DNS Service# group 2      (Select the real server group)
```

3. As this is TCP-based load balancing, ensure that you enable TCP DNS queries.

```
>> Virtual Server 2 DNS Service# protocol tcp
```

4. Apply and save your configuration.

```
>> Virtual Server 2 DNS Service# apply
>> Virtual Server 2 DNS Service# save
```

Layer 7 DNS Load Balancing

The Internet name registry has become so large that a single server cannot keep track of all the entries. This is resolved by splitting the registry and saving it on different servers.

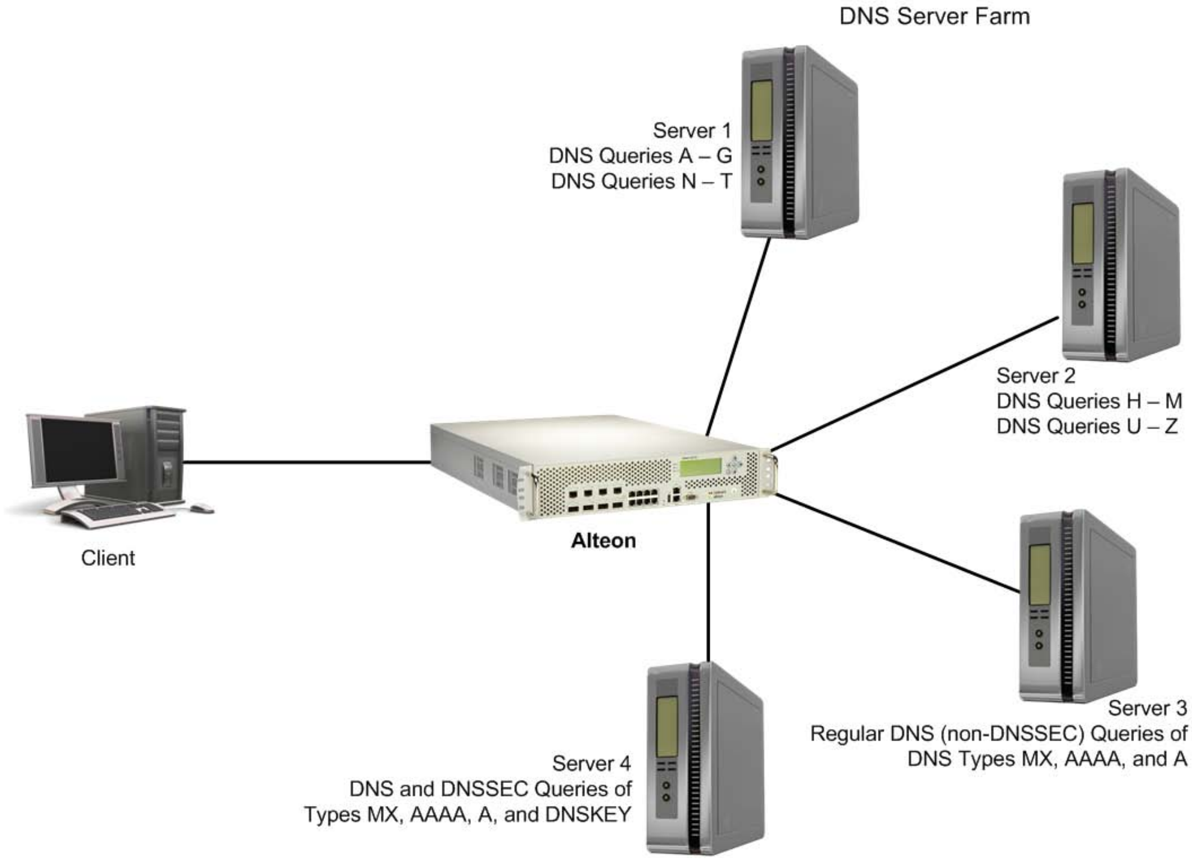
If you have large DNS server farms, Alteon lets you load balance traffic based on DNS names, DNS query types and DNS versus DNSSEC queries. To load balance DNS queries, the DNS protocol elements are extracted from the query, processed by Alteon DNS Layer 7 processing engine, and the request is sent to the appropriate real server.

Layer 7 DNS load balancing is supported for TCP/DNS and UDP/DNS (stateful) in a pure IPv4 environment (IPv4 clients and servers), and UDP/DNS (stateful) in a pure IPv6 environment (IPv6 clients and servers). For UDP stateful DNS load balancing, Alteon creates session entries in its session table, and removes them when a response is sent from the server to the client.

For example, as illustrated in Figure 43 - Load Balancing DNS Queries, page 289 a DNS server farm load balances DNS queries based on DNS names.

- Regular DNS requests with DNS names beginning with A through G and N through T are sent to Server 1.
- DNS names beginning with H through M and U through Z are sent to Server 2.
- Server 3 is an old DNS server not supporting DNSSEC queries and answers DNS queries of types MX, AAAA and A for all hostnames.
- Server 4 supports only DNSSEC queries and answers DNS types A, AAAA, MX and DNSKEY for all hostnames.

Figure 43: Load Balancing DNS Queries



To configure Alteon for DNS load balancing

1. Before you can configure DNS load balancing, ensure that Alteon is configured for basic SLB:
   — Assign an IP address to each of the real servers in the server pool.
   — Define an IP interface on Alteon.
   — Define each real server (DNS server address).
   — Assign servers to real server groups.
   — Define virtual servers and services.
   — Enable SLB.
   — Define server port and client port.

   For information on how to configure your network for SLB, see Server Load Balancing, page 165

2. Enable DNS load balancing.

   For servers 1 through 3, configure and enable a virtual server that supports only DNS load balancing (default). Virtual Server 1 performs DNS SLB for regular DNS queries and serves servers 1 through 3.

`>> # /cfg/slb/virt 1`	(Select the virtual server)
`>> Virtual Server 1 # service 53`	(Select the DNS service)

`>> Virtual Server 1 DNS Service # dnsslb ena`	(Enable DNS SLB)
`>> Virtual Server 1 DNS Service # dnstype both`	(Support DNS queries of type DNS only)

3. In addition to the TCP settings, for the virtual server, if using a TCP-based DNS server, enable delayed binding (if using a UDP-based DNS server, do not enable delayed binding).

```
>> Virtual Server 1 DNS Service # protocol tcp
>> Virtual Server 1 DNS Service# dbind ena
```

4. Define the hostnames used by servers 1 through 2.

```
>> /cfg/slb/layer7/slb/addstr DNSQ=any;TP=dns;HN=[abcdefg]+\\.com
>> /cfg/slb/layer7/slb/addstr DNSQ=any;TP=dns;HN=[hijklm]+\\.com
>> /cfg/slb/layer7/slb/addstr DNSQ=any;TP=dns;HN=[nopqrst]+\\.com
>> /cfg/slb/layer7/slb/addstr DNSQ=any;TP=dns;HN=[uvwxyz]+\\.com
```

Alternatively, use the interactive CLI. For example:

```
>> Server Load balance Resource# /cfg/slb/layer7/slb/addstr
Enter type of string [l7lkup|pattern]: l7lkup
Select Application (http|dns|other) [other]: dns
Enter DNS Type (dns, dnssec, any) [any]: dns
Enter DNS Query Type(s) (by number, query type name, or any) [any]: any
Enter DNS hostname or none [none]:   [uvwxyz]+.com
```

Note: When using the interactive menu the '\' is not inserted as in the regex format. The '\' is used to cancel the '.' as a wildcard.

5. Define the DNS query types (used by servers 3 through 4).

```
# /cfg/slb/layer7/slb/addstr DNSQ=A,AAAA,MX;TP=dns
# /cfg/slb/layer7/slb/addstr DNSQ=A,AAAA,MX,DNSKEY;TP=dnssec
```

6. Apply and save your configuration changes.

For easy configuration and identification, each defined string has an ID attached, as shown in the following table:

ID	SLB String
1	any, cont 1024
2	DNSQ=any;TP=dns;HN=[abcdefg]+\.com, cont 1024
3	DNSQ=any;TP=dns;HN=[hijklm]+\.com, cont 1024
4	DNSQ=any;TP=dns;HN=[nopqrst]+\.com, cont 1024
5	DNSQ=any;TP=dns;HN=[uvwxyz]+\.com, cont 1024
6	DNSQ=A,AAAA,MX;TP=dns, cont 1024
7	DNSQ=A,AAAA,MX,DNSKEY;TP=dnssec, cont 1024

7. Add the defined string IDs to the real server:

```
>> # /cfg/slb/real 1/layer7/addlb 2
>> # /cfg/slb/real 1/layer7/addlb 4
>> # /cfg/slb/real 2/layer7/addlb 3
>> # /cfg/slb/real 2/layer7/addlb 5
>> # /cfg/slb/real 3/layer7/addlb 6
>> # /cfg/slb/real 2/layer7/addlb 7
```

Note: If you do not add a defined string (or add the defined string **any**) the server handles any request.

Real Time Streaming Protocol SLB

The Real Time Streaming Protocol (RTSP) is an application-level protocol for control over the delivery of data with real-time properties as documented in RFC 2326. RTSP is the proposed standard for controlling streaming data over the Internet. RTSP uses RTP (Real-Time Transport Protocol) to format packets of multimedia content. RTSP is designed to efficiently broadcast audio-visual data to large groups.

Typically, a multimedia presentation consists of several streams of data (for example, video stream, audio stream, and text) that must be presented in a synchronized fashion. A multimedia client like Real Player or Quick Time Player downloads these multiple streams of data from the multimedia servers and presents them on the player screen.

RTSP is used to control the flow of these multimedia streams. Each presentation uses one RTSP control connection and several other connections to carry the audio/video/text multimedia streams. In this section, the term *RTSP server* refers to any multimedia server that implements the RTSP protocol for multimedia presentations.

Note: RTSP SLB cannot be set to **None** for the RTSP service 554.

How RTSP Server Load Balancing Works

The objective of RTSP SLB is to intelligently switch an RTSP request, and the other media streams associated with a presentation, to a suitable RTSP server based on the configured load-balancing metric. Typically, an RTSP client establishes a control connection to an RTSP server over TCP port 554 and the data flows over UDP or TCP. This port can be changed however.

Alteon supports two Layer 7 metrics, URL hashing and URL pattern matching, and all Layer 4 load-balancing metrics. This section discusses load balancing RTSP servers for Layer 4. For information on load balancing RTSP servers for Layer 7, see Content-Intelligent RTSP Load Balancing, page 295.

For information on using RTSP with cache redirection, see RTSP Cache Redirection, page 461.

Note: This feature is not applicable if the streaming media (multimedia) servers use the HTTP protocol to tunnel RTSP traffic. To ensure that RTSP SLB works, ensure the streaming media server is configured for the RTSP protocol.

Supported RTSP Servers

In a typical scenario, the RTSP client issues several sequences of commands to establish connections for each component stream of a presentation. There are several variations to this procedure, depending upon the RTSP client and the server involved. For example, there are two prominent RTSP server and client implementations.

The RTSP stream setup sequence is different for these two servers, and Alteon handles each differently:

- **Real Server**—Real Server from RealNetworks Corporation supports both UDP and TCP transport protocols for the RTSP streams. The actual transport is negotiated during the initialization of the connection. If TCP transport is selected, then all streams of data flow in the TCP control connection itself. If UDP transport is chosen, the client and server negotiate a client UDP port, which is manually configurable.

  The real media files that the Real Server plays have the extension ".rm", ".ram" or ".smil".
- **QuickTime Streaming Server**—QuickTime Streaming Server from Apple Incorporated supports a QuickTime presentation that typically has two streams and therefore uses four UDP connections exclusively for transport and one TCP control connection. QuickTime clients use a UDP port, which is manually configurable. The QuickTime files have the extension ".mov".

Alteon can also support other RTSP-compliant applications such as Microsoft Windows Media Server 9.

RTSP Port Configuration

You can also configure RTSP to use a port other than the default of 554.

To configure an RTSP port

1. Select a non-standard port to use for RTSP.

```
>> Main# /cfg/slb/virt 1/service 808
```

2. Configure RTSP load balancing on the selected port.

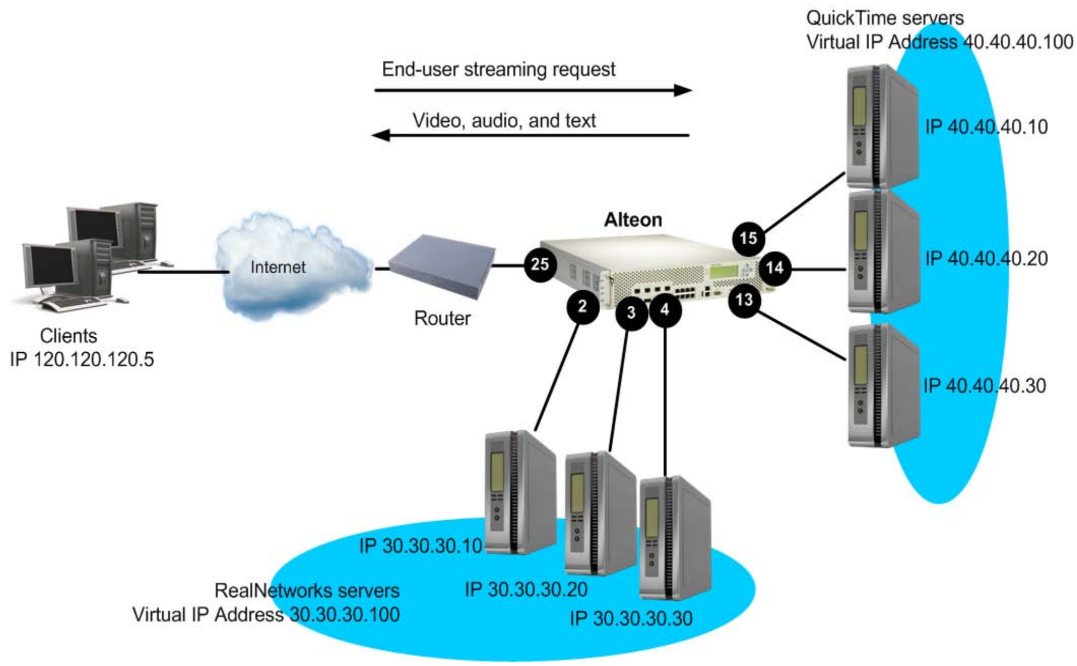
```
>> Main# /cfg/slb/virt 1/service 808/rtsp
>> Main# /cfg/slb/virt 1/service 808/rtsp/rtspslb hash
Note: The rtspslb options are: hash, pattern, l4hash, and none.
```

Configuring RTSP Load Balancing

In the example configuration illustrated in Figure 44 - Load Balancing RTSP Servers, page 293, Alteon load balances RTSP traffic between two media server farms. One group of media servers consist of QuickTime servers and the other group of servers consist of RealNetworks servers. Each group has its own virtual server IP address. For example, three Real Networks servers host media files for GlobalNews. Similarly, another three QuickTime servers host media files for GlobalNews. The content is duplicated among the servers in each group. Depending on the client request type, Alteon is configured to load balance in the following way:

- **Retrieving files from the Real Networks server group**—RTSP://www.GlobalNews.com/*.ram, RTSP://www.GlobalNews.com/*.rm, and RTSP://www.GlobalNews.com/*.smil are load balanced among the Real Networks media servers using virtual IP address 30.30.30.100.
- **Retrieving files from the QuickTime server group**—RTSP://www.GlobalNews.com/*.mov is load balanced among the Quick Time media servers using virtual IP address 40.40.40.100.

Figure 44: Load Balancing RTSP Servers



To configure RTSP load balancing

1. On Alteon, before you start configuring RTSP load balancing:
   - Connect each QuickTime server to the Layer 2 switch
   - Connect each RealNetworks server to the Layer 2 switch
   - Configure the IP addresses on all devices connected to Alteon
   - Configure the IP interfaces on Alteon
   - Enable Direct Access Mode (DAM)
   - Disable Bandwidth Management
   - Disable proxy IP addressing
2. Enable SLB.

```
>> # /cfg/slb/on
```

3. Configure IP addresses for the real servers.

```
>> # /cfg/slb/real 1/rip 30.30.30.10/ena        (Define IP address for Real Server 1)
>> # /cfg/slb/real 2/rip 30.30.30.20/ena        (Define IP address for Real Server 2)
>> # /cfg/slb/real 3/rip 30.30.30.30/ena        (Define IP address for Real Server 3)
>> # /cfg/slb/real 4/rip 40.40.40.10/ena        (Define IP address for Real Server 4)
>> # /cfg/slb/real 5/rip 40.40.40.20/ena        (Define IP address for Real Server 5)
>> # /cfg/slb/real 6/rip 40.40.40.30/ena        (Define IP address for Real Server 6)
```

4. Create a group to support RealNetworks servers.

`>> # /cfg/slb/group 100`	(Define a group)
`>>Real Server Group 100# add 1`	(Add Real Server 1)
`>>Real Server Group 100# add 2`	(Add Real Server 2)
`>>Real Server Group 100# add 3`	(Add Real Server 3)

5. Create a group to support QuickTime servers.

`>> # /cfg/slb/group 200`	(Define a group)
`>>Real Server Group 200# add 4`	(Add Real Server 4)
`>>Real Server Group 200# add 5`	(Add Real Server 5)
`>>Real Server Group 200# add 6`	(Add Real Server 6)

6. Create a virtual server for the RealNetworks servers. To configure a virtual server for Layer 4 load balancing of RTSP, select **rtsp**, or port 554, as a service for the virtual server.

`>> # /cfg/slb/virt 1`	(Select the virtual server)
`>>Virtual Server 1# vip 30.30.30.100`	(Set IP address for the virtual server
`>>Virtual Server 1# service 554`	(Add the RTSP service for the virtual server)
`>>Virtual Server 1 rtsp Service# group 100`	(Set the real server group)
`>>Virtual Server 1 rtsp Service# /cfg/slb/virt 1/ ena`	(Enable virtual server)

7. Create a virtual server for the QuickTime servers. To configure a virtual server for Layer 4 load balancing of RTSP, select rtsp, or port **554,** as a service for the virtual server.

`>> # /cfg/slb/virt 2`	(Select the virtual server)
`>>Virtual Server 2# vip 40.40.40.100`	(Set IP address for the virtual server
`>>Virtual Server 2# service 554`	(Add the RTSP service for the virtual server)
`>>Virtual Server 2 rtsp Service# group 200`	(Set the QuickTime server group)
`>>Virtual Server 2 rtsp Service# /cfg/slb/virt ena`	(Enable virtual server)

8. Enable server and client processing at the port level.

`>> # /cfg/slb/port 25`	(Select the client port)
`>>SLB port 25# client ena`	(Enable client processing)
`>>SLB port 1# /cfg/slb/port 2`	(Select the server port)
`>>SLB port 2# server ena`	(Enable server processing)
`>>SLB port 2# /cfg/slb/port 3`	(Select the server port)
`>>SLB port 3# server ena`	(Enable server processing)
`>>SLB port 3# /cfg/slb/port 4`	(Select the server port)

`>>SLB port 4# server ena`	(Enable server processing)
`>>SLB port 4# /cfg/slb/port 13`	(Select the server port)
`>>SLB port 13# server ena`	(Enable server processing)
`>>SLB port 13# /cfg/slb/port 14`	(Select the server port)
`>>SLB port 14# server ena`	(Enable server processing)
`>>SLB port 14# /cfg/slb/port 15`	(Select the server port)
`>>SLB port 15# server ena`	(Enable server processing)

9. Apply and save your configuration.

```
>> SLB port 15# apply
>> SLB port 15# save
```

Clients retrieving files of type RTSP://Globalnews.com/headlines.ram use virtual IP address 30.30.30.100 of the RealNetworks server group, and clients retrieving files of type RTSP://Globalnews.com/headlines.mov use virtual IP address 40.40.40.100 of the QuickTime server group.

Content-Intelligent RTSP Load Balancing

Alteon supports RTSP load balancing based on URL hash metric or string matching to load balance media servers that contain multimedia presentations. Because multimedia presentations consume a large amount of Internet bandwidth, and their correct presentation depends upon the real time delivery of the data over the Internet, several media servers contain the same multimedia data.

For more conceptual information on RTSP, see Real Time Streaming Protocol SLB, page 291.

Figure 45 - RTSP Load Balancing, page 296 shows two groups of media servers: Group 1 is configured for URL hashing, and Group 2 is configured for string matching. The media servers are cache servers configured in reverse proxy mode.

URL Hash

Use the URL hash metric to maximize client requests to hash to the same media server. The original servers push the content to the cache servers ahead of time. For example, an ISP is hosting audio-video files for GlobalNews on media servers 1, 2, 3, and 4. The domain name GlobalNews.com associated with the virtual IP address 120.10.10.10 is configured for URL hash.

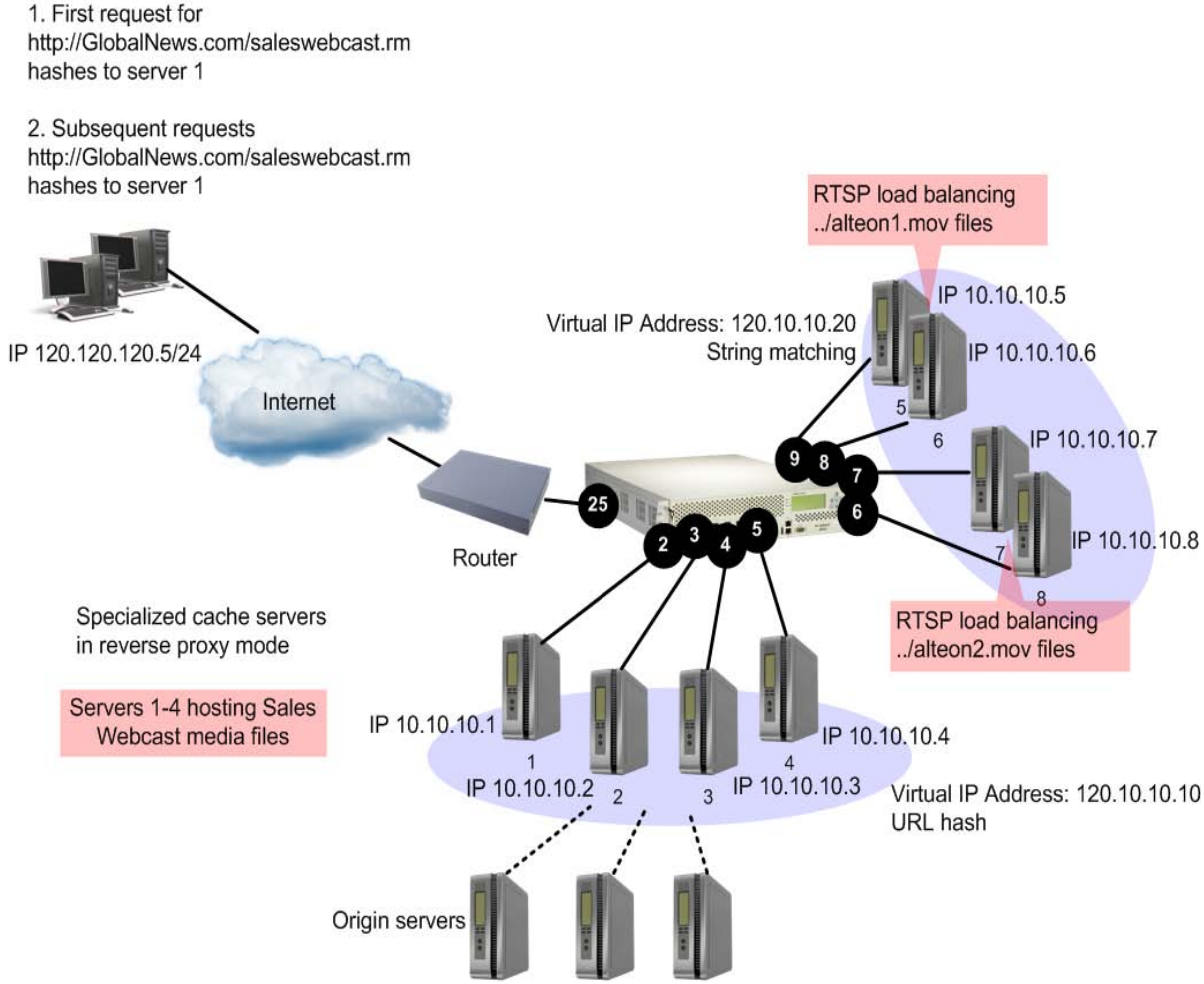
The first request for http://Globalnews.com/saleswebcast.rm hashes to media server 1. Subsequent requests for http://Globalnews.com/saleswebcast.rm from other clients or from client 1 hashes to the same Server 1. Similarly, another request for http://Globalnews.com/marketingwebcast.rm may hash to media Server 2, provided saleswebcast and marketingwebcast media files are located in the origin servers.

Typically, a set of related files (audio, video, and text) of a presentation are usually placed under the same directory (called a *container directory*). Alteon URL hashing ensures that the entire container is cached in a single cache by using the entire URL to compute the hash value and omitting the extension (for example, .ram, .rm, .smil) occurring at the end of the URL.

String Matching

Use the string matching option to populate the RTSP servers with content-specific information. For example, you have clients accessing audio-video files on Radware1 and clients accessing audio-video files on Globalnews2. You can host the Globalnews1 media files on media Servers 5 and 6, and host Globalnews2 media files on media Servers 7 and 8.

Figure 45: RTSP Load Balancing



To configure content-intelligent RTSP load balancing

1. Before you start configuring RTSP load balancing, configure Alteon for standard server load balancing, as described in Server Load Balancing Configuration Basics, page 171:
   - Connect each Media server to Alteon.
   - Configure the IP addresses on all devices connected to Alteon.
   - Configure the IP interfaces on Alteon.
   - Enable SLB **(**`/cfg/slb/on`**)**
   - Enable client processing at the client port **(**`/cfg/slb/port 1/client ena`**)**
   - Enable server processing at the Server Ports 2 and 7 (for example: `/cfg/slb/port 2/server ena`**)**
   - Enable Direct Access Mode (DAM)
   - Disable proxy IP addressing

2. Configure IP addresses for the real servers.

`>> # /cfg/slb/real 1/rip 10.10.10.1/ena`	(Define IP address for Real Server 1)
`>> # /cfg/slb/real 2/rip 10.10.10.2/ena`	(Define IP address for Real Server 2)
`>> # /cfg/slb/real 3/rip 10.10.10.3/ena`	(Define IP address for Real Server 3)
`>> # /cfg/slb/real 4/rip 10.10.10.4/ena`	(Define IP address for Real Server 4)
`>> # /cfg/slb/real 5/rip 10.10.10.5/ena`	(Define IP address for Real Server 5)
`>> # /cfg/slb/real 6/rip 10.10.10.6/ena`	(Define IP address for Real Server 6)
`>> # /cfg/slb/real 7/rip 10.10.10.7/ena`	(Define IP address for Real Server 7)
`>> # /cfg/slb/real 8/rip 10.10.10.8/ena`	(Define IP address for Real Server 8)

3. Create a group to support RealNetworks servers.

`>> # /cfg/slb/group 100`	(Define a group)
`>>Real Server Group 100# add 1`	(Add Real Server 1)
`>>Real Server Group 100# add 2`	(Add Real Server 2)
`>>Real Server Group 100# add 3`	(Add Real Server 3)
`>>Real Server Group 100# add 4`	(Add Real Server 4)

4. Create a group to support QuickTime servers.

`>> # /cfg/slb/group 200`	(Define a group)
`>>Real Server Group 200# add 5`	(Add Real Server 5)
`>>Real Server Group 200# add 6`	(Add Real Server 6)
`>>Real Server Group 200# add 7`	(Add Real Server 7)
`>>Real Server Group 100# add 8`	(Add Real Server 8)

5. Create a virtual server for Group 1 media servers. Configure a virtual server and select **rtsp**, or port 554, as a service for the virtual server.

`>> # /cfg/slb/virt 1`	(Select the virtual server)
`>>Virtual Server 1# vip 120.10.10.10`	(Set IP address for the virtual server
`>>Virtual Server 1# service 554`	(Add the RTSP service for the virtual server)
`>>Virtual Server 1 rtsp Service# group 100`	(Set the real server group)
`>>Virtual Server 1 rtsp Service# /cfg/slb/virt 1 ena`	(Enable virtual server)

6. Configure URL hash-based RTSP load balancing for Group 1 servers. URL hashing maintains persistency by enabling the client to hash to the same media server.

```
>> Virtual Server 1 rtsp Service# rtspslb hash
```

7. Create another virtual server for Group 2 media servers. Configure a virtual server and select **rtsp**, or port 554, as a service for the virtual server.

`>> # /cfg/slb/virt 2`	(Select the virtual server)
`>>Virtual Server 2# vip 120.10.10.20`	(Set IP address for the virtual server)
`>>Virtual Server 2# service 554`	(Add the RTSP service for the virtual server)
`>>Virtual Server 2 rtsp Service# group 200`	(Set the real server group)
`>>Virtual Server 2 rtsp Service# /cfg/slb/virt 2 ena`	(Enable virtual server)

8. Configure string matching-based RTSP load balancing for Group 2 servers.
   — Enable Layer 7 pattern matching

```
>> Virtual Server 2 rtsp Service# rtspslb pattern
```

   — Add URL strings.

```
>> # /cfg/slb/layer7/slb/addstr radware1.mov
>> Server Loadbalance Resource# addstr radware2.mov
```

   — Apply and save the configuration.

```
>> Server Loadbalance Resource# apply
>> Server Loadbalance Resource# save
```

   — Identify the defined string IDs.

```
>> Server Loadbalance Resource# cur
```

For easy configuration and identification, each defined string has an ID attached, as shown in the following table:

ID	SLB String
1	any, cont 1024
2	radware1.mov, cont 1024
3	radware2.mov, cont 1024

   — Add the defined string IDs to the real servers as shown in Figure 45 - RTSP Load Balancing, page 296.

```
>> # /cfg/slb/real 5/layer7
>> Real server 5 Layer 7 Commands# addlb 2
>> Real server 5# /cfg/slb/real 6/layer7
>> Real server 6 Layer 7 Commands# addlb 2
>> Real server 6# /cfg/slb/real 7/layer7
>> Real server 7 Layer 7 Commands# addlb 3
>> Real server 7# /cfg/slb/real 8/layer7
>> Real server 8 Layer 7 Commands# addlb 3
```

9. Apply and save your configuration.

```
>> Real server 8# apply
>> Real server 8# save
```

Clients retrieving RTSP://Globalnews.com/saleswebcast.rm hash to the same media server—1, 2, 3, or 4.

A client request of the form RTSP://120.10.10.20/../Globalnews1.mov is load balanced between RTSP Servers 5 and 6 using string matching. A client request of the form RTSP://120.10.10.20/../Globalnews2.mov is load balanced between RTSP Servers 7 and 8.

Secure Socket Layer (SSL) SLB

Alteon can provide SSL offloading services to any application. For HTTP over SSL (HTTPS), Alteon offers comprehensive support (see Offloading SSL Encryption and Authentication, page 337). For other applications that do not require special SSL support, Alteon can provide simple SSL offloading where the SSL is decrypted and forwarded to the servers.

Applications that require special SSL support and are not supported by Alteon include FTPS, POPS, SMTPS.

For Alteon to perform SSL offloading, you must define an SSL virtual service and associate both a server certificate and an SSL policy to it. As with other Alteon features, the virtual service is assigned to an application, in this case either HTTPS or another protocol encrypted by SSL.

For details on defining SSL policies, see SSL Policies, page 338. For details on defining server certificates, see Certificate Repository, page 338.

The following procedures are discussed in this section:

- Associating an SSL Policy to a Virtual Service, page 299
- Associating a Server Certificate to a Virtual Service, page 300

Associating an SSL Policy to a Virtual Service

When configuring an SSL virtual service, you must associate an SSL policy which defines the SSL behavior.

To associate an SSL Policy to a virtual service

1. Access the *Virtual Server Service* menu for the virtual service to which you want to associate an SSL policy. In this example, Virtual Server 1 is associated with a general SSL application.

```
>> Main# /cfg/slb/virt 1/service 12345/ssl/sslpol
```

2. Enter a new SSL policy ID (1 to 32 characters).

```
Current SSL policy:
Enter new SSL policy or none:
```

The following message displays

```
For SSL policy configuration use /cfg/slb/ssl/sslpol
```

The SSL policy name you entered is now associated with virtual service HTTPS.

3. To configure the SSL policy, see the section on the **/cfg/slb/ssl/sslpol** menu in the *Alteon Application Switch Operating System Command Reference*.

Associating a Server Certificate to a Virtual Service

When configuring an SSL virtual service, you must associate a server certificate to it. Alteon requires the server certificate and private key in order to perform SSL handshaking and be able to decrypt and encrypt traffic related to the virtual service.

To associate a server certificate to a virtual service

1. Access the *Virtual Server Service* menu for the virtual service to which you want to associate a server certificate. In this example, Virtual Server 1 is associated with a general SSL service.

```
>> Main# /cfg/slb/virt 1/service 12345/ssl/srvrcert
```

2. Enter a new server certificate ID (1 to 32 characters).

```
Current Server certificate name:
Enter new Server certificate name or none:
```

The following message displays:

```
For Server certificate configuration use /cfg/slb/ssl/certs/srvrcert
```

The server certificate name you entered is now associated with virtual service 12345.

3. To configure to the server certificate, see the section on the **/cfg/slb/ssl/certs/srvrcert** menu in the *Alteon Application Switch Operating System Command Reference*.

Notes

- You can associate only a single server certificate to a virtual service.
- When the virtual service is enabled and you associate an SSL policy with a virtual service without a certificate and try to apply the changes with the ***apply*** command, you receive an error message. The SSL offloading capabilities can be set only with both a server certificate and SSL policy associated with a virtual service.

Wireless Application Protocol (WAP) SLB

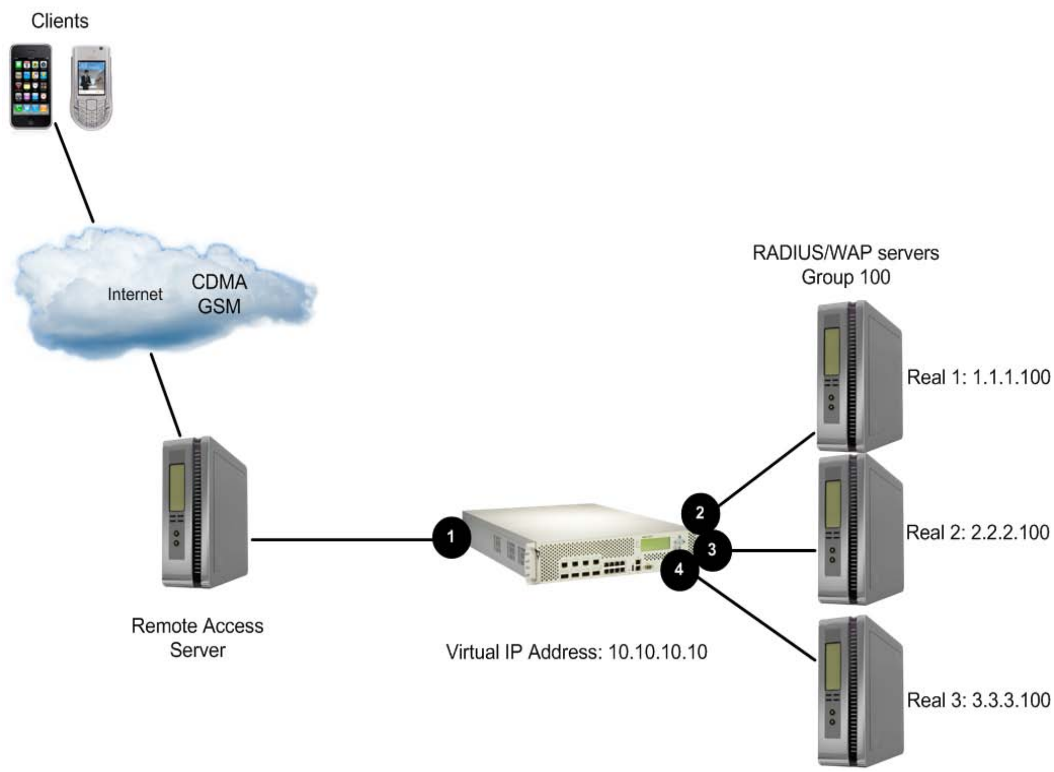
The Wireless Application Protocol (WAP) is an open, global specification for a suite of protocols designed to allow wireless devices to communicate and interact with other devices. It empowers mobile users with wireless devices to easily access and interact with information and services instantly by allowing non-voice data, such as text and images, to pass between these devices and the Internet. Wireless devices include cellular phones, pagers, Personal Digital Assistants (PDAs), and other hand-held devices.

WAP supports most wireless networks and is supported by all operating systems, with the goal of inter-operability. A WAP gateway translates Wireless Markup Language (WML) (which is a WAP version of HTML) into HTML/HTTP so that requests for information can be serviced by traditional Web servers.

To load balance WAP traffic among available parallel servers, Alteon must provide *persistency* so that the clients can always go to the same WAP gateway to perform WAP operation.

Figure 46 - Load Balancing WAP Gateways, page 301 illustrates how the user is first authenticated by the remote access server. In this example, the RADIUS servers are integrated with the WAP gateways:

Figure 46: Load Balancing WAP Gateways



You can configure Alteon to select a WAP gateway for each client request based on one of the following three methods:

- WAP SLB with RADIUS Static Session Entries, page 301
- WAP SLB with RADIUS Snooping, page 304
- WAP SLB with RADIUS/WAP Persistence, page 306

WAP SLB with RADIUS Static Session Entries

RADIUS, a proposed IETF standard, is a client/server protocol that enables remote access servers to communicate with a central server to authenticate dial-in users and authorize their access to the requested network or service. RADIUS allows a company to maintain user profiles in a central database that all remote servers can share. It provides better security, allowing a company to set up a policy that can be applied at a single-administered network point.

The RADIUS server uses a static session entry to determine which real WAP gateway should receive the client sessions. Typically, each WAP gateway is integrated with a RADIUS server on the same host, and a RADIUS request packet is allowed to go to any of the RADIUS servers. Upon receiving a request from a client, the RADIUS server instructs Alteon to create a static session entry via the Transparent Proxy Control Protocol (TPCP). TPCP is a proprietary protocol that is used to establish communication between the RADIUS servers and Alteon. It is UDP-based and uses ports 3121, 1812, and 1645.

The RADIUS servers use TPCP to add or delete static session entries on Alteon. Typically, a regular session entry is added or removed by Alteon itself. A static session entry, like a regular session entry, contains information such as the client IP address, the client port number, real port number, virtual (destination) IP address, and virtual (destination) port number.

A static session entry added via TPCP to Alteon does not age out. The entry is only deleted by another TPCP *Delete Session* request. If the user adds session entries using the traditional server load balancing methods, the entries will continue to age out.

Because TPCP is UDP-based, the *Add/Delete Session* requests may get lost during transmission. The WAP gateway issues another Add Session request on detecting that it has lost a request. The WAP gateway detects this situation when it receives WAP traffic that does not belong to that WAP gateway. If a Delete Session request is lost, it is overwritten by another Add Session request.

How WAP SLB Works with Static Session Entries

1. On dialing, the user is first authenticated by the Remote Access Server (RAS).
2. The RAS sends a RADIUS authentication request to one of the RADIUS servers, which can be integrated with a WAP gateway.
3. If the user is accepted, the RADIUS server determines which WAP gateway is right for this user and informs Alteon of the decision via TPCP.
4. Alteon receives a request from the RADIUS server, and adds a session entry to its session table to bind a WAP gateway with that user.
5. A response packet is sent back to the RAS by the RADIUS server.
6. The RAS receives the packet and allows the WAP traffic for that user.
7. If the user disconnects, the RAS detects it and sends this information to the RADIUS server.
8. The RADIUS server removes the session entry for that user.

Configuring WAP SLB using Static Session Entries

This procedure references Figure 46 - Load Balancing WAP Gateways, page 301.

To configure WAP SLB using static session entries

1. Before you start configuring WAP load balancing:
   — Enable Layer 3 server load balancing.

```
>> # /cfg/slb/virt <number> /layr3 ena
```

   — Enable UDP under the *WAP services* (ports 9200 to 9203) menu.

```
>> # /cfg/slb/virt <number> /service <name|number> /protocol udp
```

— Configure for RADIUS services 1812, 1813, and 1645.

Note: If the application is not recognized by the port, set the application to **basic-slb**.

```
>> # /cfg/slb/virt <number> /service <name|number> /protocol udp
```

Note: The RADIUS service number specified on Alteon must match with the service specified on the server.

2. Configure Alteon for basic SLB.

```
>> # /cfg/slb/on
```

3. Configure IP addresses for the RADIUS/WAP gateways.

```
>> # /cfg/slb/real 1/rip 1.1.1.100        (Define address for WAP Gateway1)
>> Real server 1# ena                     (Enable Real Server 1)
>> # /cfg/slb/real 2/rip 2.2.2.100        (Define address for WAP Gateway 2)
>> Real server 2# ena                     (Enable Real Server 2)
>> # /cfg/slb/real 3/rip 3.3.3.100        (Define address for WAP Gateway 3)
>> Real server 3# ena                     (Enable Real Server 3)
```

4. Create a group to load balance the WAP gateways.

```
>> # /cfg/slb/group 100                   (Define a group)
>>Real Server Group 100# add 1            (Add Real Server 1)
>>Real Server Group 100# add 2            (Add Real Server 2)
>>Real Server Group 100# add 3            (Add Real Server 3)
```

5. Enable the external notification from the WAP gateway to add and delete session requests if you are using static session via TPCP.

```
>> # cfg/slb/adv/tpcp ena
```

6. Enable TPCP for adding and deleting WAP sessions.

```
>> # cfg/slb/wap/tpcp ena
```

7. Apply and save your configuration.

```
>> WAP Options# apply
>> WAP Options# save
```

WAP SLB with RADIUS Snooping

RADIUS snooping is similar to the static session entry method in the way that a static session entry is added to, or removed from, Alteon for the WAP traffic for a user. It is different from the static session entry method in the way that RADIUS accounting packets are snooped by Alteon instead of by the RADIUS server using TPCP.

RADIUS snooping enables Alteon to examine RADIUS accounting packets for client information. This information is needed to add to or delete static session entries in the Alteon session table so that it can perform the required persistency for load balancing. A static session entry does not age out. Such an entry, added using RADIUS snooping, is only deleted using RADIUS snooping. Alteon load balances both the RADIUS and WAP gateway traffic using the same virtual server IP address.

How WAP SLB Works with RADIUS Snooping

Before the Remote Access Service (RAS) allows the WAP traffic for a user to pass in and out of the gateway, it sends a *RADIUS Accounting Start* message to one of the RADIUS servers. Alteon then *snoops* on the packet to extract the required information. It needs to know the type of the RADIUS Accounting message, the client IP address, the caller ID, and the user's name. If it finds this information, Alteon adds a session entry to its session table. If any of this information is missing, Alteon does not take any action to handle the session entry.

When the client ends the WAP connection, the RAS sends an *RADIUS Accounting Stop* packet. If Alteon finds the needed information in a RADIUS Accounting Stop packet, it removes the corresponding session entry from its table.

The following steps occur when using RADIUS snooping:

1. The user is authenticated on dialing.
2. The RAS establishes a session with the client and sends a RADIUS Accounting Start message with the client IP address to the RADIUS server.
3. Alteon snoops on the RADIUS accounting packet and adds a session entry if it finds enough information in the packet.
4. Alteon load balances the WAP traffic to a specific WAP gateway.
5. When the client terminates the session, the RAS sends an Accounting Stop message to the RADIUS server, and the session entry is deleted from Alteon.

Review the following guidelines before configuring RADIUS snooping:

- The same virtual server IP address must be used when load balancing both the RADIUS accounting traffic and WAP traffic.
- All the RADIUS servers must use the same UDP port for RADIUS accounting services.
- Before a session entry is recorded on Alteon, WAP packets for a user can go to any of the real WAP gateways.
- If a session entry for a client cannot be added because of resource constraints, the subsequent WAP packets for that client will not be load balanced correctly. The client will need to drop the connection and then reconnect to the wireless service.
- The persistence of a session cannot be maintained if the number of healthy real WAP gateways changes during the session. For example, if a new WAP server comes into service or some of the existing WAP servers are down, the number of healthy WAP gateway changes and, in this case, the persistence for a user cannot be maintained.
- Persistence cannot be maintained if the user moves from one ISP to another, or if the base of the user's session changes (that is, from CALLING_STATION_ID to USER_NAME, or vice versa). For example, if a user moves out of a roaming area, it is possible that the user’s CALLING_STATION_ID is not available in the RADIUS accounting packets. In such a case, Alteon uses USER_NAME to choose a WAP server instead of CALLING_STATION_ID. As a result, persistence cannot be maintained.

Configuring WAP SLB using RADIUS Snooping

This procedure references Figure 46 - Load Balancing WAP Gateways, page 301.

To configure WAP SLB using RADIUS snooping

1. Before you start configuring WAP load balancing:
   — Enable Layer 3 server load balancing.

```
>> # /cfg/slb/virt <number> /layr3 ena
```

   — Enable UDP under the *WAP services* (ports 9200 to 9203) menu.

```
>> # /cfg/slb/virt <number> /service <name|number> /protocol udp
```

   — Configure for RADIUS services 1812, 1813, and 1645.

```
>> # /cfg/slb/virt <number> /service <name|number> /protocol udp
```

Note: The RADIUS service number specified on Alteon must match the service specified on the server.

2. Configure Alteon for basic SLB.

```
>> # /cfg/slb/on
```

3. Configure IP addresses for the RADIUS/WAP gateways.

```
>> # /cfg/slb/real 1/rip 1.1.1.100      (Define address for WAP Gateway1)
>> Real server 1# ena                   (Enable Real Server 1)
>> # /cfg/slb/real 2/rip 2.2.2.100      (Define address for WAP Gateway 2)
>> Real server 2# ena                   (Enable Real Server 2)
>> # /cfg/slb/real 3/rip 3.3.3.100      (Define address for WAP Gateway 3)
>> Real server 3# ena                   (Enable Real Server 3)
```

4. Create a group to load balance the WAP gateways.

```
>> # /cfg/slb/group 100                 (Define a group)
>>Real Server Group 100# add 1          (Add Real Server 1)
>>Real Server Group 100# add 2          (Add Real Server 2)
>>Real Server Group 100# add 3          (Add Real Server 3)
```

5. Enable the external notification from WAP gateway to add and delete session requests if you are using static session via TPCP.

```
>> # cfg/slb/adv/tpcp ena
```

6. Enable TPCP for adding and deleting WAP sessions.

```
>> # cfg/slb/wap/tpcp ena
```

7. Configure the following filter on Alteon to examine a RADIUS accounting packet. Set the basic filter parameters

`>> # /cfg/slb/filt 1`	(Select the filter)
`>> Filter 1  # ena`	(Enable the filter)
`>> Filter 1  # dip 10.10.10.100`	(Set the destination IP address)
`>> Filter 1  # dmask 255.255.255.255`	(Set the destination IP mask)
`>> Filter 1  # proto udp`	(Set the protocol to UDP)
`>> Filter 1  # dport 1813`	(Set the destination port)
`>> Filter 1  # action redir`	(Set the action to redirect)
`>> Filter 1  # group 1`	(Set the group for redirection)
`>> Filter 1  # rport 1813`	(Set server port for redirection)

8. Enable proxy and RADIUS snooping.

`>> Filter 1  # adv`	(Select the *Advanced Filter* menu)
`>> Filter 1 Advanced# proxy ena`	(Enable proxy)
`>> Filter 1 Advanced# layer7`	(Select the *Layer 7 Advanced* menu)
`>> Layer 7 Advanced# rdsnp ena`	(Enable RADIUS snooping)

9. Apply and save your configuration.

```
>> Layer 7 Advanced# apply
>> Layer 7 Advanced# save
```

Note: Alteon supports Virtual Router Redundancy Protocol (VRRP) and stateful failover, using both static session entries and RADIUS snooping. However, the active-active configuration with stateful failover is not supported.

WAP SLB with RADIUS/WAP Persistence

This feature enables RADIUS and WAP persistence by binding both RADIUS accounting and WAP sessions to the same server.

A WAP client is first authenticated by the RADIUS server on UDP port 1812. The server replies with a RADIUS accept or reject frame. Alteon forwards this reply to the RAS. After the RAS receives the RADIUS accept packet, it sends a RADIUS accounting start packet on UDP port 1813 to the bound server. Alteon snoops on the RADIUS accounting start packet for the **framed IP address** attribute. The **framed IP address** attribute is used to rebind the RADIUS accounting session to a new server.

The following steps occur when using RADIUS/WAP persistence:

1. The user is authenticated on dialing.

   The RAS sends a RADIUS authentication request on UDP port 1812 to one of the servers. Alteon receives the authentication request. If there is no session corresponding to this request, a new session is allocated and the client is bound to a server. Alteon then relays the authentication request to the bound server.
2. The RAS establishes a session with the client and sends a RADIUS accounting start message to the RADIUS server on UDP port 1813.
3. Alteon snoops on the RADIUS accounting start packet for the **framed IP address** attribute.

   This attribute in a RADIUS accounting packet contains the IP address of the specific client (the IP address of the wireless device).

   Note: The RADIUS accounting packet and the RADIUS accounting service must share the same rport.
4. The **framed IP address** attribute is used to rebind the RADIUS session to a new server.

   Alteon hashes on the framed IP address to select a real server for the RADIUS accounting session. If the **framed IP address** is not found in the RADIUS accounting packet, then persistence is not maintained for the RADIUS/WAP session. The load-balancing metric of the real server group has to be hash for RADIUS/WAP Persistence
5. When the client begins to send WAP requests to the WAP gateways on ports 9200 through 9203, a new session is allocated and a server is bound to the WAP session.

   The RADIUS session and the WAP session are now both bound to the same server because both sessions are using the same source IP address.

Configuring WAP SLB using RADIUS/WAP Persistence

This procedure references Figure 46 - Load Balancing WAP Gateways, page 301.

1. Configure Alteon for basic SLB.

```
>> # /cfg/slb/on
```

2. Configure IP addresses for the RADIUS/WAP Gateways.

`>> # /cfg/slb/real 1/rip 1.1.1.100`	(Define address for WAP Gateway1)
`>> Real server 1# ena`	(Enable Real Server 1)
`>> # /cfg/slb/real 2/rip 2.2.2.100`	(Define address for WAP Gateway 2)
`>> Real server 2# ena`	(Enable Real Server 2)
`>> # /cfg/slb/real 3/rip 3.3.3.100`	(Define address for WAP Gateway 3)
`>> Real server 3# ena`	(Enable Real Server 3)

3. Create a group to load balance the WAP gateways.

`>> # /cfg/slb/group 100`	(Define a group)
`>>Real Server Group 100# metric hash`	(Select hash as load-balancing metric)
`>>Real Server Group 100# add 1`	(Add Real Server 1)

`>>Real Server Group 100# add 2`	(Add Real Server 2)
`>>Real Server Group 100# add 3`	(Add Real Server 3)

4. Configure a virtual server.

`>> # cfg/slb/virt 1/vip 10.10.10.10` `>>Virtual Server 1# ena`	(Enable Virtual Server 1)

5. Configure the services for Virtual Server 1.

Notes

- The RADIUS service number specified on Alteon must match with the service specified on the server.
- If the application is not recognized by the port, set the application as **basic-slb**.

```
>>Virtual Server 1# service 1812
>>Virtual Server 1 radius-auth service# protocol udp
>>Virtual Server 1 radius-auth service# /cfg/slb/virt 1/service 1813
>>Virtual Server 1 radius-acc service# protocol udp
>>Virtual Server 1 radius-auth service# /cfg/slb/virt 1/service 9200
>>Virtual Server 1 9200 service# protocol udp
>>Virtual Server 1 radius-auth service# /cfg/slb/virt 1/service 9201
>>Virtual Server 1 9201 service# protocol udp
>>Virtual Server 1 radius-auth service# /cfg/slb/virt 1/service 9202
>>Virtual Server 1 9202 service# protocol udp
>>Virtual Server 1 radius-auth service# /cfg/slb/virt 1/service 9203
>>Virtual Server 1 9203 service# protocol udp
```

6. Configure the following filter to examine a RADIUS accounting packet. Set the basic filter parameters.

`>> # /cfg/slb/filt 1`	(Select the filter)
`>> Filter 1 # ena`	(Enable the filter)
`>> Filter 1 # dip 10.10.10.10`	(Set the destination IP address)
`>> Filter 1 # dmask 255.255.255.255`	(Set the destination IP mask)
`>> Filter 1 # proto udp`	(Set the protocol to UDP)
`>> Filter 1 # dport 1813`	(Set the destination port)
`>> Filter 1 # action redir`	(Set the action to redirect)
`>> Filter 1 # group 100`	(Set the group for redirection)
`>> Filter 1 # rport 1813`	(Set server port for redirection)

7. Enable RADIUS/WAP persistence.

`>> # /cfg/slb/filt 1`	(Select the filter)
`>> Layer 7 Advanced# rdswap ena`	(Enable RADIUS/WAP persistence)

8. Enable client and server ports and enable filtering on client ports.

```
>> # /cfg/slb/port 1/client ena
>> SLB port 1# filt ena                    (Enable filtering on Port 1)
>> SLB port 1# /cfg/slb/port 1
>> SLB port 2# /cfg/slb/server ena
>> SLB port 1# /cfg/slb/port 2
>> SLB port 2# /cfg/slb/server ena
>> SLB port 1# /cfg/slb/port 3
>> SLB port 3# /cfg/slb/server ena
>> SLB port 3# /cfg/slb/port 4
>> SLB port 4# /cfg/slb/server ena
```

9. Apply and save your configuration.

```
>> SLB port 4# apply
>> SLB port 4# save
```

Intrusion Detection System (IDS) SLB

The Intrusion Detection System (IDS) is a type of security management system for computers and networks. An Intrusion Detection System gathers and analyzes information from various areas within a computer or a network to identify possible security breaches, which include both intrusions (attacks from outside the organization) and misuse (attacks from within the organization).

This section includes the following topics:

- How Intrusion Detection Server Load Balancing Works, page 309
- Setting Up IDS Servers, page 311
- IDS Load Balancing Configurations, page 311

Intrusion detection functions include:

- Monitoring and analyzing both user and system activities
- Analyzing system configurations and vulnerabilities
- Assessing system and file integrity
- Recognizing patterns typical of attacks
- Analyzing abnormal activity patterns
- Tracking user policy violations

Intrusion detection devices inspect every packet before it enters a network, looking for any signs of an attack. The attacks are recorded and logged in an attempt to guard against future attacks and to record the information about the intruders.

IDS SLB helps scale intrusion detection systems since it is not possible for an individual server to scale information being processed at Gigabit speeds.

How Intrusion Detection Server Load Balancing Works

Alteon can forward a copy of the IP packets to an Intrusion Detection server. IDS SLB must be enabled on the incoming ports and enabled for the groups containing the IDS real servers. The IDS SLB-enabled device copies packets entering IDS-enabled ports. An SLB session is created on Alteon to a group of intrusion detection servers. The IDS server is selected based on the IDS group metric.

The following summarizes the primary steps involved in configuring IDS load balancing:

1. Set up the IDS servers.

   Determine if you want to setup the IDS servers in stealth mode (without IP addresses) or with non-routable IP addresses. For more information about setting up IDS servers, see Setting Up IDS Servers, page 311.

2. Create the IDS groups.

   Create real server groups for the IDS servers. You may create multiple IDS groups to segregate incoming traffic based on protocols.

   — Choose the metric for the group as **hash**

   — Choose the health check for the group: **link**, **icmp**, **arp**, or **snmp**

   — Enable IDS on the group

   — Select the type of traffic that is captured by the group by defining the IDS rport value. Default: **any**

     If multiple groups are configured for the same rport, then only *one* of the groups is used for SLB.

3. Enable IDS on the incoming ports (both client and server ports).

   Enabling IDS at the port level enables Alteon to make a copy of the frames ingressing the port and forward the copy to the IDS server group.

4. Configure filter processing on the incoming ports with the IDS hash metric.

   This allows a session entry to be created for frames ingressing the port. IDS load balancing requires a session entry to be created in order to store the information regarding which IDS server to send the request.

   If the allow filter is configured to hash on both the client and server IP address, then this ensures that both client and server traffic belonging to the same session is sent to the same IDS server. For more information, see Example 2: Load Balancing to Multiple IDS Groups, page 315. If the port is configured for client processing only, then Alteon hashes on the source IP address only.

Setting Up IDS Servers

Table 27 illustrates how to configure IDS servers, depending on the IDS server type:

Table 27: Setting Up IDS Servers

IDS Server Configuration	Health Check Type	Port Configuration	Explanation
Stealth mode (without IP addresses or dummy IP addresses)	Link	• IDS servers must directly connect to separate physical ports on Alteon. • The real server number of IDS server must match the physical port number (1 to 26) on Alteon.	To send packets to different IDS servers, you must connect IDS servers to separate ports and associate them with different VLANs and tag the packets accordingly. Because unmodified frames are sent to the IDS servers, Alteon does not use the L2 destination field of the packet to direct it to the correct IDS server. The port or the VLAN tag is used to identify the destination IDS server. However, if the ingress packet is already tagged, you must use different ports for different IDS servers.
Stealth mode (without IP addresses or dummy IP addresses)	SNMP	IDS servers need not be directly connected to Alteon.The IDS servers may be connected to another switch via an interswitch link between it and Alteon. SNMP health checks are used to check the status of a port/VLAN on the remote device that is connected to an IDS server.	To send packets to different IDS servers, you must connect IDS servers to separate ports and associate them with different VLANs. Because unmodified frames are sent to the IDS servers, Alteon does not use the L2 destination field of the packet to direct it to the correct IDS server. The VLAN tag is used to identify the destination IDS server. However, if the ingress packet is already tagged, you must use different VLANs for different IDS servers.
With IP addresses	ICMP or ARP	IDS servers need not be directly connected to Alteon.The IDS servers may be connected via an Alteon or a Layer 2 switch.	The data packet is modified, so that it is addressed to the IDS servers. Destination MAC address is changed to the real server MAC address.

IDS Load Balancing Configurations

The examples in this section illustrate IDS load balancing in two different network environments:

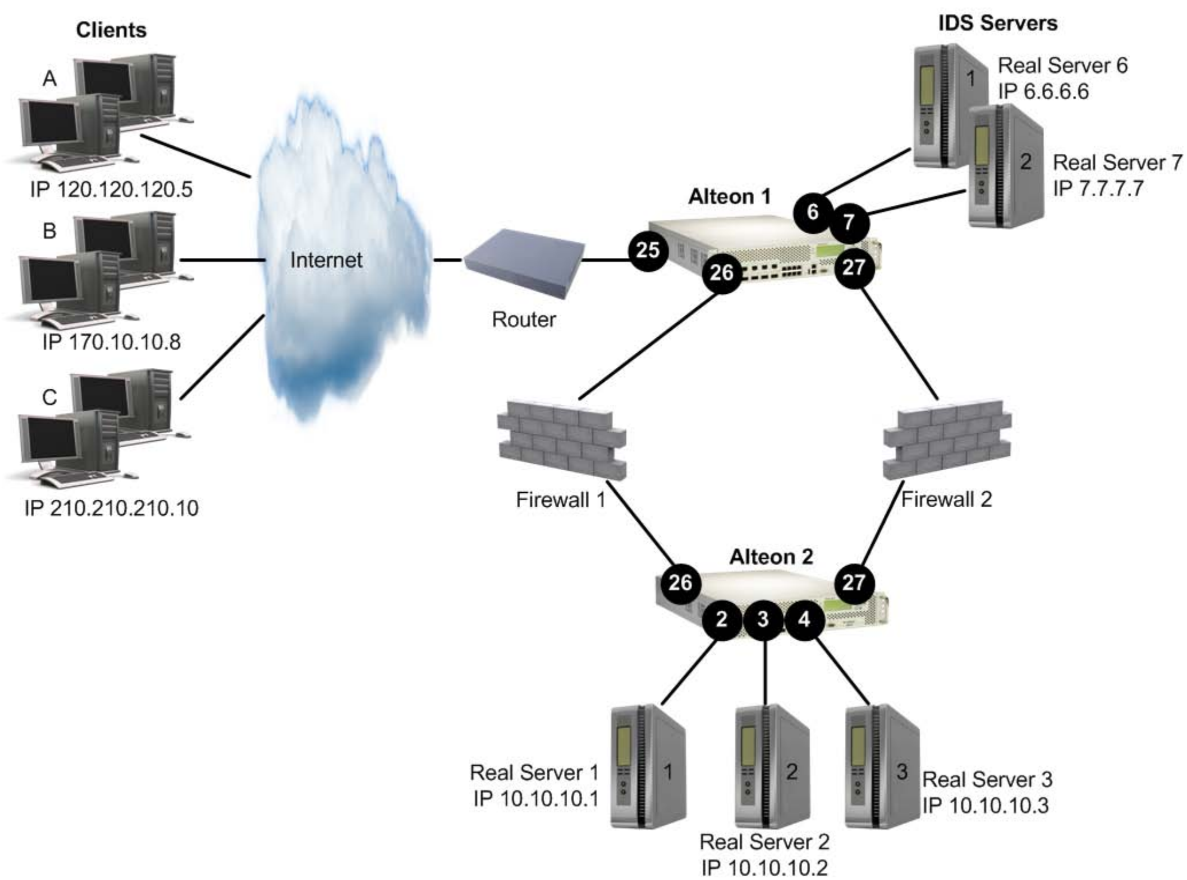
- Example 1: Load Balancing to a Single IDS Group, page 312—One Alteon is dedicated to load balancing two IDS servers in a single group, and a second Alteon performs standard server load balancing.
- Example 2: Load Balancing to Multiple IDS Groups, page 315—A single Alteon performs both IDS load balancing and standard server load balancing. Two IDS groups are configured: IDS Group 51 is for HTTP traffic only, and IDS Group 52 is for all other traffic.

- Example 3: Load Balancing IDS Servers Across Multiple Alteons, page 318—Two Alteons in a high availability configuration are connected to each other via a trunked interswitch link that is associated with all VLANs configured on both Alteons. Each Alteon is connected to IDS servers that are each on different VLANs but belong to the same IDS group. A feature to disable source MAC address learning across the interswitch link allows traffic to reach real servers even when one Alteon goes into the standby state.

Example 1: Load Balancing to a Single IDS Group

Figure 47 - Server Load Balancing and IDS Load Balancing to a Single Group, page 312 illustrates a basic configuration for load balancing client and server traffic to the IDS servers. Alteon 1 performs IDS load balancing and Alteon 2 performs standard server load balancing. IDS is enabled on the client port (port 25) and both the firewall ports (ports 26 and 27).

Figure 47: Server Load Balancing and IDS Load Balancing to a Single Group



When the client request enters port 25 on Alteon 1, Alteon 1 makes a copy of the packet. Alteon load balances the copied packet between the two IDS servers based on the configured load balancing metric (hash). The original data packet however, enters Alteon 2 through the firewall and Alteon 2 performs standard server load balancing on the client data between the three real servers. The client request is processed and returned to Alteon 1 via the firewall. An allow filter at ports 26 and port 27 causes Alteon to make a copy of the request and directs the copy to the IDS server group.

To load balance to a single IDS group

1. Set up the IDS servers.

   To configure the IDS servers as real servers you must consider the setup of the IDS servers and the selection of the health check. Typically, most IDS servers are set up in stealth mode (without IP addresses). However, they can also be set up with non-routable IP addresses. For more information about setting up IDS servers, see Setting Up IDS Servers, page 311.

2. Configure the IDS servers as real servers.

   The IDS servers are configured in stealth mode. Match the real server number with the physical port number to which the IDS servers are connected, and configure dummy IP addresses. The real servers must be numbered between 1 and 63.

`>> # /cfg/slb/real 6/rip 6.6.6.6/ena`	(Define a dummy IP address for IDS Server 6)
`>> # /cfg/slb/real 7/rip 7.7.7.7/ena`	(Define a dummy IP address for IDS Server 7)

3. Create a group and add the IDS servers. The group must be numbered between 1 and 63.

`>> # /cfg/slb/group 50`	(Define a group)
`>>Real Server Group 50# add 6`	(Add IDS Server 6)
`>>Real Server Group 50# add 7`	(Add IDS Server 7)

4. Define the group metric for the IDS server group. IDS SLB supports the hash metric only.

```
>>Real Server Group 50# metric hash
```

5. Define the health check for the group. Configure link health check which is specifically developed for IDS servers set up in stealth mode (without IP addresses).

```
>>Real Server Group 50# health link
```

6. Define the group for IDS SLB.

```
>>Real Server Group 50# ids ena
```

7. Select the rport for the IDS group.

```
>>Real Server Group 50# idsrprt any
```

8. Enable IDS on the client and server ports. This enables frames ingressing the port to be copied to the IDS servers.

`>># /cfg/slb/port 25/ids ena`	(Enable IDS processing for port 25)
`>>SLB port 25# /cfg/slb/port 26/ids ena`	(Enable IDS processing for port 26)
`>>SLB port 26# /cfg/slb/port 27/ids ena`	(Enable IDS processing for port 27)

In addition to enabling IDS at the port level, a filter must be configured to create a session entry for non-SLB frames ingressing the port. IDS load balancing requires a session entry to be created to store the information regarding which IDS server to send to.

9. Create an allow filter and configure the filter with the idshash metric.

`>> # /cfg/slb/filt 2048`	(Select the menu for Filter 2048)
`>> Filter 2048# sip any`	(From any source IP address)
`>> Filter 2048# dip any`	(To any destination IP address)
`>> Filter 2048# action allow`	(Allow matching traffic to pass)
`>> Filter 2048# ena`	(Enable the filter)
`>> Filter 2048# adv/idshash both`	(Set the hash metric parameter)

The IDS hash metric is set to hash on both the source and destination IP addresses. Hashing on both source and destination IP address ensures that the returning traffic goes to the same IDS server. If the port is configured for client processing only, then Alteon hashes on the source IP address. By default, the IDS hash metric hashes on the source IP address only.

10. Apply the allow filter to ports 25, 26, and 27. The allow filter must be applied on all ports that require Layer 4 traffic to be routed to the IDS servers.

`>> Filter 2048# /cfg/slb/port 25`	(Select the client port)
`>> SLB Port 25# add 2048`	(Apply the filter to the client port)
`>> SLB Port 25# filt ena`	(Enable the filter)
`>> SLB Port 25# /cfg/slb/port 26`	(Select port 26)
`>> SLB Port 26# add 2048`	(Apply the filter to port 26)
`>> SLB Port 26# filt ena`	(Enable the filter)
`>> SLB Port 26# /cfg/slb/port 27`	(Select port 27)
`>> SLB Port 27# add 2048`	(Apply the filter to port 27)
`>> SLB Port 27# filt ena`	(Enable the filter)

All ingressing traffic at these ports that match any of the filters configured for that port are load balanced to the IDS groups. The allow filter is used at the end of the filter list to ensure that all traffic matches a filter. A **deny all** filter can also be used as the final filter instead of an allow all filter.

11. Apply and save your changes.

```
>> SLB Port 25# apply
>> SLB Port 25# save
```

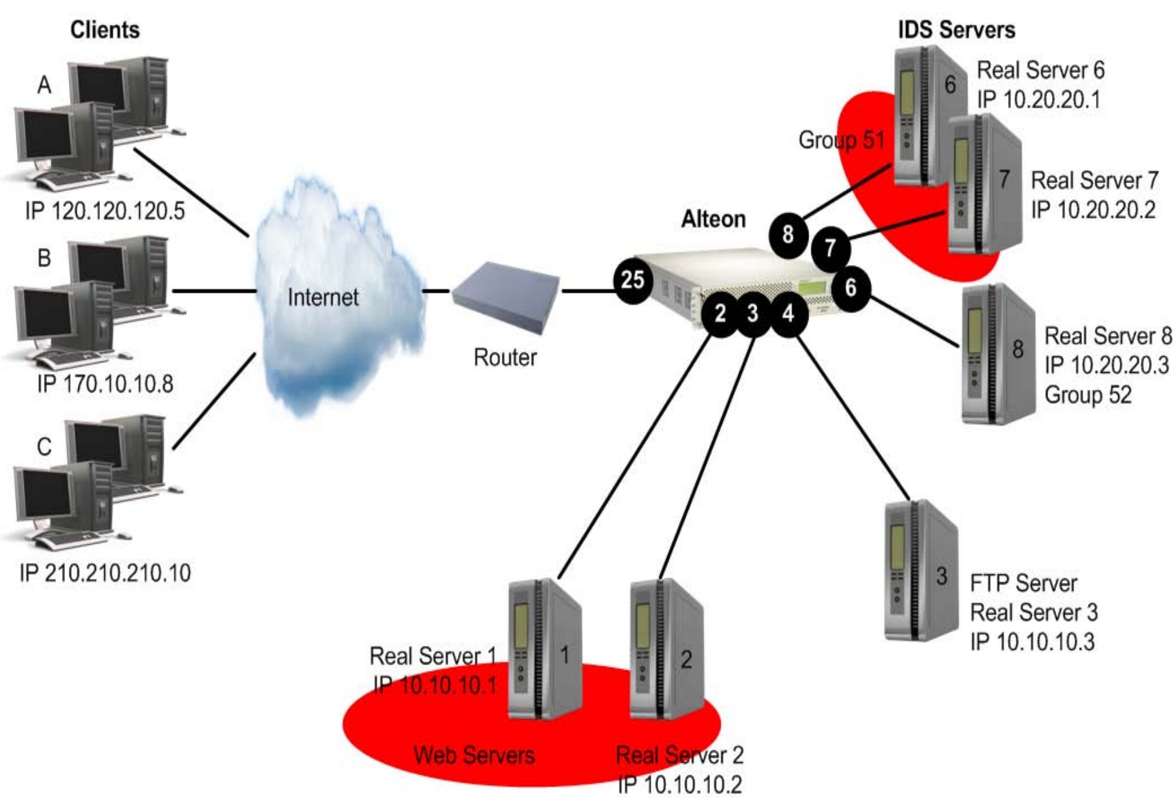
12. Configure Alteon 2 to load balance the real servers as described in Server Load Balancing Configuration Basics, page 171.
    - Configure the IP interfaces on Alteon
    - Configure the SLB real servers and add the real servers to the group
    - Configure the virtual IP address
    - Configure the SLB metric
    - Enable SLB

A copy of Layer 4 traffic from clients A, B, and C and from the real servers are directed to the IDS servers and load balanced between IDS servers 6 and 7.

Example 2: Load Balancing to Multiple IDS Groups

Figure 48 - Server Load Balancing and IDS Load Balancing to Multiple Group, page 315 illustrates a single Alteon performing both standard server load balancing and IDS SLB. In this example, two IDS groups are configured: IDS Group 51 is for HTTP traffic only, and IDS Group 52 is for all other traffic.

Figure 48: Server Load Balancing and IDS Load Balancing to Multiple Group



When the same Alteon is configured to load balance real servers and IDS servers, filter processing is not required on the client processing port (port 25). To maintain session persistency, if you add the filter to the client port, Alteon can be configured to hash on both the client IP and virtual server IP. This ensures that both client and server traffic belonging to the same session is sent to the same IDS server. If you do not add the filter on port 25, then Alteon hashes on the client IP address only.

To load balance to multiple IDS groups

1. Set up the IDS servers.

   For information on setting up the IDS servers, see Setting Up IDS Servers, page 311. To configure the IDS servers as real servers you must consider the following:

   — Connecting the IDS servers

   — Choosing the health check

   — Configuring the IP addresses

   For more information on each of the above items, see step 1 on page 310.

2. Configure the IDS servers as real servers.

   In Figure 49 - Server Load Balancing and IDS Load Balancing Across Multiple Alteons, page 319, the IDS servers are set up with non-routable IP addresses. The real servers must be numbered 1 to 63.

`>> # /cfg/slb/real 6/rip 10.20.20.1/ena`	(Specify IP address for IDS Server 6)
`>> # /cfg/slb/real 7/rip 10.20.20.2/ena`	(Specify IP address for IDS Server 7)
`>> # /cfg/slb/real 8/rip 10.20.20.3/ena`	(Specify IP address for IDS Server 8)

3. Create two IDS groups and add the IDS servers. Define the two IDS Groups 51 and 52. The IDS group numbers must be between 1 to 63.

`>> # /cfg/slb/group 51`	(Define a group)
`>>Real Server Group 51# add 6`	(Add IDS Server 6)
`>>Real Server Group 51# add 7`	(Add IDS Server 7)
`>>Real Server Group 51# /cfg/slb/group 52`	(Define another group)
`>>Real Server Group 52# add 8`	(Add IDS Server 8)

4. Define the group metric for the IDS server groups. IDS SLB supports the hash metric only.

`>>Real Server Group 51# metric hash`	(Set the metric to hash)
`>>Real Server Group 51# /cfg/slb/group 52`	(Select the other IDS group)
`>>Real Server Group 52# metric hash`	(Set the metric to hash)

   The hash metric is implemented in two ways. For more information, see step 4 on page 313.

5. Define the health check for the group. Configure ICMP health check for the group.

`>>Real Server Group 51# health icmp`	(Set the health check to ICMP)
`>>Real Server Group 51# /cfg/slb/group 52`	(Select the other IDS group)
`>>Real Server Group 52# health arp`	(Set the health check to ARP)

6. Define the group for IDS SLB.

`>>Real Server Group 51# idslb ena`	(Enable IDS for the IDS server group)
`>>Real Server Group 51# /cfg/slb/group 52`	(Select the other IDS group)
`>>Real Server Group 52# idslb ena`	(Enable IDS for the IDS server group)

7. Select the rport for the IDS group.

`>> # /cfg/slb/group 51`	(Select the IDS group)
`>>Real Server Group 51# idsrprt http`	(Select HTTP traffic for IDS group)
`>>Real Server Group 51# /cfg/slb/group 52`	(Select the IDS group)
`>>Real Server Group 52# idsrprt any`	(Select non-HTTP traffic for IDS group)

8. Enable IDS on the client and server processing ports. This enables frames ingressing the port to be copied to the IDS servers.

`>># /cfg/slb/port 25/idslb ena`	(Enable IDS SLB for port 25)
`>>SLB port 25# /cfg/slb/port 2/idslb ena`	(Enable IDS SLB for port 2)
`>>SLB port 2# /cfg/slb/port 3/idslb ena`	(Enable IDS SLB for port 3)
`>>SLB port 3# /cfg/slb/port 4/idslb ena`	(Enable IDS SLB for port 4)

In addition to enabling IDS at the port level, a filter must be configured to create a session entry for non-SLB frames ingressing the port. IDS load balancing requires a session entry to be created to store the information regarding to which IDS server to send traffic.

9. Create an allow filter and configure the filter with the idshash metric.

`>> # /cfg/slb/filt 2048`	(Select the menu for Filter 2048)
`>> Filter 2048# sip any`	(From any source IP address)
`>> Filter 2048# dip any`	(To any destination IP address)
`>> Filter 2048# action allow`	(Allow matching traffic to pass)
`>> Filter 2048# ena`	(Enable the filter)
`>> Filter 2048# adv/idshash both`	(Set the hash metric parameter)

The IDS hash metric is set to hash on both the source and destination IP addresses. Hashing on both source and destination IP address ensures that the returning traffic goes to the same IDS server. By default, the IDS hash metric hashes on the source IP address only.

10. Apply the filter to ports 2, 3, 4 and 25 only. Enable filter processing on all ports that have IDS enabled.

If you add the allow filter to the client port 25, Alteon hashes on the client IP and virtual server IP addresses for both client and server frames. This ensures that both client and server traffic belonging to the same session is sent to the same IDS server. If you do not add the allow filter on port 25, Alteon hashes on the client IP only for client frames and hashes on the client IP and virtual server IP addresses for server frames.

`>> # /cfg/slb/port 2`	(Select the port menu)
`>> SLB Port 2# add 2048`	(Apply the filter to port 2)
`>> SLB Port 2# filt ena`	(Enable the filter)
`>> SLB Port 2# /cfg/slb/port 3`	(Select port 3)
`>> SLB Port 3# add 2048`	(Apply the filter to port 3)
`>> SLB Port 3# filt ena`	(Enable the filter)
`>> SLB Port 3# /cfg/slb/port 4`	(Select port 4)
`>> SLB Port 4# add 2048`	(Apply the filter to port 4)
`>> SLB Port 4# filt ena`	(Enable the filter)
`>> SLB Port 4# /cfg/slb/port 25`	(Select port 25)
`>> SLB Port 25# add 2048`	(Apply the filter to port 25)
`>> SLB Port 25# filt ena`	(Enable the filter)

11. Apply and save your changes.

```
>> SLB Port 25# apply
>> SLB Port 25# save
```

A copy of Layer 4 Web traffic from clients A, B, and C and from the Real Servers 1, 2, and 3 is load balanced between IDS Servers 6 and 7. A copy of all non-HTTP traffic is directed to IDS Server 8.

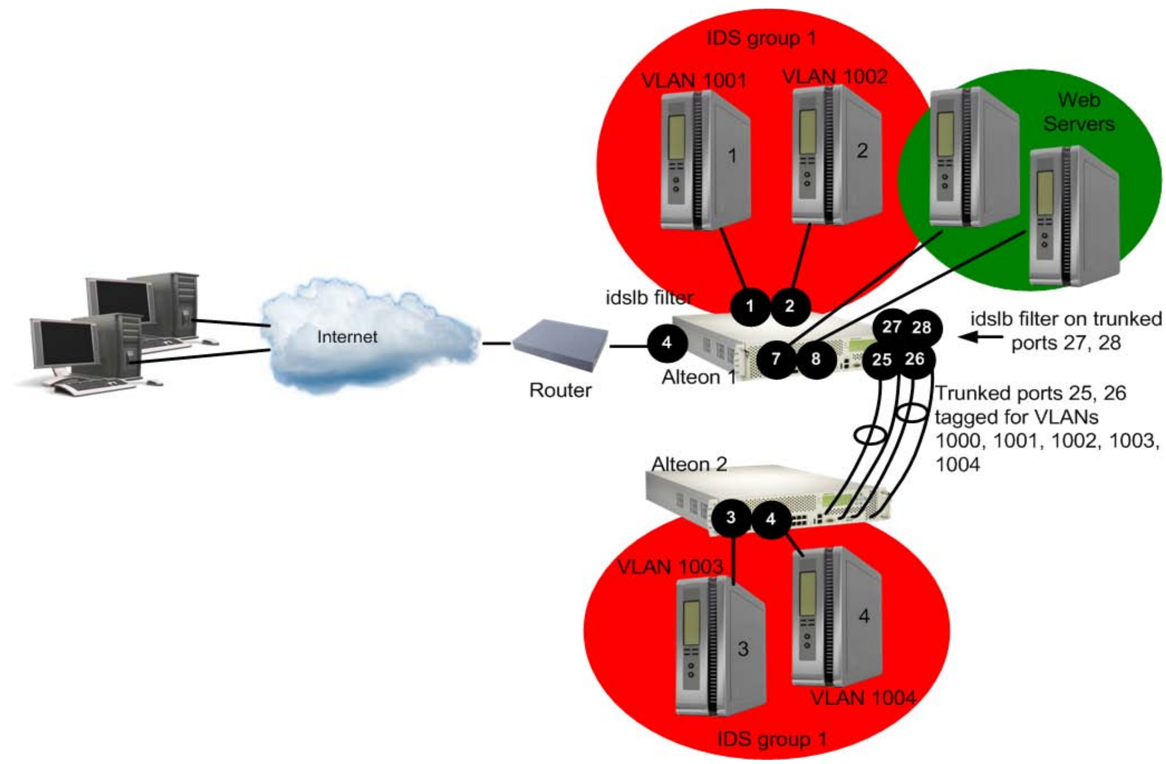
12. Configure Alteon for SLB the real servers as described in Server Load Balancing Configuration Basics, page 171.
    - Configure the IP interfaces on Alteon.
    - Configure and create a group for the SLB real servers.
    - Configure the virtual IP address.
    - Configure the SLB metric.
    - Enable SLB.

Example 3: Load Balancing IDS Servers Across Multiple Alteons

Alteon supports load balancing IDS servers across multiple Alteons in a high availability configuration. By allowing the administrator to disable learning of client and server source MAC addresses over the interswitch link, client request packets are able to reach the real servers when failover occurs.

As illustrated in Figure 49 - Server Load Balancing and IDS Load Balancing Across Multiple Alteons, page 319, the Alteons are connected to each other via a trunked interswitch link (ports 25 and 26) that is associated with all VLANs configured on Alteon. Each Alteon is connected to IDS servers that are each on different VLANs but belong to the same IDS group. For VLAN-based IDS load balancing, the ingress packets are copied by the master Alteon and flooded to the IDS servers for monitoring through the path associated with an IDS VLAN. Since the interswitch link is also associated with this IDS VLAN, the copied packet passes through the interswitch link and is flooded to the IDS VLAN on the standby Alteon.

Figure 49: Server Load Balancing and IDS Load Balancing Across Multiple Alteons



Normally, the standby Alteon learns the source MAC address of clients in the copied packet from the port that is connected to the interswitch link. The standby Alteon also learns the source MAC address of the server when the server response packets enter the master Alteon and are flooded to the IDS VLAN over the interswitch link.

In a high availability configuration, the standby Alteon becomes the master if the current master Alteon fails. The new master Alteon forwards traffic between clients and servers. Because the MAC addresses of the real servers are learned via the interswitch link port, the request packets from clients are forwarded to the interswitch link port on the new master Alteon and are received by the new standby Alteon. Because the standby Alteon does not forward traffic, the request packets do not normally reach the real servers.

Alteon remedies this situation by allowing the administrator to disable learning of client and server source MAC addresses over the interswitch link, thus ensuring that when failover occurs, the client request packets reach the real servers.

To load balance IDS servers across multiple Alteons

1. Set up the IDS servers.

   For information on setting up the IDS servers, see Setting Up IDS Servers, page 311. To configure the IDS servers as real servers you must consider the following:

   — Connecting the IDS servers

   — Choosing the health check (in this case, use the SNMP health check)

   — Configuring the IP addresses

   For more information on each of the above items, see step 1 on page 310.

2. On the master Alteon, configure the interswitch link ports/VLANs for the IDS VLAN.

```
/cfg/port 25/tag ena/pvid 1000
/cfg/port 26/tag ena/pvid 1000
```

3. Configure trunk groups.

```
/cfg/l2/trunk 1/ena/add 25/add 26          (Add ports 25, 26 to Trunk Group 1)
/cfg/l2/trunk 2/ena/add 27/add 28          (Add ports 27, 28 to Trunk Group 2)
```

4. Configure an IP interface for the SNMP health check to the other Alteon.

```
/cfg/l3/if 3/addr 11.11.11.1/mask 255.255.255.255/vlan 1000
```

5. Configure VLANs. Disable source MAC address learning only on the IDS VLANs.

   The following VLANS are configured on Alteon:

   — VLAN 1—For load balancing traffic to the real servers
   — VLAN 1000—For performing SNMP health checks on Alteon 2
   — VLAN 1001—For IDS Server 1
   — VLAN 1002—For IDS Server 2
   — VLAN 1003—For IDS Server 3
   — VLAN 1004—For IDS Server 4

```
>> Main# /cfg/l2/vlan 1001/ena
>> VLAN 1001# learn dis                 (Disable source MAC learning on the IDS
:                                       VLAN)
>> VLAN 1001# add 25/add 26             (Set port members of the VLAN)

Port 25 is an UNTAGGED port and its current PVID is 1.
Confirm changing PVID from 1 to 1001 [y/n]: y
Port 26 is an UNTAGGED port and its current PVID is 1.
Confirm changing PVID from 1 to 1001 [y/n]: y
>> Layer 2# /cfg/l2/vlan 1001/ena/learn dis/add 25/add 26
>> Layer 2# /cfg/l2/vlan 1002/ena/learn dis/add 25/add 26
>> Layer 2# /cfg/l2/vlan 1003/ena/learn dis/add 25/add 26
>> Layer 2# /cfg/l2/vlan 1004/ena/learn dis/add 25/add 26
```

6. Configure the IDS servers as real servers.

   In Figure 49 - Server Load Balancing and IDS Load Balancing Across Multiple Alteons, page 319, the IDS servers are set up with non-routable IP addresses. The real servers must be numbered between 1 and 63. SNMP health checks are configured to check the status of the ports on Alteon 2 that are connected to the IDS servers.

```
>> # /cfg/slb/real 1/rip 11.11.11.1/ena       (Set IP address for IDS Server 1)
>> Real server 1 # ids/idsvlan 1001           (Set IDS VLAN for IDS Server 1)
>> Real Server 1 IDS# idsport 25              (Set IDS VLAN port)
>> Real Server 1 IDS# oid 1.3.6.1.2.1.2.2.1.8.257
                                              (Set OID to health check port 1)
>> # /cfg/slb/real 2/rip 11.11.11.1/ena
>> Real server 2 # ids/idsvlan 1002
```

`>> Real Server 2 IDS# idsport 25`	
`>> Real Server 2 IDS# oid 1.3.6.1.2.1.2.2.1.8.258`	(Set OID to health check port 2)
`>> # /cfg/slb/real 3/rip 11.11.11.100/ena`	(Set the IP interface for Alteon 2)
`>> Real server 3 # ids/idsvlan 1003`	
`>> Real Server 3 IDS# idsport 25`	
`>> Real Server 3 IDS# oid 1.3.6.1.2.1.2.2.1.8.259`	(Set OID to health check port 3 on Alteon 2)
`>> # /cfg/slb/real 4/rip 11.11.11.100/ena`	
`>> Real server 4 # ids/idsvlan 1004`	
`>> Real Server 4 IDS# idsport 25`	
`>> Real Server 4 IDS# oid 1.3.6.1.2.1.2.2.1.8.260`	(Set OID to health check port 4 on Alteon 2)

7. Create an IDS group and add the IDS servers. Define the IDS group. The IDS group numbers must be between 1 to 63.

`>> # /cfg/slb/group 53`	(Define a group)
`>>Real Server Group 53# add 1`	(Add IDS Server 1)
`>>Real Server Group 53# add 2`	(Add IDS Server 2)
`>>Real Server Group 53# add 3`	(Add IDS Server 3)
`>>Real Server Group 53# add 4`	(Add IDS Server 4)

8. Define the group metric for the IDS server group. IDS SLB supports the hash metric only.

```
>>Real Server Group 53# metric hash
```

9. Define the health check for the group.

```
>>Real Server Group 50# health snmp1
```

10. Define the group for IDS SLB.

```
>>Real Server Group 50# ids ena
```

11. Select the rport for the IDS group.

```
>>Real Server Group 50# idsrprt 80
```

12. Enable IDS on the client and server ports. This enables frames ingressing the traffic ports to be copied to the IDS servers.

`/cfg/slb/port 4/ids ena`	(Enable IDS processing for port 4)
`>>SLB port 4# /cfg/slb/port 7 ids ena`	(Enable IDS processing for port 7)
`>>SLB port 7# /cfg/slb/port 8 ids ena`	(Enable IDS processing for port 8)

`>>SLB port 7# /cfg/slb/port 27/ids ena`	(Enable IDS processing for port 27)
`>>SLB port 27# /cfg/slb/port 28/ids ena`	(Enable IDS processing for port 28)

In addition to enabling IDS at the port level, a filter must be configured to create a session entry for non-SLB frames ingressing the port. IDS load balancing requires a session entry to be created to store the information regarding to which IDS server to send traffic.

13. Configure an integer value for Alteon to accept the SNMP health check.

    If the value returned by the real server for the MIB variable does not match the expected value configured in the rcvcnt field, then the server is marked down. The server is marked back up when it returns the expected value.

    In this step, the server is marked down if Alteon receives a value of 1. The real server is considers the health check to have failed.

```
>>SLB port 27# /cfg/slb/advhc/snmphc 1/rcvcnt "1"
```

14. Create an allow filter and configure the filter with the idshash metric.

    The IDS hash metric is set to hash on both the source and destination IP addresses. Hashing on both source and destination IP address ensures that the returning traffic goes to the same IDS server. If the port is configured for client processing only, then Alteon hashes on the source IP address. By default, the IDS hash metric hashes on the source IP address only.

15. Apply the allow filter to ports 4, 7, 8, 27, and 28 to enable filter processing on all ports that have IDS enabled.

    If you add the allow filter to the client port 4, Alteon hashes on the client IP and virtual server IP address for both the client and server frames. This ensures that both client and server traffic belonging to the same session is sent to the same IDS server. If you do not add the allow filter on port 5, then Alteon hashes on the client IP only for client frames and hashes on the client IP and virtual server IP addresses for server frames. The allow filter must be applied on all ports that require Layer 4 traffic to be routed to the IDS servers.

`>> Filter 2048# /cfg/slb/port 4`	(Select the client port)
`>> SLB Port 4# add 2048`	(Apply the filter to the IDS port)
`>> SLB Port 4# filt ena`	(Enable the filter)
`>> SLB Port 4# /cfg/slb/port 7`	(Select the IDS Server 7 port)
`>> SLB Port 7# add 2048`	(Apply the filter to the IDS port)
`>> SLB Port 7# filt ena`	(Enable the filter)
`>> SLB Port 7# /cfg/slb/port 8`	(Select the IDS Server 8 port)
`>> SLB Port 2# add 2048`	(Apply the filter to the client port)
`>> SLB Port 2# filt ena`	(Enable the filter)
`>> SLB Port 2# /cfg/slb/port 27`	(Select the interswitch link for IDS)
`>> SLB Port 27# add 2048`	(Apply the filter to traffic port 27)
`>> SLB Port 27# filt ena`	(Enable the filter)
`>> SLB Port 27# /cfg/slb/port 28`	(Select the interswitch link for IDS)
`>> SLB Port 28# add 2048`	(Apply the filter to traffic port 28)
`>> SLB Port 28# filt ena`	(Enable the filter)

All ingressing traffic at these ports that match any of the filters configured for that port are load balanced to the IDS groups. The allow filter is used at the end of the filter list to make sure that all traffic matches a filter. A **deny all** filter could also be used as the final filter instead of an **allow all** filter.

16. Apply and save your changes.

```
>> SLB Port 26# apply
>> SLB Port 26# save
```

17. Configure Alteon 2 to load balance the real servers as described in Server Load Balancing Configuration Basics, page 171.
    — Configure the IP interfaces on Alteon.
    — Configure the SLB real servers and add the real servers to the group.
    — Configure the virtual IP address.
    — Configure the SLB metric.
    — Enable SLB.

Session Initiation Protocol (SIP) Server Load Balancing

The Session Initiation Protocol (SIP) is an application-level control (signaling) protocol for Internet multimedia conferencing, telephony, event notification, and instant messaging. The protocol initiates call setup, routing, authentication and other feature messages to end-points within an IP domain.

The SIP protocol is used to

- locate users—where the caller and called parties are located.
- determine user capability—what type of protocol (TCP or UDP) and other capabilities the user can support.
- determine user availability, call setup—how to create the call.
- determine call handling—how to keep the call up and how to bring down the call.

This feature load balances SIP proxy servers such as Nortel MCS (Multimedia Communications Server) and TCP-based implementations like Microsoft OCS.

SIP Processing on Alteon

SIP over UDP processing provides the capability to scan and hash calls based on a SIP Call-ID header to a SIP server. The Call-ID uniquely identifies a specific SIP session. Future messages from the same Call-ID is switched to the same SIP server. This involves stateful inspection of SIP messages.

SIP is a text-based protocol with header lines preceding the content. Like HTTP, the first header line has the method specification followed by other header lines that specify other parameters like Call-ID, and so on.

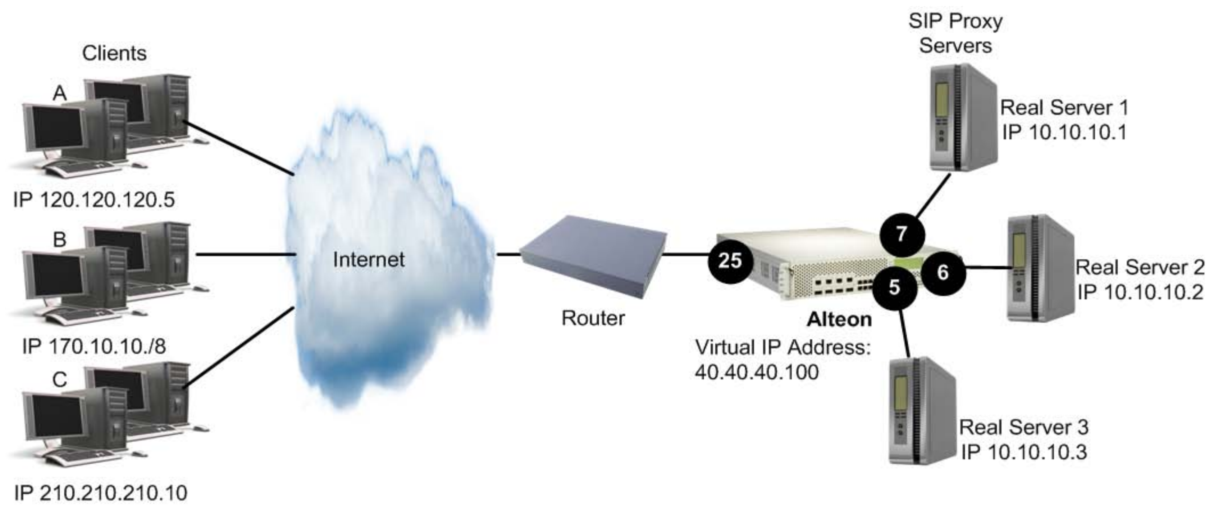
TCP-Based SIP Servers

Alteon supports TCP-based load balancing for SIP and TLS for services such as Microsoft Office Communication Services (OCS) R1 and R2, and the Nortel Multimedia Communication Server (MCS). Microsoft-approved OCS load balancing for both consolidated and expanded topologies enables support for up to 125,000 telephony users.

Configuring SIP Server Load Balancing

Figure 50 - SIP Load Balancing, page 324 illustrates an Alteon performing TCP-based SIP SLB. In this example, three SIP proxy servers are configured in a Real Server Group 100. Alteon is configured for SIP service (port 5060) for virtual server 40.40.40.100.

Figure 50: SIP Load Balancing



To configure SIP load balancing

1. Before you start configuring SIP load balancing:
   — Connect each SIP proxy server to Alteon
   — Configure the IP addresses on all devices connected to Alteon
   — Configure the IP interfaces on Alteon
   — Enable Direct Access Mode (DAM)
   — Disable proxy IP addressing
2. Enable server load balancing.

```
>> # /cfg/slb/on
```

3. Configure IP addresses for the SIP proxy servers.

```
>> # /cfg/slb/real 1/rip 10.10.10.1          (Define address for MCS 1)
>> Real server 1# ena                        (Enable Real Server 1)
>> # /cfg/slb/real 2/rip 10.10.10.2          (Define address for MCS 2)
```

`>> Real server 2# ena`	(Enable Real Server 2)
`>> # /cfg/slb/real 3/rip 10.10.10.3`	(Define address for MCS 3)
`>> Real server 3# ena`	(Enable Real Server 3)

4. Create a group to load balance the SIP proxy servers.

`>> # /cfg/slb/group 100`	(Define a group)
`>>Real Server Group 100# add 1`	(Add Real Server 1)
`>>Real Server Group 100# add 2`	(Add Real Server 2)
`>>Real Server Group 100# add 3`	(Add Real Server 3)

5. Define the group metric for the server group. TCP-based SIP load balancing supports all metrics. For example, set the metric to minmisses.

```
>>Real Server Group 100# metric minmiss
```

6. Define the health check for the group. The health check is TCP for TCP-based SIP load balancing.

```
>>Real Server Group 100# health tcp
```

7. Configure a virtual server for Layer 4 SIP load balancing.

`>> # /cfg/slb/virt 1`	(Select Virtual Server 1)
`>>Virtual Server 1# vip 40.40.40.100`	(Set IP address for the virtual server)
`>>Virtual Server 1# virt ena`	(Enable virtual server)

8. Configure the SIP service 5060 for Virtual Server 1.

`>> # /cfg/slb/virt 1/service 5060`	(Add the SIP service for Virtual Server 1)
`>> # /cfg/slb/virt 1/service 5060 Group 100`	(Add the real server group to the service)

9. Configure the SIP TLS service 5061 for Virtual Server 1.

```
>> # /cfg/slb/virt 1/service 5061/Group 100
```

10. Enable DAM.

```
>> # /cfg/slb/adv/direct ena
```

Note: Distribution of sessions between servers is achieved using the **minmisses** hash method and is not always even. Call distribution can be improved by increasing the number of Call ID bytes that are used as input to the hash function. For example:

```
>> Virtual Server 1 sip Service# sip/hashlen 16
```

11. Increase the timeout for idle sessions.

Note: SIP sessions are quite long and data may be flowing while the signaling path is idle. Because Alteon resides in the signaling path, Radware recommends increasing the real server session timeout value to 30 minutes (Default: 10 minutes).

`>> # /cfg/slb/real 1/tmout 30`	(Increase Real 1 session timeout)
`>> # /cfg/slb/real 2/tmout 30`	(Increase Real 2 session timeout)
`>> # /cfg/slb/real 3/tmout 30`	(Increase Real 3 session timeout)

12. Configure the virtual service for RPC load balancing.

```
>> /cfg/slb/virt/service 135
>>Virtual Server 1 135 service #group 1
```

13. Enable server and client processing at the port level.

`>> # /cfg/slb/port 25`	(Select the client port)
`>>SLB port 25# client ena`	(Enable client processing)
`>>SLB port 25# /cfg/slb/port 5`	(Select the server port)
`>>SLB port 5# server ena`	(Enable server processing)
`>>SLB port 5# /cfg/slb/port 6`	(Select the server port)
`>>SLB port 6# server ena`	(Enable server processing)
`>>SLB port 6# /cfg/slb/port 7`	(Select the server port)
`>>SLB port 7# server ena`	(Enable server processing)

14. Apply and save your changes.

```
>> SLB port 7# apply
>> SLB port 7# save
```

UDP-Based SIP servers

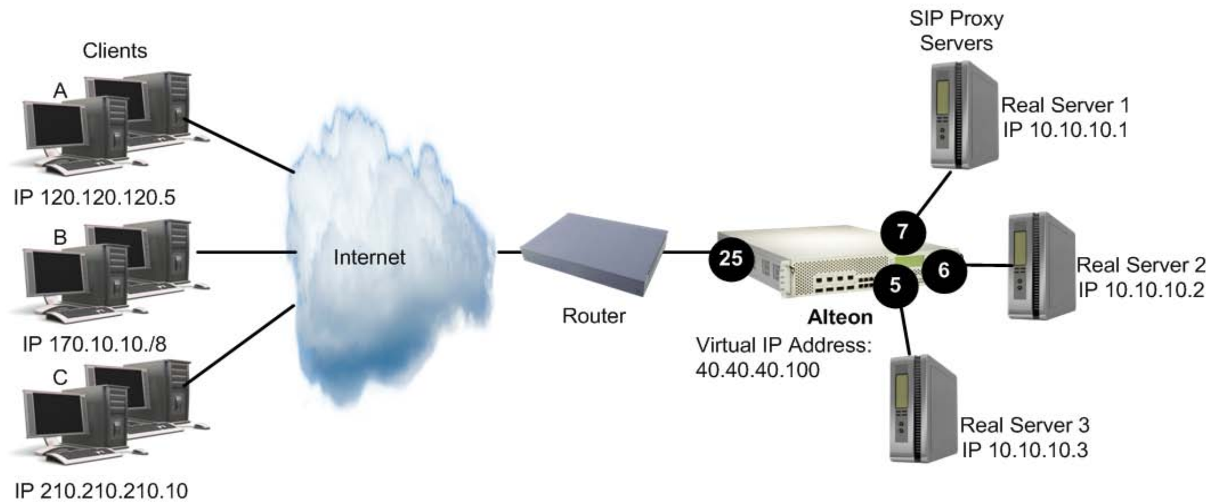
SIP processing provides the capability to scan and hash calls based on a SIP Call-ID header to a SIP server. The Call-ID uniquely identifies a specific SIP session. Future messages from the same Call-ID are switched to the same SIP server. This involves stateful inspection of SIP messages.

SIP is a text-based protocol with header lines preceding the content. Like HTTP, the first header line has the method specification followed by the other header lines that specify other parameters like Call-ID, and so on.

Configuring SIP Server Load Balancing

Figure 51 - SIP Load Balancing Configuration Example, page 327 illustrates an Alteon performing UDP-based SIP SLB. In this example, three SIP proxy servers are configured in a Real Server Group 100. Alteon is configured for SIP service (port 5060) for virtual server 40.40.40.100.

Figure 51: SIP Load Balancing Configuration Example



To configure SIP load balancing

1. Before you start configuring SIP load balancing:
   - Connect each SIP proxy server to Alteon
   - Configure the IP addresses on all devices connected to Alteon
   - Configure the IP interfaces on Alteon
   - Enable Direct Access Mode (DAM)
   - Disable proxy IP addressing
2. Enable server load balancing.

```
>> # /cfg/slb/on
```

3. Configure IP addresses for the SIP proxy servers.

```
>> # /cfg/slb/real 1/rip 10.10.10.1        (Define address for MCS 1)
>> Real server 1# ena                      (Enable Real Server 1)
>> # /cfg/slb/real 2/rip 10.10.10.2        (Define address for MCS 2)
>> Real server 2# ena                      (Enable Real Server 2)
>> # /cfg/slb/real 3/rip 10.10.10.3        (Define address for MCS 3)
>> Real server 3# ena                      (Enable Real Server 3)
```

4. Create a group to load balance the SIP proxy servers.

```
>> # /cfg/slb/group 100                    (Define a group)
>>Real Server Group 100# add 1             (Add Real Server 1)
>>Real Server Group 100# add 2             (Add Real Server 2)
>>Real Server Group 100# add 3             (Add Real Server 3)
```

5. Define the group metric for the server group. Because SIP load balancing is performed based on Call-ID, only the minmisses metric is supported to ensure persistency.

```
>>Real Server Group 100# metric minmiss
```

6. Define the health check for the group. Configure SIP PING request health check which is specifically developed for SIP-enabled servers.

```
>>Real Server Group 100# health sip
```

7. Configure a virtual server for Layer 4 SIP load balancing.

`>> # /cfg/slb/virt 1`	(Select Virtual Server 1)
`>>Virtual Server 1# vip 40.40.40.100`	(Set IP address for the virtual server)
`>>Virtual Server 1# virt ena`	(Enable virtual server)

8. Configure the SIP service 5060 for Virtual Server 1.

`>>> # /cfg/slb/virt 1/service 5060`	(Add the SIP service for Virtual Server 1)
`>> # /cfg/slb/virt  1/service 5060 Group  100`	(Add the real server group to the service)

9. Enable SIP SLB.

```
>>Virtual Server 1 sip Service # sip/sip ena
```

10. Enable DAM.

```
>>Virtual Server 1 sip Service # direct ena
```

11. Enable UDP load balancing
12. Increase the timeout for idle sessions.

Note: SIP sessions are quite long and data may be flowing while the signaling path is idle. Because Alteon resides in the signaling path, Radware recommends increasing the real server session timeout value to 30 minutes (Default: 10 minutes).

When the call terminates with a BYE command, Alteon releases the session entry immediately.

`>> # /cfg/slb/real 1/tmout 30`	(Increase Real 1 session timeout)
`>> # /cfg/slb/real 2/tmout 30`	(Increase Real 2 session timeout)
`>> # /cfg/slb/real 3/tmout 30`	(Increase Real 3 session timeout)

13. Enable server and client processing at the port level.

`>> # /cfg/slb/port 25`	(Select the client port)
`>>SLB port 25# client ena`	(Enable client processing)
`>>SLB port 25# /cfg/slb/port 5`	(Select the server port)
`>>SLB port 5# server ena`	(Enable server processing)

`>>SLB port 5# /cfg/slb/port 6`	(Select the server port)
`>>SLB port 6# server ena`	(Enable server processing)
`>>SLB port 6# /cfg/slb/port 7`	(Select the server port)
`>>SLB port 7# server ena`	(Enable server processing)

14. Apply and save your changes.

```
>> SLB port 7# apply
>> SLB port 7# save
```

Enhancements to SIP Server Load Balancing

Alteon supports the following enhancements to SIP SLB:

- **User-defined SIP port**—Lets you modify the SIP port (in earlier versions, the SIP port was supported on UDP 5060 only).

  To define the SIP port, enter the command:

```
>> Main# /cfg/slb/virt <Virtual Server> /service 5060/rport <Port>
```

- **Session persistency using the refer method**—The refer method of load balancing SIP servers is required for *call transfer* services. The refer method indicates that the recipient should contact a third party using the contact information provided in the request.

  To maintain session persistency, the new request from the recipient to the third party should also hash the same real server. To maintain persistency, whenever Alteon receives a session configured for the refer method, Alteon creates a persistent session.

  When creating a session for a new request, Alteon looks up the session table and selects the correct real server. If there is a persistent session, then the real server specified in the session entry is used if that real server is up. Otherwise, the normal minmiss method is used to select the real server.
- **Supports standard health check options**—Alteon supports the standard method to health check SIP servers. The options method (like HTTP and RTSP) is supported by all RFC 3261 compliant proxies.

  Alteon sends an **options** request to the SIP server when configured to use the SIP options health check. If the response from the server is a “200 OK”, then the server is operational. Otherwise, the server is marked down.
- **Translating the the source port in SIP responses**—Alteon supports the translation of the source port to the application port before forwarding a response to the client in cases where the server uses a source port different to the application port in its response.

```
>> Main# /cfg/slb/sipspat enable
```

To configure the SIP options health check

```
>> Main# /cfg/slb/virt<Virtual Server>/service 5060/rport<Port>
>> Main# /cfg/slb/group <Real Server Group> /health sipoptions
```

- **Support for RTP (SDP) Media Portal NAT**—This feature is useful if you have several media portal servers with private IP addresses. When the proxy servers respond to an INVITE request, the private IP address of the media portal is embedded in the SDP. Alteon translates this private IP address to a public IP address.

  The private media portal address is sent in the response to an INVITE request. Alteon replaces the private IP with the public IP address in the SDP.

To support Media Portal NAT

1. Configure the private to public address mapping

```
>> Main# /cfg/slb/layer7
>> Layer 7 Resource Definition# sdp
>> SDP Mapping# add <private_IP>  <public_IP>
```

2. Enable SDP Media Portal NAT.

```
>> Main# /cfg/slb/virt 1
>> Virtual Server 1# service 14
>> Virtual Server 1 14 Service# sip
>> SIP Load Balancing# sdpnat
```

3. Create static NAT filters.

   This allows RTP traffic destined for the public media portal address to be translated to the actual private media portal address. Create static NAT filters to operate in both directions: one to translate the public address to the private address, and one to translate the private address to the public address.

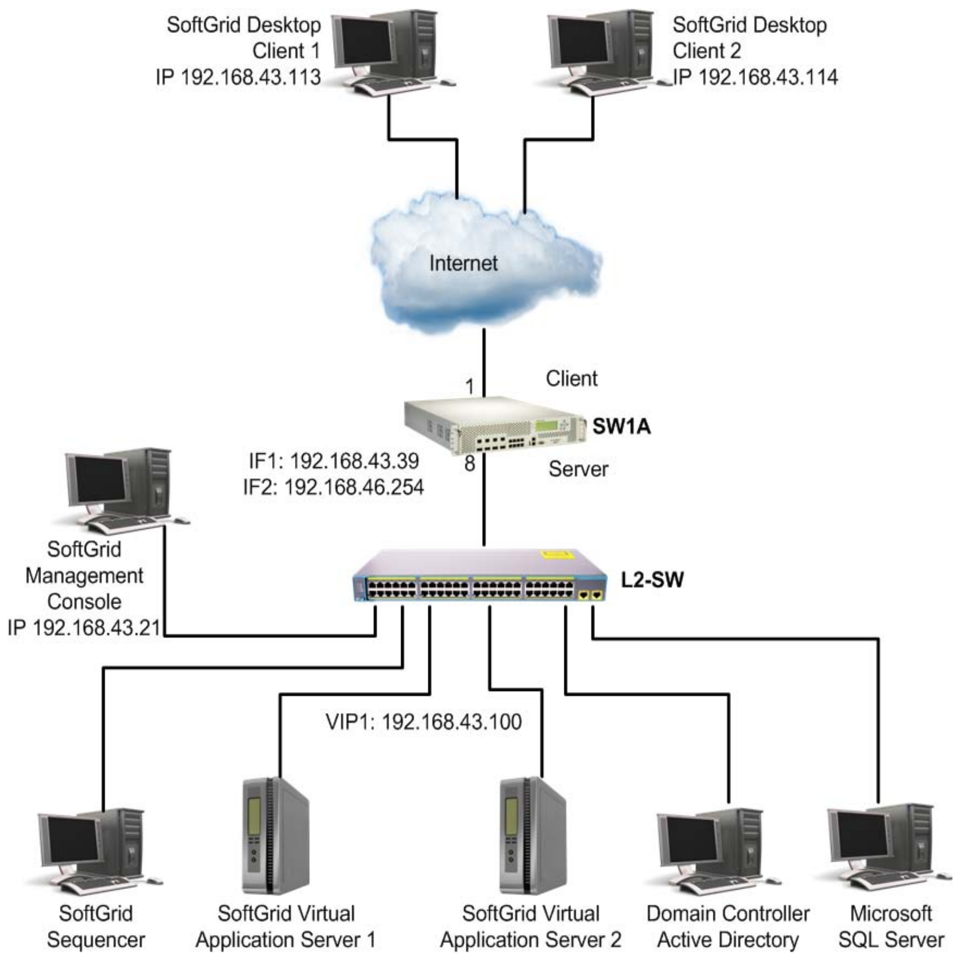
   For more information on static NAT filters, see Network Address Translation, page 384.

SoftGrid Load Balancing

The Softricity SoftGrid platform is used to provide sequenced applications from a SoftGrid Server to a SoftGrid Client. Alteon supports load balancing tailored to the SoftGrid suite for the delivery of sequenced applications and the maintaining of persistency while applications are launched from the SoftGrid Client. Once an application is delivered to the SoftGrid Client, it can be run on the client computer.

Figure 52 - SoftGrid Load Balancing Network Topology, page 331 illustrates an example of a SoftGrid Load Balancing network topology:

Figure 52: SoftGrid Load Balancing Network Topology



The SoftGrid platform supports TCP unicast connections using the following protocols:

1. **Real Time Streaming Protocol (RTSP)**—RTSP is an application-level protocol that is responsible for controlling the transport of multimedia content, session announcements, and tear downs.
2. **Real Time Transport Protocol (RTP)**—RTP is used to transport the application data between the server and the client.
3. **Real Time Control Protocol (RTCP)**—RTCP is used to control the streaming of the application data that is transported by RTP.

   The SoftGrid platform uses three channels to complete the application delivery process. Initially, the SoftGrid Client uses the RTSP channel to create a connection with the SoftGrid Server. The SoftGrid Server then opens two ports for the RTP and RTCP channels and sends these port numbers to the client. The client then opens TCP connections to the ports created on the server. The requested application is then streamed over the RTP channel while the RTCP channel provides control over the RTP channel.

Configuring SoftGrid Load Balancing

The following procedure is an example configuration for SoftGrid SLB.

To configure the SoftGrid load balancing

1. Configure a hostname for the virtual IP address on the DNS server.

   Note: This step is performed on the *network domain controller*.

   Make an entry in the network domain controller for the SoftGrid Server. For example, **<sw_name> 10.10.10.10**. where **<sw_name>** was configured on Alteon using the command **cfg/slb/virt 1/vname <sw_name>**.

2. Edit the SoftGrid Server OSD files.

   When the SoftGrid platform is set up for load balancing, change the .OSD files in the SoftGrid Servers to point to the Alteon virtual IP address or virtual server name:

```
rtsp:// <Device VIP> :554/DefaultApp.sft OR
rtsp:// <Device Virtual NAME> :554/DefaultApp.sft
```

3. Enable SoftGrid load balancing.

```
>> Main# /cfg/slb/virt <virtual server number> /service rtsp/softgrid enable
```

   If SoftGrid is enabled for an RTSP service, the SoftGrid RTSP mode performs the RTSP SLB for that service.

Workload Manager (WLM) Support

Alteon supports the Server/Application State Protocol (SASP) used by the Enterprise Workload Management (WLM) tool. This section includes the following topics:

- How Alteon Works with the DM, page 333
- Configuring WLM Support, page 333
- Verifying WLM Configurations, page 334
- Limitations for WLM Support, page 336

This feature is used to monitor server resources and provide additional input on load balancing decisions. WLM takes into account a server's CPU, storage capacity, and network traffic in any final weighting decisions. WLM uses an implementation of the SASP protocol to perform this task.

The WLM software developed by IBM lets you specify end-to-end performance goals for distributed requests. WLM runs on an entity responsible for reporting or managing a group of members. This entity is known as the Domain Manager (DM). The DM recommends a weight for each application or server in the group. This weight recommendation is based on the business importance, topology, and ability of the system to meet its business goals. This recommended weight helps Alteon make intelligent SLB decisions.

Alteon also supports WLM in the redirect filter environment. The SASP protocol enables Alteon to

- receive traffic weight recommendations from the DM
- register Alteon members with the DM
- enable the Generic Window Manager (GWM) to propose new group members to Alteon

How Alteon Works with the DM

Alteon initiates a TCP connection with the GWM for all the configured IP address and port numbers. After establishing the connection, Alteon registers various WLM-configured groups of real servers with the GWM.

When using application load balancing, the representation of a member is the real server's IP address and the application's port and protocol. When the members are registered, the GWM starts monitoring and computes the weight. The DM periodically sends the weights for all the registered groups.

When a real server is disabled or enabled operationally, Alteon sends a request to temporarily remove the server from the weight calculation.

Configuring WLM Support

Before you start configuring for WLM support, ensure you have configured the following for all the groups and real servers participating in dynamic weights with WorkLoad Managers (WLM):

- Alteon name (`/cfg/sys/ssnmp/name`)
- group name (`/cfg/slb/group #/name`)
- real server names (`/cfg/slb/real #/name`)

Note: You can configure up to 16 Work Load Managers (WLM).

To configure WLM support

1. Configure the IP address and the TCP port number to connect to the WLM.

```
>> Main# /cfg/slb/wlm 11
>> Workload Manager 1# addr 10.10.10.10        (IP address of the WLM)
>> Workload Manager 1# port 10                 (TCP port to connect to the WLM)
```

2. Assign the WLM number to the server or application group.

```
>> Main# /cfg/slb/group 2
>> Real Server Group 1# wlm 11
>> Default gateway 1# apply
```

   If the WLM number is not specified, the group is not registered with the WLM. As a result, dynamic weights are not used for this group.

3. Specify if the WLM should use data or management port.

```
>> Main# /cfg/sys/mmgmt
>> Management Port# wlm mgmt
```

4. Apply and save the configuration.

```
>> Management Port# apply
>> Management Port# save
```

Verifying WLM Configurations

The following are example commands to display and verify WLM configurations.

To display WLM information

```
>> Main# /info/slb/wlm
Workload Manager Information:
ID IP address        Port       State
 1 47.81.25.66       3860       Connected
10 47.80.23.245      3860       Not Connected
```

To display statistics on Work Load Manager 11

```
 >> Main# /stats/slb/wlm 11
Workload Manager 11 Statistics:
Registration Requests:                          1
Registration Replies:                           1
Registration Reply Errors:                      0

Deregisteration Requests:                       1
Deregisteration Replies:                        1
Deregisteration Reply Errors:                   0

Set LB State Requests:                          1
Set LB State Replies:                           1
Set LB State Reply Errors:                      0

Set Member State Requests:                      0
Set Member State Replies:                       0
Set Member State Reply Errors:                  0

Send Weights Messages received:                47
Send Weights Message Parse Errors:              0
Total Messages with Invalid LB Name:            0
Total Messages with Invalid Group Name:         0
Total Messages with Invalid Real Server Name:   0
Messages with Invalid SASP Header:              0
Messages with parse errors:                     0
Messages with Unsupported Message Type:         0
```

To display weights updates for the WLM-configured group

```
>> Main# /stats/slb/group 2
Real server group 2 stats:

Total weight updates from WorkLoad Manager : 10

                  Current    Total  Highest
Real IP address Sessions Sessions Sessions  Octets
---- ---------- -------- -------- --------  ------
  1  1.1.1.1    0        0        0         0
  2  2.2.2.2    0        0        0         0
  3  3.3.3.3    0        0        0         0
  4  4.4.4.4    0        0        0         0
---- ---------- -------- -------- --------  -----
     group 2    0        0        0         0
```

To display the current weight for the real servers for a particular service for application load balancing

Note: The WLM-assigned weights are displayed as dynamic weight.

```
>> Main# /info/slb
>> Server Load Balancing Information# virt 1
  1: 10.10.7.1,       00:01:81:2e:a0:8e
    virtual ports:
     http: rport http, group 1, backup none, slowstart
         real servers:
           1: 192.168.2.11, backup none, 0 ms, group ena, up
              dynamic weight 20
           2: 192.168.2.12, backup none, 0 ms, group ena, up
              dynamic weight 40
```

To display the current weight for the real server for application redirection

```
>> Main# /info/slb
>> Server Load Balancing Information# filt 224
 224:  action allow
    group 1, health 3, backup none, vlan any, content web.gif
    thash auto, idsgrp 1
    proxy: enabled
    layer 7 parse all packets: enabled
    real servers:
       1: 192.168.2.11, backup none, 0 ms, group ena, up
         dynamic weight 40
```

To clear WLM SASP statistics

```
>> Main# /stats/slb/wlm <#> clear
```

Limitations for WLM Support

Alteon does not support the following:

- SASP over SSL.
- Real server weights per service. If multiple services are configured to use the same group, then changing the weight for one service affects the weight of real server for all other services.
- Workload manager de-registration after a Layer2 or Layer 3 change. If you make any changes to the VLAN or IP Interface as the eWLM environment, then WLM de-registration updates are sent to all the DMs.
- Workload manager de-registration after an SLB change. WLM de-registration is sent to all DMs after an SLB update.

Chapter 14 – Offloading SSL Encryption and Authentication

Secure Sockets Layer (SSL) is a security layer that can be added to various communication protocols in order to serve four main purposes that contribute together to establishing a secure communication channel.

This chapter discusses the Alteon SSL offloading capabilities which performs encryption, decryption, and verification of Secure Sockets Layer (SSL) transmissions between clients and servers, relieving the back-end servers of this task. This enables the back-end servers to maximize their performance and efficiency, resulting in faster server response times and increased server capacity to handle more concurrent users.

SSL encryption and authentication includes the following characteristics:

- **Authentication—**Each communicating partner should be able to verify that the other is who it claims to be and not an impostor.
- **Privacy—**A third party should not be able to eavesdrop on a private communication.
- **Integrity—**The protocol should or easily detect any tampering with the transmission.
- **Non-repudiation—**Senders should not be able to claim that they did not send what the receiver received.

The chapter includes the following sections:

SSL Offloading Implementation

For Alteon to provide SSL offloading, you must configure, enable, and apply the following components:

- **SSL Virtual Service—**As discussed in SSL Offloading Implementation, page 337, you must define an HTTPS or SSL virtual service and associate to it both an SSL server certificate, and an SSL policy that governs the behavior of the SSL virtual service.
- **SSL Policy—**As discussed in SSL Policies, page 338, you must define an SSL policy and associate it to the SSL virtual service. An SSL policy includes the definition of the ciphers that enable SSL handshaking, as well as the type of traffic that is sent to the back-end servers.

  An single SSL policy can be reused across multiple virtual services.
- **Certificate Repository—**As discussed in Certificate Repository, page 338, you must supply a server certificate that you associate with the SSL virtual service. The server certificate includes the attributes needed to perform SSL handshaking and enable the decryption and encryption of the traffic related to the virtual service.

  You can associate only a single server certificate to a virtual service, but the same server certificate can be used by multiple services.

  You can associate multiple server certificates to a virtual service using Server Name Indication (SNI). With SNI, the browser sends the requested hostname, enabling the server to recognize which certificate to use before an SSL handshake and an actual HTTP request was made. The same server certificate can also be used by multiple services.

- **Client Authentication Policy—**Optionally, you can define a client authentication policy that validates a client's identity as part of the SSL handshake. In addition to defining the client authentication policy, you must associate it to the SSL policy for it to take effect. For more information, see Client Authentication Policies, page 343.

  A single client authentication policy can be reused across multiple SSL policies, and by extension across multiple virtual services.

Note: The order of configuring these components is not important, as long that you eventually enable and apply them all as a unified configuration at one time. This means that you can configure one or more of them individually and then configure the remaining items at a later time.

SSL Policies

An SSL policy determines the behavior of the SSL or HTTPS service that it is associated with. The SSL policy determines the following:

- Which SSL/TLS versions are allowed during handshake
- Which cipher suites are used during handshake and encryption
- Which intermediate Certificate Authority (CA) to use
- Which SSL information to pass to the back-end servers
- When and if to use HTTP protocol-based location redirection conversion from HTTP to HTTPS
- Whether to use back-end encryption
- Whether and how to use client authentication
- Whether to use SSL/TLS on the front-end connection

An single SSL policy can be associated to multiple virtual services if they share the same SSL configuration.

For details on defining the SSL policy parameters, see the section on the **/cfg/slb/ssl/sslpol** menu in the *Alteon Application Switch Operating System Command Reference*.

Note: Alteon lets you explicitly select or deselect supported SSL and TLS protocol versions for the front-end and back-end connections.

Certificate Repository

Certificates are digitally signed indicators that identify a server or a user. They are usually provided in the form of an electronic key or value. The digital certificate represents the certification of an individual business or organizational public key but can also be used to show the privileges and roles for which the holder has been certified. It also includes information from a third-party verifying identity. Authentication is needed to ensure that users in a communication or transaction are who they claim to be.

A basic certificate includes:

- The certificate holder's identity
- The certificate serial number
- The certificate expiry date
- A copy of the certificate holder's public key

- The identity of the Certificate Authority (CA) and its digital signature to affirm the digital certificate was issued by a valid authority.

The certificate repository is a secured stronghold of all PKI-related components such as encryption keys, certificates of different types, and Certificate Signing Requests (CSRs). Certificate components are required for Alteon to supply SSL offloading services and client authentication. Alteon supports the X.509 standard for PKIs.

For details on configuring the components of the certificate repository, see the section on the **/cfg/slb/ssl/certs** menu in the *Alteon Application Switch Operating System Command Reference*.

Certificate Types in the Certificate Repository

The certificate repository may include the following certificate types:

- Server Certificates, page 339
- Intermediate CA Certificates, page 339
- Trusted CA Certificates, page 340

Server Certificates

A server certificate is a type of certificate used to identify servers during SSL handshake. For details on associating server certificates to SSL-based virtual services, see SSL Offloading Implementation, page 337. You either import a pre-existing server certificate using the **/cfg/slb/ssl/certs/import** command, or you can generate your own in Alteon.

When you generate your own server certificate, if an underlying Certificate Signing Request (CSR) and/or key pair do not already exist by the same name as the server certificate, they are generated along with the server certificate. The resulting server certificate is a "self-signed" server certificate, meaning it was issued by the server for itself. This kind of a certificate is good for testing purposes, as real users will experience various warning messages if used for the real SSL service. In order to be used in the real-life SSL environment, the server certificate must be issued (signed) by a Certificate Authority (CA) which is trusted by the client's browsers.

To achieve this, once the certificate's CSR is generated, you must submit it to a trusted Certificate Authority (CA) for signing. If the request is successful, the CA sends back a certificate that has been digitally signed by its own key, which you import using the **/cfg/slb/ssl/certs/import** command, ensuring that it is not imported to the same entity name as the CSR.

Intermediate CA Certificates

Intermediate CA certificates are used when the CA providing the virtual service's server certificate is not directly trusted by the end-user's Web browsers. This is typical in an organization that has its own CA server for generating server's certificates. In order to construct the trust chain from the user's browser list of trusted CAs to the organization's CA server, an intermediate CA certificate or chain of certificates can be provided.

You can optionally bind an intermediate Certificate Authority (CA) certificate to the SSL policy (see SSL Policies, page 338). These certificates are not created in Alteon—you must first import them. You can also create a group of intermediate certificates (a complete CA chain) and bind it to the SSL policy.

For details on associating an Intermediate CA certificate to an SSL policy, see the section on the **/cfg/slb/ssl/sslpol** menu in the *Alteon Application Switch Operating System Command Reference*.

Trusted CA Certificates

Trusted CA certificates are certificates that come from a Certificate Authority that your organization uses to provide users with certificates (client certificates). Trusted CA certificates are associated with client authentication policies (see Client Authentication Policies, page 343). If you use this option, you must specify the trusted client CA certificate or group of trusted client CA certificates to allow Alteon to know which client certificates to accept.

Trusted CA certificates are not created in Alteon—you must first import them. You select the trusted CA certificates from those you have imported.

For details on associating a trusted CA certificate to a client authentication policy, see the section on the **/cfg/slb/ssl/authpol** menu in the *Alteon Application Switch Operating System Command Reference*.

Importing and Exporting Certificate Components to and from the Repository

You import and export components to and from the certificate repository as described in Table 28 - Import and Export of Certificate Repository Components, page 340. For more information on exporting and importing certificate repository components, see the section on the **/cfg/slb/ssl/certs** menu in the *Alteon Application Switch Operating System Command Reference*.

Table 28: Import and Export of Certificate Repository Components

Component	Export/ Import	Description
Key pair	Export, Import	Key pairs include a private key and public key. The private key is used to decrypt and encrypt the SSL handshake, making it the most sensitive piece of information in the PKI, and should be kept as secure as possible. It is usually exported for backup purposes only. When a key pair is exported, it is encrypted with a one-time passphrase supplied at the time of export. The same passphrase must be supplied during import to allow decrypting of the keys. Public keys construct the other side of the asymmetric encryption key pair and are published as part of the certificate to allow decrypting traffic encrypted by the private key, and vice-versa. Keys are exported in encrypted PEM format. **Note:** The maximum file size for importing SSL components (excluding the 2424-SSL configuration) is 200 KB.
CSR	Export	You export a CSR to a CA to get a trusted CA signature for a server certificate that you want created.
Certificate	Export, Import	Certificates are usually exported for backup purposes. Certificate are exported in PEM format. **Note:** The maximum file size for importing SSL components (excluding the 2424-SSL configuration) is 200 KB.

Table 28: Import and Export of Certificate Repository Components

Component	Export/ Import	Description
Certificate and key	Export, Import	A combined key pair and server certificate. Alteon allows importing and exporting certificates and keys encapsulated into a single PKCS#12 (p12) file. This file is secured by a passphrase that must be supplied during the import or export operation. **Note:** The maximum file size for importing SSL components (excluding 2424-SSL configuration) is 200 KB. See the explanations for certificates and key pairs in this table.
Intermediate CA certificate	Export, Import	Intermediate CA certificates are not created in Alteon—you must first import them. Intermediate CA certificates are usually exported for backup purposes. **Note:** The maximum file size for importing SSL components (excluding 2424-SSL configuration) is 200 KB.
Trusted CA certificate	Export, Import	Trusted CA certificates are not created in Alteon—you must first import them from the CA. Trusted CA certificates are usually exported for backup purposes. **Note:** The maximum file size for importing SSL components (excluding 2424-SSL configuration) is 200 KB.
2424-SSL configuration	Import	If you are migrating your SSL configuration from an Alteon 2424-SSL platform to an Alteon platform running Alteon version 27.0.0.0 or later, you can import the entire 2424-SSL certificates and key pairs repository in a single bulk operation. For detailed procedures on migrating the SSL configuration of an Alteon 2424-SSL platform, refer to *Migrating the SSL Offloading Configuration of the Alteon Application Switch 2424-SSL to AlteonOS version 27.0.0.0*. When importing this configuration, all associated certificates are imported by default, including server certificates, intermediate CA certificates, and trusted CA certificates. Other certificates may also be imported on request. **Note:** This procedure does not transfer the SSL server configuration from the 2424-SSL configuration file.

SSL Server Certificate Renewal Procedure

The SSL server certificate renewal procedure comprised two cases:

1. Renewal of a self-signed server certificate (The certificate was created on the Alteon itself, and the certificate signer (CA) is same as the certificate subject name.)
2. Renewal of a real server certificate signed by a third-party trusted CA.

In both cases, in order to facilitate a timely renewal process, you can track Alteon SNMP alerts. Alteon generates SNMP alerts 30, 15, 10, 5, 4, 3, 2, and 1 day before certificate expiration. Once a certificate has expired a daily alert is issued.

To renew a self-signed certificate

1. Log in over a secure management interface (SSH, HTTPS).
2. Enter the certificate repository (**/cfg/slb/ssl/certs/**) and select the server certificate to be renewed.
3. Select **Generate**.
   Alteon will recognize this as a self-signed certificate (SubjectName=Issuer) and will prompt with:

```
A self-signed server certificate already generated.
Expire:  Sat Nov 10 02:51:59 2013
To renew, enter certificate validation period in days (1-3650) [365]:
```

4. Enter the new validation period.
5. Enter **Apply** and **Save**.

To renew a real server certificate signed by a third-party trusted CA

1. Log in over a secure management interface (SSH, HTTPS).
2. Enter the certificate repository (**/cfg/slb/ssl/certs/**).
3. If the original server certificate was generated on this Alteon platform, then a corresponding Certificate Signing Request (CSR) will exist for it in the certificate repository. Skip to step 5.
4. If there is no existing CSR, create a CSR for the server certificate:
   a. Select the server certificate to be renewed.
   b. Enter **cur** to list all certificate information.
   c. Exit and enter the **Request** menu using the same ID as the to-be-renew server certificate.
   d. Select **Generate** and specify all information as shown for the existing server certificate (from the **cur** command).
5. Export the to-be-renewed server certificate CSR and send it to the third-party CA for signing.
6. When the newly-signed certificate is received from the third-party CA import it to the Alteon platform with the same ID as the existing server certificate.
7. Enter **Apply** and **Save**.

Alternatively you can follow the procedure in example1 for generating a new server certificate, and when completed, replace the associated server certificate in the virtual service. This allows easy roll-back to a previous certificate if needed.

Client Authentication Policies

SSL client authentication enables a server to confirm a client's identity as part of the SSL handshake process. A client's certificate and public ID are checked to be valid and that they were issued by a trusted Certificate Authority (CA). If the certificate is valid, the handshake process is completed, allowing data to be sent to the intended destination. If the certificate is not valid, the session is terminated.

When using SSL offloading, you can optionally define a client authentication policy that authenticates the client’s identity. You associate a client authentication policy to an SSL policy, and the SSL policy, in turn, is associated to a virtual service.

To authenticate the client's identity, you import a CA certificate into Alteon. This CA certificate is used when Alteon receives a client certificate to validate it. By checking that it was generated by this trusted CA. Additionally, you can configure Alteon to ensure that the client certificates were not revoked by checking their statuses using OCSP (Online Certificate Status Protocol).

Following an SSL handshake where client authentication was performed successfully (for example the client provided a valid certificate that identifies it and was issued by the trusted CA), you may want to validate the certificate was not revoked since it was generated. Alteon enables you to perform ad hoc certificate validation using Online Certificate Status Protocol (OCSP).

Note: Certificate validation is using the SSL handshake process, which means the TCP handshake was already completed. This implies that Alteon will open the connection to the back-end server even if the OCSP validation failed.

For details on configuring client authentication policies, see the section on the **/cfg/slb/ssl/authpo**l menu in the *Alteon Application Switch Operating System Command Reference*.

To offload OCSP servers from frequent, repetitive validation requests, Alteon saves OCSP responses in a cache for a defined period of time. In some cases you may want to purge the OCSP cache of OCSP responses. For more details, see the section on the **/oper/slb/ocsppurg** command in the *Alteon Application Switch Operating System Command Reference*.

Common SSL Offloading Service Use Cases

The following are examples of common use cases for configuring an SSL offloading service:

Example 1: Configuring a Basic SSL Offloading Service

1. Before you can configure an SSL offloading service, ensure that Alteon is configured for basic SLB:
   — Assign an IP address to each of the real servers in the server pool.
   — Define an IP interface.
   — Define each real server.

- Assign servers to real server groups.
- Enable SLB.
- Define server port and client port.
- Define virtual server

For more information on how to configure Alteon for SLB, see Server Load Balancing, page 165.

2. Define the SSL Policy which will govern the SSL offloading behavior.

`>> Main# /cfg/slb/ssl/sslpol myPol`	(Define an ID to identify the SSL Policy. The ID may be alphanumeric or numeric.)
`>> SSL Policy myPol# cipher high`	(Select the cipher suite to use during SSL handshake. By default, the RSA cipher suite is selected. Radware recommends using the PCI-DSS pre-configured cipher suite for enhanced SSL security.)
`>> SSL Policy myPol# ena`	(Enable the policy)

For details on defining additional SSL policy parameters, see the section on the **/cfg/slb/ssl/sslpol** menu in the *Alteon Application Switch Operating System Command Reference*.

3. Define a server certificate for this service:
   - Import a third-party signed server certificate. For details on configuring the certificate repository, see the section on the **/cfg/slb/ssl/certs** menu in the *Alteon Application Switch Operating System Command Reference*.
   - Alternatively, generate a self-signed server certificate, as shown in the following example:

```
>> Main# /cfg/slb/ssl/certs/srvrcert MyCert
>> Server certificate MyCert# generate
This operation will generate a self-signed server certificate.
Enter key size [512|1024|2048|4096] | [1024]:
Enter server certificate hash algorithm [md5|sha1|sha256|sha384|sha512] |
[sha1]: sha256
Enter certificate Common Name (e.g. your site's name):  www.mysite.com
Use certificate default values? [y/n]: [y/n]: y
Enter certificate validation period in days (1-3650) [365]:

Self signed server certificate, certificate signing request and key pair added.
```

4. Globally enable SSL.

```
>> Main# /cfg/slb/ssl/on
```

5. Set the HTTPS virtual service to be used in the defined virtual server.

`>> Main# /cfg/slb/virt 1/service https`	(Define the HTTPS service)
`>> Virtual Server 1 443 https Service# group 1`	(Associate the server group to be used in that service)
`>> Virtual Server 1 443 https Service# ssl`	(Switch to the *SSL* menu under the HTTPS service)

`>> SSL Load Balancing# srvrcert`
`Current SSL server certificate: none`
`Enter new SSL server certificate or group`
`[cert|group|none] [none]: cert`
`Enter new SSL server certificate: MyCert`
	(Associate the defined server certificate)

`>> SSL Load Balancing# sslpol myPol`	(Associate the defined SSL Policy)

Note: The back-end server listening port (rport) changes from 443 to 80 because you did not enable back-end encryption. For a different network setting, rport can be configured manually.

6. Optionally, import an Intermediate CA certificate or group and bind it to the SSL policy. For details on Intermediate CA certificates and groups, see the section on the **`/cfg/slb/ssl/certs`** menu in the *Alteon Application Switch Operating System Command Reference*.

   To bind the intermediate CA certificate to the SSL policy use the following command:

`>> Main# /cfg/slb/ssl/sslpol myPol`	(Enter the defined SSL policy)
`>> SSL Policy myPol# intermca <cert|group> <cert/group ID>`	(Select the intermediate CA certificate or group to be used)

7. Enable DAM or configure proxy IP addresses and enable proxy on the client port.

Example 2: Configuring a Basic SSL Offloading Service for a Non-HTTP Protocol

1. Before you can configure an SSL offloading service, ensure that Alteon is configured for basic SLB:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface.
   - Define each real server.
   - Assign servers to real server groups.
   - Enable SLB.
   - Define server port and client port.
   - Define virtual server.

   For more information on how to configure Alteon for SLB, see Server Load Balancing, page 165.
2. Define the SSL Policy which will govern the SSL offloading behavior.

`>> Main# /cfg/slb/ssl/sslpol myPol`	(Define an ID to identify the SSL Policy. The ID may be alphanumeric or numeric.)
`>> SSL Policy myPol# cipher high`	(Select the cipher suite to be used during SSL handshake. By default, the RSA cipher suite is selected. Radware recommends using the PCI-DSS pre-configured cipher suite for best SSL security.)
`>> SSL Policy myPol# ena`	(Enable the policy)

For details on defining additional SSL policy parameters, see the section on the **/cfg/slb/ssl/sslpol** menu in the *Alteon Application Switch Operating System Command Reference*.

3. Define a server certificate for this service:
   - Import a third-party signed server certificate. For details on configuring the certificate repository, see the section on the **/cfg/slb/ssl/certs** menu in the *Alteon Application Switch Operating System Command Reference*.
   - Alternatively, generate a self-signed server certificate, as shown in the following example:

```
>> Main# /cfg/slb/ssl/certs/srvrcert MyCert
>> Server certificate MyCert# generate
This operation will generate a self-signed server certificate.
Enter key size [512|1024|2048|4096] | [1024]:
Enter server certificate hash algorithm [md5|sha1|sha256|sha384|sha512] |
[sha1]: sha256
Enter certificate Common Name (e.g. your site's name):  www.mysite.com
Use certificate default values? [y/n]: [y/n]: y
Enter certificate validation period in days (1-3650) [365]:

Self signed server certificate, certificate signing request and key pair added.
```

4. Globally enable SSL.

```
>> Main# /cfg/slb/ssl/on
```

5. Set the non-HTTP virtual service to be used in the defined virtual server.

<table>
<tr><td>>> Main# /cfg/slb/virt 1/service 12345
Application usage:
http|https|ssl|dns|rtsp|wts|basic-slb
Enter application: ssl</td><td>(Define the service port and select SSL as the service's application type)</td></tr>
<tr><td>>> Virtual Server 1 12345 Service# group 1</td><td>(Associate the server group to be used in that service)</td></tr>
<tr><td>>> Virtual Server 1 12345 Service# ssl</td><td>(Switch to the SSL menu under the service menu)</td></tr>
<tr><td>>> SSL Load Balancing# srvrcert
Current SSL server certificate: none
Enter new SSL server certificate or group
[cert|group|none] [none]: cert
Enter new SSL server certificate: MyCert</td><td>(Associate the defined server certificate)</td></tr>
<tr><td>>> SSL Load Balancing# sslpol myPol</td><td>(Associate the defined SSL Policy)</td></tr>
</table>

Note: The back-end server listening port (rport) is set to 12345. For a different setting, rport can be configured manually.

6. Optionally, import an Intermediate CA certificate or group and bind it to the SSL policy. For details on Intermediate CA certificates and groups, see the section on the **/cfg/slb/ssl/certs** menu in the *Alteon Application Switch Operating System Command Reference*.

   To bind the intermediate CA certificate to the SSL policy use the following command:

<table>
<tr><td>>> Main# /cfg/slb/ssl/sslpol myPol</td><td>(Enter the defined SSL policy)</td></tr>
<tr><td>>> SSL Policy myPol# intermca <cert|group> <cert/group ID></td><td>(Select the intermediate CA certificate or group to be used)</td></tr>
</table>

7. Enable DAM or configure proxy IP addresses and enable proxy on the client port.

Example 3: Configuring an SSL Offloading Service with Back-End Encryption

1. Before you can configure an SSL offloading service, ensure that Alteon is configured for basic SLB:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface.
   - Define each real server.
   - Assign servers to real server groups.
   - Enable SLB.
   - Define server port and client port.
   - Define virtual server.

   For more information on how to configure Alteon for SLB, see Server Load Balancing, page 165.

2. Define the SSL policy which will govern the SSL offloading behavior:

`>> Main# /cfg/slb/ssl/sslpol myPol`	(Define an ID to identify the SSL Policy. The ID may be alphanumeric or numeric.)
`>> SSL Policy myPol# cipher rsa`	(Select the cipher suite to use during SSL handshake. By default, the RSA cipher suite is selected. Radware recommends using the PCI-DSS pre-configured cipher suite for enhanced SSL security.)
`>> SSL Policy myPol# bessl enabled`	(Enable back-end SSL)
`>> SSL Policy myPol# becipher low`	(Set the cipher to be used for back-end connections)
`>> SSL Policy myPol# ena`	(Enable the policy)

For details on defining additional SSL policy parameters, see the section on the **/cfg/slb/ssl/sslpol** menu in the *Alteon Application Switch Operating System Command Reference*.

3. Define a server certificate for this service:
   - Import a third-party signed server certificate. For details on configuring the certificate repository, see the section on the **/cfg/slb/ssl/certs** menu in the *Alteon Application Switch Operating System Command Reference*.
   - Alternatively, generate a self-signed server certificate, as shown in the following example:

```
>> Main# /cfg/slb/ssl/certs/srvrcert MyCert
>> Server certificate MyCert# generate
This operation will generate a self-signed server certificate.
Enter key size [512|1024|2048|4096] | [1024]:
Enter server certificate hash algorithm [md5|sha1|sha256|sha384|sha512] |
[sha1]: sha256
Enter certificate Common Name (e.g. your site's name):  www.mysite.com
Use certificate default values? [y/n]: [y/n]: y
Enter certificate validation period in days (1-3650) [365]:

Self signed server certificate, certificate signing request and key pair added.
```

4. Globally enable SSL.

```
>> Main# /cfg/slb/ssl/on
```

5. Set the HTTPS virtual service to be used in the defined virtual server.

<table>
<tr><td>>> Main# /cfg/slb/virt 1/service https</td><td>(Define the HTTPS service)</td></tr>
<tr><td>>> Virtual Server 1 443 https Service# group 1</td><td>(Associate the servers group to be used in that service)</td></tr>
<tr><td>>> Virtual Server 1 443 https Service# ssl</td><td>(Switch to SSL menu under HTTPS service)</td></tr>
<tr><td>>> SSL Load Balancing# srvrcert
Current SSL server certificate: none
Enter new SSL server certificate or group
[cert|group|none] [none]: cert
Enter new SSL server certificate: MyCert</td><td>(Associate the defined server certificate)</td></tr>
<tr><td>>> SSL Load Balancing# sslpol myPol</td><td>(Associate the defined SSL policy)</td></tr>
</table>

Note: The back-end server listening port (rport) is set to 443 because you enabled back-end encryption. For a different network setting, rport can be configured manually. If the back-end server listening port was previously configured to a specific port, it will not be modified and must be configured manually if required.

6. Optionally, import an Intermediate CA certificate or group and bind it to the SSL policy. For details on Intermediate CA certificates and groups, see the section on the **/cfg/slb/ssl/certs** menu in the *Alteon Application Switch Operating System Command Reference*.

   To bind the intermediate CA certificate to the SSL policy use the following command:

<table>
<tr><td>>> Main# /cfg/slb/ssl/sslpol myPol</td><td>(Enter the defined SSL policy)</td></tr>
<tr><td>>> SSL Policy myPol# intermca <cert|group> <cert/group ID></td><td>(Select the intermediate CA certificate or group to be used)</td></tr>
</table>

7. Enable DAM or configure proxy IP addresses and enable proxy on the client port.
8. When using HTTP SSL offloading with back-end encryption enabled, Radware recommends using multiplexing to minimize the server load of performing new SSL handshakes. For more details on multiplexing, see Content-Intelligent Connection Management, page 277.

Example 4: Configuring an SSL Offloading Service for Multiple Domains on the Same Virtual IP Using Server Name Indication (SNI)

To configure SSL offloading for multiple domains behind a single virtual IP, SSL handshake server name indication (SNI) is used.

1. Before you can configure an SSL offloading service, ensure that Alteon is configured for basic SLB:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface.
   - Define each real server.
   - Assign servers to real server groups.
   - Enable SLB.
   - Define server port and client port.

— Define virtual server.

For more information on how to configure Alteon for SLB, see Server Load Balancing, page 165.

2. Create or import SSL server certificates of all the servers that are SSL offloaded according to Example 1: Configuring a Basic SSL Offloading Service, page 343.
3. Create a certificate group that includes all the server certificates to be used in this VIP.

`/cfg/slb/ssl/certs/`
 	
`>> Certificate Repository# group/`
 	
`Enter group id: 1`
	(Enter the Group menu)

`>> 4416-2 - Group 1# type`
`Current certificate group type: intermca`
`Enter new certificate group type`
`[srvrcert|trustca|intermca]: srvrcert`
	(Select the Group type of the Server Certificate Group)

`>> 4416-2 - Group 1# add`	(Add the server certificate)

`Enter certificate ID:servercert1`
`Certificate servercert1 is added to group 1`
 	
`>> 4416-2 - Group 1# add`
 	
`Enter certificate ID:servercert2`
`Certificate servercert2 is added to group 1`
	(Press the **tab** key to list all existing server certificates or for name completion)

4. Optionally, define a default certificate that to be used for browsers or clients not supporting SNI:

`/cfg/slb/ssl/certs/group`
 	
`>> Group 1# default`
`Current default srvrcert certificate:`
`Enter new default server certificate id to use for`
`non-SNI clients or none: def-cert`
`default srvrcert certificate def-cert is added to`
`group 1`
	(Select **def-cert** as the default certificate)

This certificate can include the various domains for which you do SSL-offloading, using wildcard domain names or a Subject Alternative Name (SAN).

5. Associate the server certificate group to a virtual service according to Example 1: Configuring a Basic SSL Offloading Service, page 343 with the following change:

`>> Main# /cfg/slb/virt 1/service https`	(Define the HTTPS service)
`>> Virtual Server 1 443 https Service# group 1`	(Associate the server group to be used in that service)
`>> Virtual Server 1 443 https Service# ssl`	(Switch to the SSL menu under HTTPS service)

`>> SSL Load Balancing# srvrcert`
`Current SSL server certificate: none`
`Enter new SSL server certificate or group`
`[cert|group|none] [none]: group`
`Enter new SSL server certificate: group1`
	(Associate the defined server certificate group)

`>> SSL Load Balancing# sslpol myPol`	(Associate a SSL policy)

Alteon supports both SSL offloading with and without SNI, and there are various ways to indicate domain names in certificates (common name, wildcards, subject alternative name extension). The following is the order in which certificates are used in various scenarios (SSL offloading certificate matching logic).

— Non-SNI configuration (i.e. a specific server certificate is associated to the virtual service)—in this scenario, no matter whether or not there is an SNI in the SSL hello from the client, the associated server certificate is returned to the client.

Note: Alteon is oblivious to the contents of the certificate. Therefore wildcard certificates or Subject Alternative names (SAN) play no role and are supported.

— SNI configuration—in this scenario, the Alteon matching logic is as follows:

a. Match the client SNI content to the server's certificate common name (CNAME) in the associated certificate group. If there is an exact match, send the matched server certificate to the client.

b. Match the client SNI content to the server's certificate with wildcards, looking for a match in the domain name, and ignoring the hostname. If there is a domain name match (ignoring the hostname), send the matched wildcard server certificate to the client.

c. Match the client SNI content to the server's certificate with Subject Alternative Names (SAN) appearing in each of the servers' certificates in the certificate group. If there is an exact match, send the matched server certificate to the client.

d. If there is no match between client SNI and any of the server domain names, the SSL handshake fails.

e. Whenever no SNI is sent by the client in SSL hello, use the "default" certificate defined in the certificates group and return it to the client.

6. Create Layer7 content switching rules to select the Server group by domain name. See Example Content-Intelligent Server Load Balancing, page 219 for more information about using content switching rules and classes.

```
>>  HTTP Content Class 1# /cfg/slb/layer7/slb/
cntclss 1/hostname 1
```

(Create a content switching rule for each of the domains)

```
>> Hostname 1# hostname
Current hostname to match:
Enter new hostname to match: mydomain.com

>> Hostname 1# match
Current matching type: include
Enter new matching type
[sufx|prefx|equal|include|regex]: eq
```

```
>> Hostname 1# /cfg/slb/virt 1/service 443
```

(Associate the defined content class for every rule)

```
>> Virtual Server 1 443 https Service# cntrules 1

>> HTTPS Content Rule 1# cntclss
Current content class:
Enter new content class or none: domain1
For content class updates use /cfg/slb/layer7/slb
```

```
>> HTTPS Content Rule 1# group 10
Current real server group:     1
New pending real server group: 10
```

(Select the server group to be used for serving each of the domains)

Note: Each of the created objects in this procedure must be enabled.

7. Apply and save your configuration.

Example 5: Configuring an SSL Offloading Service with Client Authentication

1. Before you can configure an SSL offloading service, ensure that Alteon is configured for basic SLB:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface.
   - Define each real server.
   - Assign servers to real server groups.
   - Enable SLB.
   - Define server port and client port.
   - Define virtual server.

   For more information on how to configure Alteon for SLB, see Server Load Balancing, page 165.
2. Define the SSL offloading service which will govern the SSL offloading behavior.
   - For basic SSL offloading, see Example 1: Configuring a Basic SSL Offloading Service, page 343.
   - For SSL offloading with back-end encryption enabled, see Example 3: Configuring an SSL Offloading Service with Back-End Encryption, page 347.

3. Define the Trusted CA used to authenticate the client's certificate by importing its certificate to Alteon.
   a. Import a Trusted CA Certificate into the certificate repository. For details on importing a Trusted CA Certificate, see the section on the **/cfg/slb/ssl/certs/import** menu in the *Alteon Application Switch Operating System Command Reference*.
   b. Optionally, you can define a group of Trusted CA certificates. For details on defining a Trusted CA Certificate group, see the section on the **/cfg/slb/ssl/certs/group** menu in the *Alteon Application Switch Operating System Command Reference*.
4. Define the client authentication policy.

<table>
<tr><td>>> Main#/cfg/slb/ssl/authpol Cauth</td><td>(Define an ID to identify the client authentication policy. The ID may be alphanumeric or numeric.)</td></tr>
<tr><td>>> Client Authentication Policy Cauth# trustca <cert|group> <cert/group ID></td><td>(Select the trust CA certificate or group to be used)</td></tr>
<tr><td>>> Client Authentication Policy Cauth# ena</td><td>(Enable the policy)</td></tr>
<tr><td>>> Client Authentication Policy Cauth# validity</td><td>(Optionally, switch to the Validity menu and set the certificate validation method to OCSP)</td></tr>
<tr><td>>> Client Authentication Policy clientauth Validation# method ocsp</td><td></td></tr>
</table>

For details on defining additional client authentication policy parameters, see the section on the **/cfg/slb/ssl/authpol** menu in the *Alteon Application Switch Operating System Command Reference*.

5. Associate the defined client authenticating policy to the SSL policy used in the HTTPS service.

>> Main# /cfg/slb/ssl/sslpol myPol	(Enter the defined SSL policy)
>> SSL Policy myPol# authpol Cauth	(Associate the defined client Authentication Policy)

6. Enable DAM or configure proxy IP addresses and enable proxy on the client port.

Example 6: Configuring a Clear-text HTTP Service with Back-end Encryption

1. Before you can configure an SSL offloading service, ensure that Alteon is configured for basic SLB, as follows:
   — Assign an IP address to each of the real servers in the server pool.
   — Define an IP interface.
   — Define each real server.
   — Assign servers to real server groups.
   — Enable SLB.
   — Define a server port and client port.
   — Define a virtual server.

   For more information on how to configure Alteon for SLB, see Server Load Balancing, page 165.
2. Define the SSL policy which will govern the SSL offloading behavior:

`>> Main# /cfg/slb/ssl/sslpol myPol`	(Define an ID to identify the SSL Policy. The ID may be alphanumeric or numeric.)
`>> SSL Policy myPol# fessl disable`	(Disable front-end SSL)
`>> SSL Policy myPol# bessl enable`	(Enable back-end SSL)
`>> SSL Policy myPol# becipher high`	(Set the cipher to be used for back-end connections)
`>> SSL Policy myPol# ena`	(Enable the policy)

3. Globally enable SSL.

```
>> Main# /cfg/slb/ssl/on
```

4. Set the HTTP virtual service to be used in the defined virtual server.

`>> Main# /cfg/slb/virt 1/service http`	(Define the HTTP service)
`>> Virtual Server 1 80 http Service# group 1`	(Associate the server group to be used with that service)
`>> Virtual Server 1 80 http Service# ssl`	(Access the *SSL* menu for the HTTP service)
`>> SSL Load Balancing# sslpol myPol`	(Associate the defined SSL policy)

Note: The back-end server listening port (rport) is set to 80 (vport). For a different network setting, rport can be configured manually. If the back-end server listening port was previously configured to a specific port, it will not be modified and must be configured manually if required.

5. Enable DAM or configure proxy IP addresses, and enable proxy on the client port.
6. When using back-end encryption, Radware recommends using multiplexing to minimize the server load of performing new SSL handshakes. For more details on multiplexing, see Content-Intelligent Connection Management, page 277.

Chapter 15 – Filtering and Traffic Manipulation

Alteon enables traffic classification, manipulation and redirection. This chapter includes an overview of filters, load balancing modes, and configuration examples.

Filters are policies that enable classification, manipulation and redirection of traffic for load balancing purposes, network security, Network Address Translation (NAT) and more.

Starting with version 28.1.50, Alteon includes additional filtering features, such as reverse session and redirection to proxy, to support the different load balancing modes. For more information, see Filtering Enhancements, page 363.

Alteon supports the following load balancing modes:

- Routing mode or non-transparent load balancing—Alteon is responsible for full traffic manipulation.
- Semi-transparent load balancing—Alteon redirects traffic to services which perform minor adjustments to the client's packet.
- Transparent load balancing—Alteon performs traffic inspection and classification of all layers, load balancing traffic with one or more service farms while forwarding it to the original destination without any change to the original packet.

The following topics are discussed in this chapter:

Basic Filtering Features

Alteon includes extensive filtering capabilities at the Layer 2 (MAC), Layer 3 (IP), Layer 4 (TCP/UDP), and Layer 7 (content-based) levels.

This section includes an overview of the following topics:

Filtering Benefits

Filtering provides the following benefits:

- **Filtering of Layer 2 non-IP frames**—In Alteon, a filter can specify only source MAC and destination MAC addresses, and capture and apply an allow.
- **Increased security for server networks**—Filtering gives you control over the types of traffic permitted through Alteon. Filters can be configured to allow or deny traffic from Layer 2 through Layer 7, including: MAC address, IP address, protocol, Layer 4 port, Layer 7 string or pattern content.

  Layer 2-only filters, as described in MAC-Based Filters for Layer 2 Traffic, page 373, can be configured to allow or deny non-IP traffic.
- **Map the source or destination IP addresses and ports**—Generic NAT can be used to map the source or destination IP addresses and the ports of private network traffic to or from advertised network IP addresses and ports.

Note: When applied to ports, Alteon filters work exclusively in ingress and not egress.

Filtering Classification Criteria

Up to 2048 filters can be configured. Descriptive names can be used to define filters. Each filter can be set to perform Filtering Actions, page 357 based on any combination of the following filter options:

Table 29: Filter Options

Filter Option	Description
smac	Source MAC address
dmac	Destination MAC address
ipver	IP version
sip	Source IP address or range (see IP Address Ranges, page 360)

Table 29: Filter Options (cont.)

Filter Option	Description
dip	Destination IP address or range (dip and dmask)
proto	Protocol number or name
sport	TCP/UDP application or source port or source port range (such as 31000 through 33000) **Note:** The service number specified on Alteon must match the service specified on the server.
dport	TCP/UDP application or destination port or destination port range (such as 31000 through 33000)
nat	Addresses that are network address translated
vlan	VLAN ID
invert	Reverses the filter logic at layer 4 to activate the filter whenever the specified conditions are *not* met. **Note:** Starting with version 28.1.50, it is possible to reverse the filter logic at layer 7 using an advanced filter option. For more information, see Layer 7 Invert Filter, page 363.

In addition, Alteon supports advanced filtering options, such as TCP flags (Matching TCP Flags, page 391) ICMP message types (Matching ICMP Message Types, page 395), and Layer 7 inversion (Layer 7 Invert Filter, page 363).

Using these filter criteria, you can create a single filter that can potentially perform a very wide variety of actions. Examples of such filters are:

- Block external Telnet traffic to your main server except from a trusted IP address.
- Warn you if FTP access is attempted from a specific IP address.
- Redirect all incoming e-mail traffic to a server where it can be analyzed for spam.

Filtering Actions

A filtering action (**/cfg/slb/filt/action**) instructs the filter what to do when the filtering criteria are matched.

Alteon supports the following filtering actions:

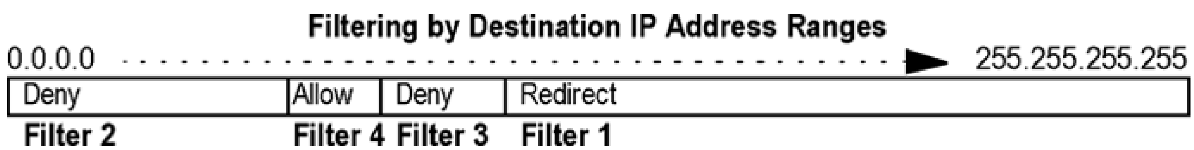
- **allow**—Allows the frame to pass (by default). This filtering action can be used to redirect the returning traffic to the service farm if the reverse session is enabled. For more information, see Reverse Session, page 363.
- **deny**—Discards frames that fit the filter profile. This can be used for building basic security profiles.
- **redir**—Redirects frames that fit the filter profile, such as for Web cache redirection. In addition, Layer 4 processing must be activated using the /cfg/slb/on command.
- **nat**—Performs generic Network Address Translation (NAT). This can be used to map the source or destination IP address and port information of a private network scheme to and from the advertised network IP address and ports. This is used in conjunction with the **nat** option and can also be combined with proxies.
- **goto**—Allows the user to specify a target filter ID that the filter search should jump to when a match occurs. The “goto” action causes filter processing to jump to a designated filter, effectively skipping over a block of filter IDs. Filter searching then continues from the designated filter ID. To specify the new filter to goto, use the **/cfg/slb/filt/adv/goto** command.

Stacking Filters

Stacking filters are assigned and enabled on a per-port basis. Each filter can be used by itself or in combination with any other filter on any given port. The filters are numbered 1 through 2048. When multiple filters are stacked together on a port, the filter number determines its order of precedence; the filter with the lowest number is checked first. When traffic is encountered at the port, if the filter matches, its configured action takes place and the rest of the filters are ignored. If the filter criteria do not match, Alteon tries to match the criteria of the following filter.

As long as the filters do not overlap, you can improve filter performance by making sure that the most heavily used filters are applied first. For example, consider a filter system where the Internet is divided according to destination IP address:

Example Stacking Filters

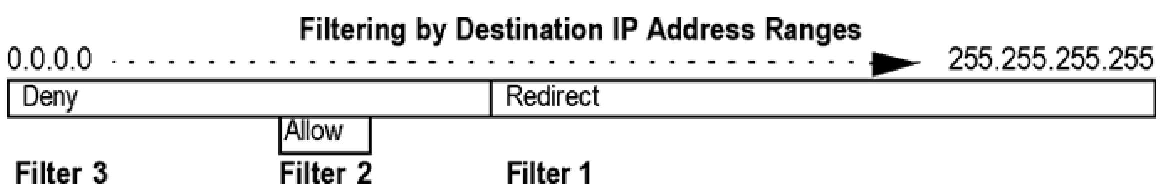


Assuming that traffic is distributed evenly across the Internet, the largest area would be the most used and is assigned to Filter 1. The smallest area is assigned to Filter 4.

Overlapping Filters

Filters are permitted to overlap, although special care must be taken to ensure the proper order of precedence. When there are overlapping filters, the more specific filters (those that target fewer addresses or ports) must be applied before the generalized filters. For example:

Example Overlapping Filters

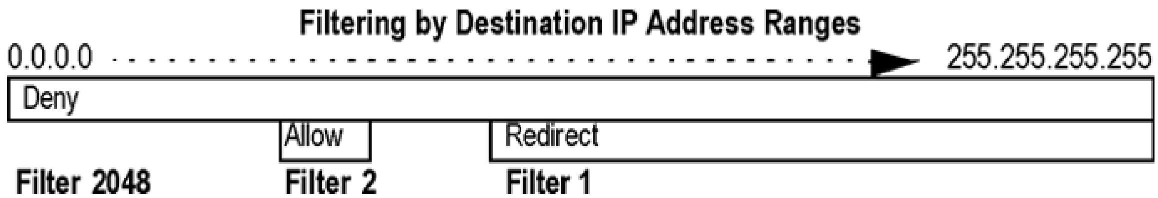


In this example, Filter 2 must be processed prior to Filter 3. If Filter 3 is permitted to take precedence, Filter 2 is never triggered.

Default Filter

Before filtering can be enabled on any given port, a default filter should be configured. This filter handles any traffic not covered by any other filter. All the criteria in the default filter must be set to the fullest range possible (**any**). For example:

Example Default Filter



In this example, the default filter is defined as Filter 2048 to give it the lowest order of precedence. All matching criteria in Filter 2048 are set to **any**. If the traffic does not match the filtering criteria of any other filter and no action is triggered, Filter 2048 processes it, denying and logging unwanted traffic.

`>> # /cfg/slb/filt 2048`	(Select the default filter)
`>> Filter 2048# sip any`	(From any source IP addresses)
`>> Filter 2048# dip any`	(To any destination IP addresses)
`>> Filter 2048# proto any`	(For any protocols)
`>> Filter 2048# action deny`	(Deny matching traffic)
`>> Filter 2048# name deny unwanted traffic`	(Provide a descriptive name for the filter)
`>> Filter 2048# ena`	(Enable the default filter)
`>> Filter 2048# adv`	(Select the advanced menu)
`>> Filter 2048 Advanced# log enable`	(Log matching traffic to syslog)

Default filters are recommended, but not required, when configuring filters for IP traffic control and redirection. Using default filters can increase session performance but takes some of the session binding resources. If you experience an unacceptable number of binding failures, as shown in the Server Load Balancing Maintenance statistics (`/stats/slb/maint`), you may want to remove some of the default filters.

Optimizing Filter Performance

Filter efficiency can be increased by placing filters that are used most often near the beginning of the filtering list.

Note: Radware recommends numbering filters in small increments (5, 10, 15, 20, and so on) to make it easier to insert filters into the list at a later time. However, as the number of filters increases, you can improve performance by minimizing the increment between filters. For example, filters numbered 2, 4, 6, and 8 are more efficient than filters numbered 20, 40, 60, and 80. Peak processing efficiency is achieved when filters are numbered sequentially beginning with 1.

Filtering with Network Classes

You can perform faster searches of ranges, subnets, or single IP addresses by assigning a network class to a filter, identified by a network class name. Using network classes, you can add or remove IP addresses without changing filter or Alteon configurations.

You use a network class to define a filter source IP (sip) or filter destination IP (dip).

For more information on network classes, see Server Load Balancing, page 165.

To assign a network class to a filter

1. Access the *Filter* menu.

```
>> # /cfg/slb/filter 22
```

2. Enter **sip** to specify the source IP address of the filter.

```
>> Filter 22  #sip
Current IP source address or a network class Id : 0.0.0.0
Enter new IP source address or a network class Id :
```

3. Enter the network class ID.

IP Address Ranges

You can specify a range of IP addresses for filtering both the source and/or destination IP address for traffic. When a range of IP addresses is needed, the source IP (sip) address or destination IP (dip) address defines the base IP address in the desired range. The source mask (smask) or destination mask (dmask) is the mask that is applied to produce the range.

For example, to determine if a client request's destination IP address should be redirected to the cache servers attached to a particular Alteon, the destination IP address is masked (bit-wise AND) with the dmask and then compared to the destination IP address.

Example IP Address Ranges

Alteon can be configured with two filters so that each would handle traffic filtering for one half of the Internet. To do this, you could define the following parameters:

Table 30: Filtering IP Address Ranges

Filter	Internet Address Range	dip	dmask
1	0.0.0.0–127.255.255.255	0.0.0.0	128.0.0.0
2	128.0.0.0–255.255.255.255	128.0.0.0	128.0.0.0

Filter Logs

To provide enhanced troubleshooting and session inspection capabilities, packet source and destination IP addresses are included in filter log messages. Filter log messages are generated when a Layer 3 or Layer 4 filter is triggered and has logging enabled. The messages are output to the console port, system host log (syslog), and the Web-based interface message window.

Note: Filter logging should only be used for debugging purposes and not run on production environments, as this may cause excessive CPU utilization if the filter firings are excessive.

Example Filter Logs

A network administrator has noticed a significant number of ICMP frames on one portion of the network and wants to determine the specific sources of the ICMP messages. The administrator uses the CLI to create and apply the following filter:

`>> # /cfg/slb/filt 15`	(Select filter 15)
`>> Filter 15# sip any`	(From any source IP address)
`>> Filter 15# dip any`	(To any destination IP address)
`>> Filter 15# action allow`	(Allows matching traffic to pass)
`>> Filter 15# name allow matching traffic`	(Provide a descriptive name for the filter)
`>> Filter 15# proto icmp`	(For the ICMP protocol)
`>> Filter 15# ena`	(Enable the filter)
`>> Filter 15# adv/log enable`	(Log matching traffic to syslog)
`>> Filter 15 Advanced# /cfg/slb/port 7`	(Select a port to filter)
`>> SLB port 7# add 15`	(Add the filter to the port)
`>> SLB port 7# filt ena`	(Enable filtering on the port)
`>> SLB port 7# apply`	(Apply the configuration changes)
`>> SLB port 7# save`	(Save the configuration changes)

When applied to one or more ports, this simple filter rule produces log messages that show when the filter is triggered, and what the IP source and destination addresses were for the ICMP frames traversing those ports.

Note: After port filtering is enabled or disabled and you apply the change, session entries are deleted immediately.

The following is a filter log message output, displaying the filter number, port, source IP address, and destination IP address:

```
slb: filter 15 fired on port 7, 206.118.93.110 -> 20.10.1.10
```

Cached Versus Non-Cached Filters

To improve efficiency, Alteon by default performs filter processing only on the first frame in each session. Subsequent frames in a session are assumed to match the same criteria and are treated in the same way as the initial frame. These filters create a session entry and are known as *cached*.

Some types of filtering (TCP flag and ICMP message-type filtering) require each frame in the session to be filtered separately. These filters are known as *non-cached*. A Layer 2 filter, which specifies only smac and dmac criteria, is a non-cached filter.

All filters are cached by default. To change the status of a filter, use the following commands:

`>> # /cfg/slb/filt <filter number> /adv`	(Select the *Advanced Filter* menu)
`>> Filter 1 Advanced # cache ena|dis`	(Enable or disable filter caching)

Note: Do not apply cache-enabled filters to the same ports as cache-disabled filters. Otherwise, the cache-disabled filters could potentially be bypassed for frames matching the cache-enabled criteria.

Logging Non-Cached Filter Hits

A non-cached filter hit occurs when a session entry is not cached. Cache-disabled filters are used when a session is either very short-lived or contains minimal data.

In order to log cache-disabled filters without generating an excess amount of syslog messages, the log message displays only a single non-cached filter message within a given window of time, which includes the number of times the cache-disabled filter has fired.

To enable logging of both cached and cache-disabled filters

1. Issue the following command:

```
>> # /cfg/slb/filt <filter number> /adv/log enable
```

2. Apply and save the configuration change.

```
>> Filter <#> Advanced# apply
>> Filter <#> Advanced# save
```

The following is an example of a non-cached filter log message:

```
Jun 28 3:57:57 WARNING slb: NON-cached filter 1 fired on port 1
repeated 4 times.
```

Filtering Enhancements

Starting with version 28.1.50, Alteon simplifies session management through filters. While filters classify user traffic and qualify the proper action, Alteon transparently takes care of session management and proper handling in cases of proxy deployments.

Alteon supports the following filtering enhancements:

- Reverse Session, page 363
- Return to Proxy, page 363
- Layer 7 Invert Filter, page 363

Reverse Session

Filters only handle and search for a match of incoming traffic sent from the client server. In previous versions, filters only created one entry in a session table per session. To handle reverse traffic, either Direct Access Mode (DAM) or a reverse session must be defined.

When using DAM, Alteon changes the source port of the session and identifies the return session by its changed source port. Alteon then reverts the session parameters to the original parameters of the client session.

Previously, when using reverse session, Alteon created a reverse session entry in the session table, handled the packet and reversed its parameters to those of the original client session. However, reverse session could only handle traffic at layer 4.

Starting with version 28.1.50, reverse session returns traffic to the original session without changing the source port and handles traffic at all layers. Return traffic is redirected to the original session table and forwarded to the client with the original parameters.

Reverse session is defined per filter. At Layer 4, if DAM is activated, it takes precedence over reverse session and overrides it. At Layer 7, reverse session takes precedence over DAM. That is, if reverse session is enabled, DAM is automatically overridden.

To view an example using reverse session, see Redirecting Traffic with a Transparent Server, page 364.

Return to Proxy

Alteon supports a wide range of server deployments. In some deployment scenarios, the servers must have the traffic destined to their own assigned IP address, while the service must maintain transparent. Starting with version 28.1.50, you can redirect traffic to such servers by changing the session destination IP to match that of the server. To maintain persistency, that is for the return traffic to return via the proxy, you must enable the reverse session option when using the redirecting to proxy option.

To view an example using return to proxy, see Redirecting Traffic with a NAT Filter, page 366.

Layer 7 Invert Filter

Previously, traffic that matched the layer 7 filtering criteria was redirected to the origin server (internet) and traffic that did not match was redirected real servers.

The layer 7 invert filter now enables the opposite result. A layer 7 invert filter works just like a basic invert filter, except that the invert action is delayed until the string content is examined to see if the session needs to be redirected because of its content.

Traffic that matches the layer 7 invert filtering criteria can be redirected to VAS servers when enabling /cfg/slb/filt/adv/invert .

Load Balancing Modes

Alteon supports a wide range of deployment scenarios, and can perform traffic and flow manipulation, and redirection based on the service requirement. The supported load balancing modes range from being completely transparent to the user to services that are completely visible.

This supported modes include

- Transparent Load Balancing, page 364
- Semi-Transparent Load Balancing, page 367
- Non-Transparent Load Balancing, page 371

Transparent Load Balancing

Transparent load balancing is the deployment of a server load balancer where the network and/or client traffic is not interrupted. That is, Alteon redirects the traffic and returns it to the client without changing any of its parameters. Transparent load balancing can be performed in various ways.

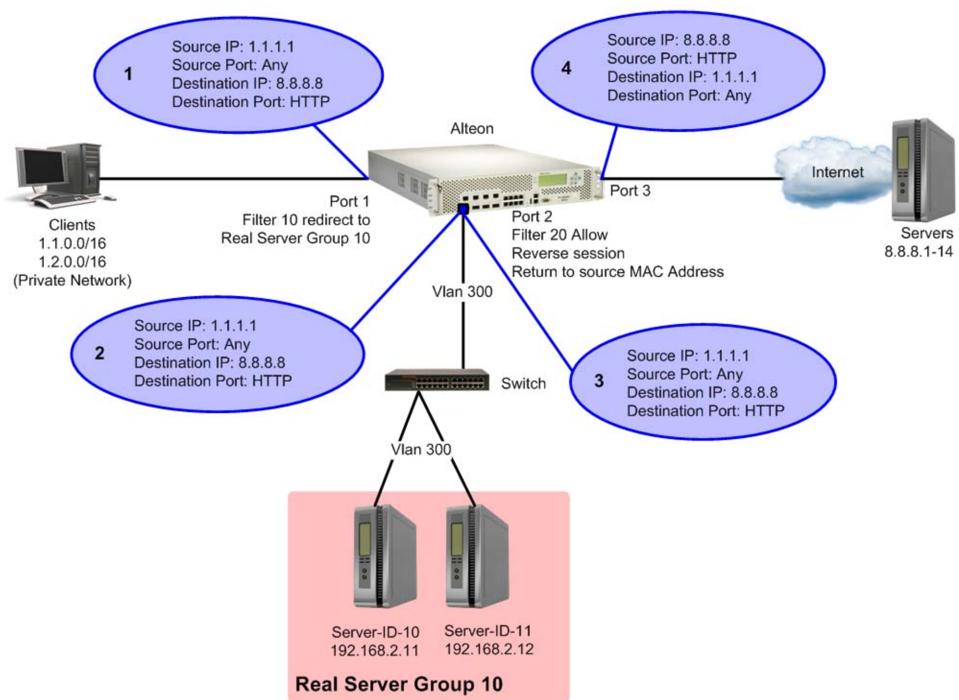
The following are examples of supported transparent load balancing scenarios:

- Redirecting Traffic with a Transparent Server, page 364
- Redirecting Traffic with a NAT Filter, page 366

Redirecting Traffic with a Transparent Server

When redirecting traffic with a transparent server, the client traffic is redirected to a VAS server group. By using reverse session, an opposite entry is added to the session table so that the return traffic matches its source MAC address and is redirected to the VAS server group before returning to the client. None of the client traffic parameters are changed in the process.

Figure 53: Redirecting Traffic with a Transparent Server



To redirect traffic with a transparent server

1. Configure Filter 10 to redirect traffic to Real Server Group 10 (VAS server).

`>> # /cfg/slb/filt 10`	(Select the menu for Filter 10)
`>> Filter 10# sip 1.1.0.0`	(From a specific source IP address)
`>> Filter 10# smask 255.255.0.0`	(From a specific source IP mask)
`>> Filter 10# dip any`	(To any network destination address)
`>> Filter 10# dmask 0.0.0.0`	(For any subnet range)
`>> Filter 10# proto tcp`	(For TCP protocol traffic)
`>> Filter 10# sport any`	(From any source port)
`>> Filter 10# dport http`	(To any HTTP destination port)
`>> Filter 10# action redirect`	(Redirect matching traffic)
`>> Filter 10# group 10`	(Redirect to Real Server Group 10)
`>> Filter 10# vlan any`	(To any VLAN)
`>> Filter 10# ena`	(Enable the filter)

2. Configure Filter 20 to allow client traffic to browse the Web.

`>> # /cfg/slb/filt 20`	(Select the menu for Filter 20)
`>> Filter 20# sip any`	(From any source IP address)
`>> Filter 20# smask 0.0.0.0`	(From any source IP mask)
`>> Filter 20# dip any`	(To any network destination address)
`>> Filter 20# dmask 0.0.0.0`	(For any subnet range)
`>> Filter 20# ipver v4`	(Set filter IP version to IP Version 4)
`>> Filter 20# action allow`	(Allow matching traffic to pass)
`>> Filter 20# vlan any`	(To any VLAN)
`>> Filter 20# ena`	(Enable the filter)

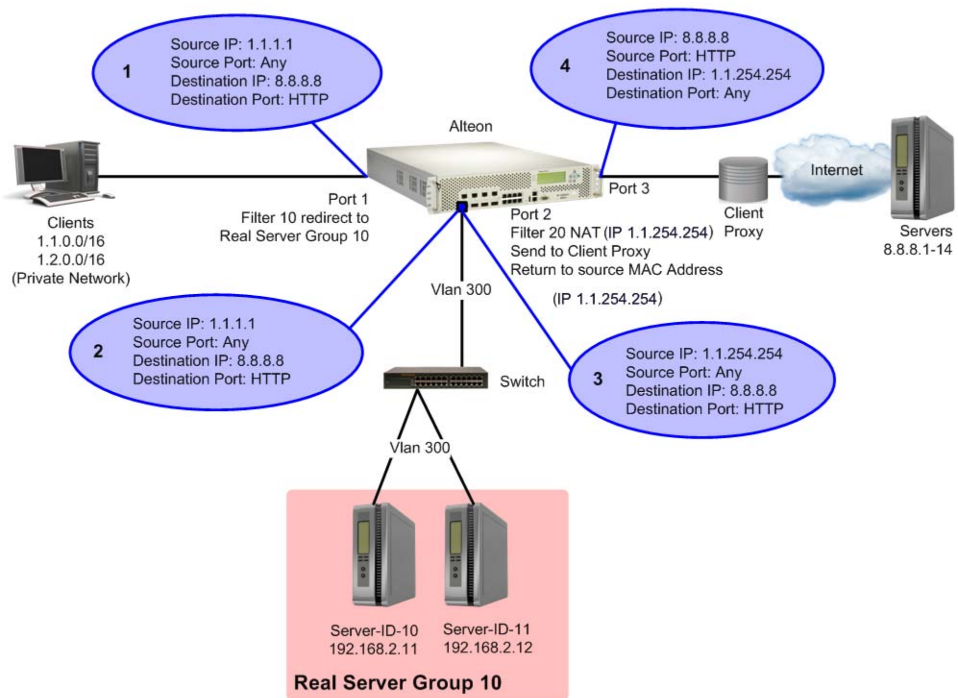
3. Configure Filter 20 to enable the Return to Source MAC address option. This adds an opposite entry in the session table so that the return traffic matches its source MAC address.

`>> # /cfg/slb/filt 20/adv`	(Select the Advanced menu for Filter 20)
`>> Filter 20 Advanced# rtsrcmac ena`	(Enable Return to Source MAC Address)

Redirecting Traffic with a NAT Filter

When redirecting traffic with a NAT filter, the client traffic is first redirected to a VAS server group. Traffic is returned to Alteon transparently through a NAT filter, which changes the client address to CNAT before sending it to the HTTP port. The NAT filter translates the CNAT of the return traffic back to its original state before returning it to the client.

Figure 54: Redirecting Traffic with a NAT Server



To redirect traffic with a NAT filter

1. Configure Filter 10 to redirect traffic to Real Server Group 10 (VAS server).

`>> # /cfg/slb/filt 10`	(Select the menu for Filter 10)
`>> Filter 10# sip 1.1.0.0`	(From a specific source IP address)
`>> Filter 10# smask 255.255.0.0`	(From a specific source IP mask)
`>> Filter 10# dip any`	(To any network destination address)
`>> Filter 10# dmask 0.0.0.0`	(For any subnet range)
`>> Filter 10# proto tcp`	(For TCP protocol traffic)
`>> Filter 10# rport 0`	(To any real server port)
`>> Filter 10# dport http`	(To any HTTP destination port)
`>> Filter 10# ipver v4`	(Set filter IP version to IP Version 4)
`>> Filter 10# action redirect`	(Redirect matching traffic)
`>> Filter 10# group 10`	(Redirect to Real Server Group 10)

`>> Filter 10# vlan any`	(To any VLAN)
`>> Filter 10# ena`	(Enable the filter)

2. Configure Filter 20 as NAT filter to translate source IP addresses before sending client traffic to browse the Web.

`>> # /cfg/slb/filt 20`	(Select the menu for Filter 20)
`>> Filter 20# sip 1.1.0.0`	(From a specific source IP address)
`>> Filter 20# smask 255.255.0.0`	(From a specific source IP mask)
`>> Filter 20# dip any`	(To any network destination address)
`>> Filter 20# dmask 0.0.0.0`	(For any subnet range)
`>> Filter 20# ipver v4`	(Set filter IP version to IP Version 4)
`>> Filter 20# action nat`	(Allow matching traffic to pass)
`>> Filter 20# nat source`	(Set source addresses to be network address translated)
`>> Filter 20# vlan any`	(To any VLAN)
`>> Filter 20# ena`	(Enable the filter)

3. Configure Filter 20 to enable the Return to Source MAC address option. This adds an opposite entry in the session table so that the return traffic matches its source MAC address.

`>> # /cfg/slb/filt 20/adv`	(Select the Advanced menu for Filter 20)
`>> Filter 20 Advanced# rtsrcmac ena`	(Enable Return to Source MAC Address)

4. Configure a client proxy for Filter 20.

`>> # /cfg/slb/filt 20/adv/proxyadv`	(Select the Proxy Advanced menu for Filter 20)
`>> Proxy Advanced# proxyip 1.1.254.254`	(Set client proxy IP address)
`>> Proxy Advanced# proxy enable`	(Enable client proxy on this filter)

Semi-Transparent Load Balancing

When employing semi-transparent load balancing, Alteon redirects the traffic and returns it to the client and changes one or more source parameters in the process.

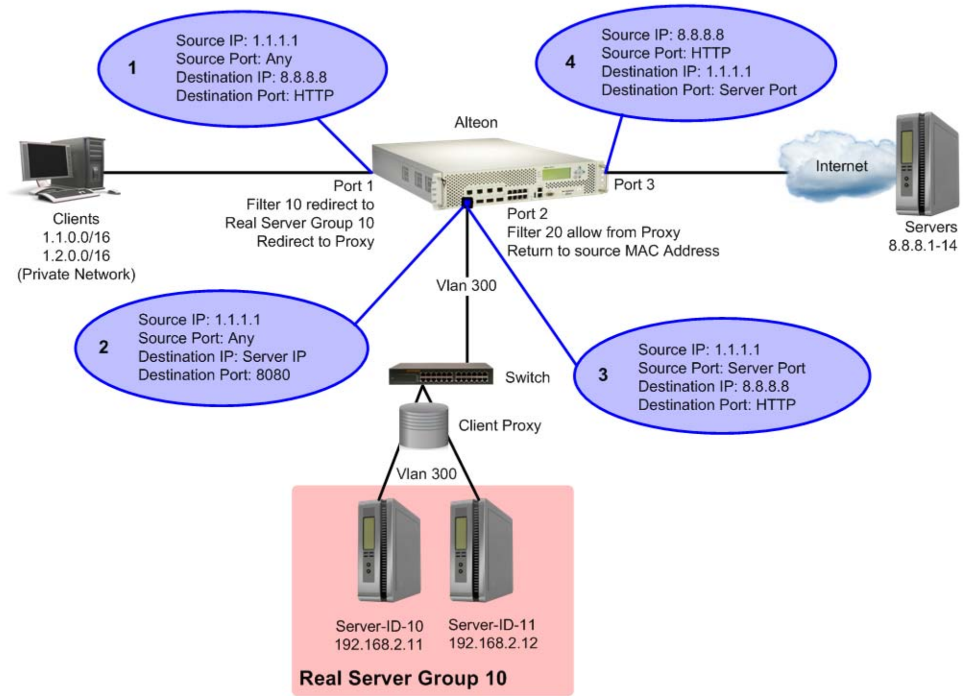
The following are examples of supported semi-transparent load balancing scenarios:

- Redirecting Traffic with a Semi-Transparent Server and Return to Proxy, page 368
- Redirecting Traffic with a Semi-Transparent Server, page 370

Redirecting Traffic with a Semi-Transparent Server and Return to Proxy

When redirecting traffic with a semi-transparent server, the client traffic is redirected to a VAS server group through a proxy server, changing the destination IP and destination port. By using reverse session, an opposite entry is added to the session table so that the return traffic matches its source MAC address and is redirected to the VAS server group. The return traffic is then redirected back to the proxy server and its source IP and source port are translated back to the original before returning to the client.

Figure 55: Redirecting Traffic with a Semi-Transparent Server and Return to Proxy



To redirect traffic with a semi-transparent server and return to proxy

1. Configure Filter 10 to redirect traffic to Real Server Group 10 (VAS server).

`>> # /cfg/slb/filt 10`	(Select the menu for Filter 10)
`>> Filter 10# sip 1.1.0.0`	(From a specific source IP address)
`>> Filter 10# smask 255.255.0.0`	(From a specific source IP mask)
`>> Filter 10# dip any`	(To any network destination address)
`>> Filter 10# dmask 0.0.0.0`	(For any subnet range)
`>> Filter 10# proto tcp`	(For TCP protocol traffic)
`>> Filter 10# rport 8080`	(To real server port 8080)
`>> Filter 10# dport http`	(To any destination port)
`>> Filter 10# ipver v4`	(Set filter IP version to IP Version 4)
`>> Filter 10# action redirect`	(Redirect matching traffic)

`>> Filter 10# group 10`	(Redirect to Real Server Group 10)
`>> Filter 10# vlan any`	(To any VLAN)
`>> Filter 10# ena`	(Enable the filter)

2. Configure Filter 10 to enable the Redirect to Proxy option.

`>> # /cfg/slb/filt 10/adv`	(Select the Advanced menu for Filter 10)
`>> Filter 10 Advanced# redir`	(Select the Redirection Advanced menu for Filter 10)
`>> Filter 10 Advanced# rtproxy ena`	(Enable redirect to proxy server)

3. Configure Filter 20 to allow client traffic from the proxy to browse the Web.

`>> # /cfg/slb/filt 20`	(Select the menu for Filter 20)
`>> Filter 20# sip 1.1.0.0`	(From the proxy IP address)
`>> Filter 20# smask 255.255.0.0`	(From the proxy IP mask)
`>> Filter 20# dip any`	(To any network destination address)
`>> Filter 20# dmask 0.0.0.0`	(For any subnet range)
`>> Filter 20# ipver v4`	(Set filter IP version to IP Version 4)
`>> Filter 20# action allow`	(Allow matching traffic to pass)
`>> Filter 20# vlan any`	(To any VLAN)
`>> Filter 20# ena`	(Enable the filter)

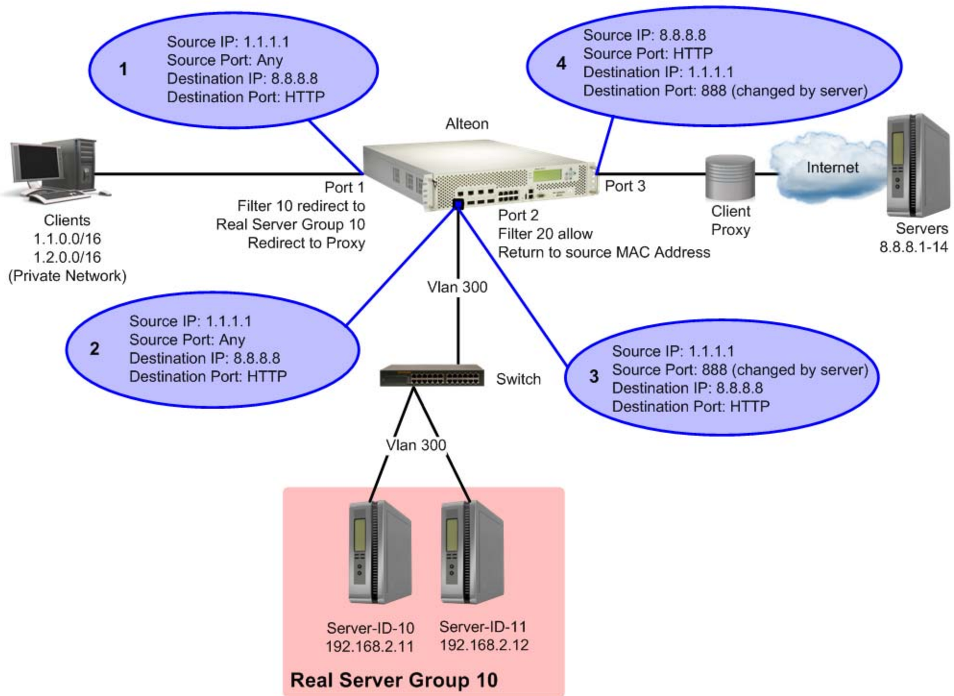
4. Configure Filter 20 to enable the Return to Source MAC address option. This adds an opposite entry in the session table so that the return traffic matches its source MAC address.

`>> # /cfg/slb/filt 20/adv`	(Select the Advanced menu for Filter 20)
`>> Filter 20 Advanced# rtsrcmac ena`	(Enable Return to Source MAC Address)

Redirecting Traffic with a Semi-Transparent Server

When redirecting traffic with a semi-transparent server, the client traffic is redirected to a VAS server group, which changes the server source port. By using reverse session, an opposite entry is added to the session table so that the return traffic matches its source MAC address and is redirected to the VAS server group. The return traffic is then redirected back to the proxy server and its source port are translated back to the original before returning to the client.

Figure 56: Redirecting Traffic with a Semi-Transparent Server



To redirect traffic with a semi-transparent server

1. Configure Filter 10 to redirect traffic to Real Server Group 10 (VAS server).

`>> # /cfg/slb/filt 10`	(Select the menu for Filter 10)
`>> Filter 10# sip 1.1.0.0`	(From a specific source IP address)
`>> Filter 10# smask 255.255.0.0`	(From a specific source IP mask)
`>> Filter 10# dip any`	(To any network destination address)
`>> Filter 10# dmask 0.0.0.0`	(For any subnet range)
`>> Filter 10# proto tcp`	(For TCP protocol traffic)
`>> Filter 10# rport 0`	(To any real server port)
`>> Filter 10# dport http`	(To any destination port)
`>> Filter 10# ipver v4`	(Set filter IP version to IP Version 4)
`>> Filter 10# action redirect`	(Redirect matching traffic)
`>> Filter 10# group 10`	(Redirect to Real Server Group 10)

`>> Filter 10# vlan any`	(To any VLAN)
`>> Filter 10# ena`	(Enable the filter)

2. Configure Filter 20 to allow client traffic to browse the Web.

`>> # /cfg/slb/filt 20`	(Select the menu for Filter 20)
`>> Filter 20# sip 1.1.0.0`	(From the proxy IP address)
`>> Filter 20# smask 255.255.0.0`	(From the proxy IP mask)
`>> Filter 20# dip any`	(To any network destination address)
`>> Filter 20# dmask 0.0.0.0`	(For any subnet range)
`>> Filter 20# ipver v4`	(Set filter IP version to IP Version 4)
`>> Filter 20# action allow`	(Allow matching traffic to pass)
`>> Filter 20# vlan any`	(To any VLAN)
`>> Filter 20# ena`	(Enable the filter)

3. Configure Filter 20 to enable the Return to Source MAC address option. This adds an opposite entry in the session table so that the return traffic matches its source MAC address.

`>> # /cfg/slb/filt 20/adv`	(Select the Advanced menu for Filter 20)
`>> Filter 20 Advanced# rtsrcmac ena`	(Enable Return to Source MAC Address)

Non-Transparent Load Balancing

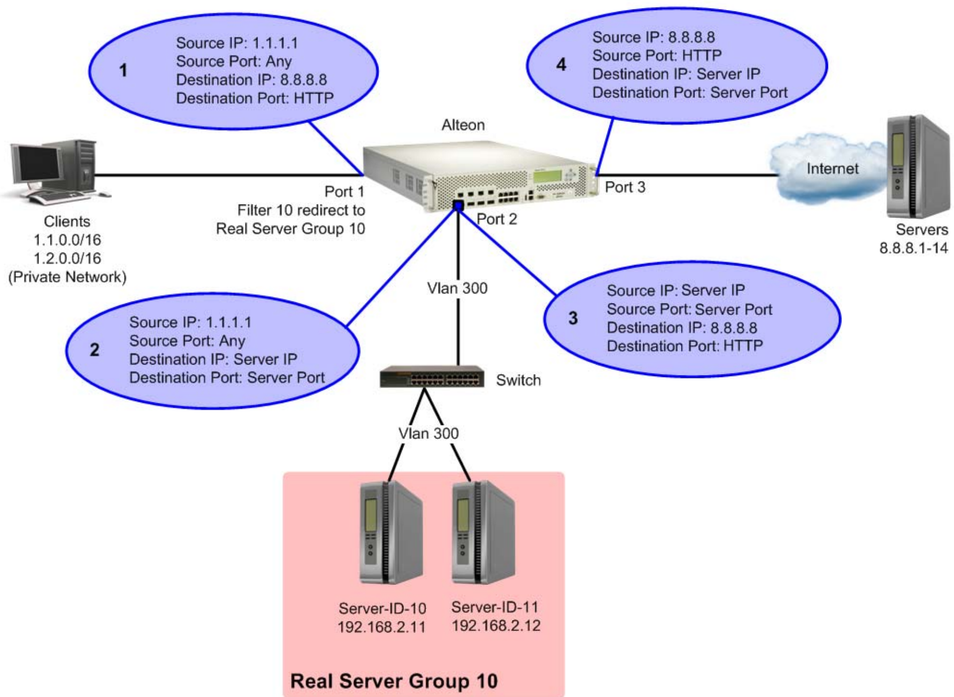
Alteon continues to support non-transparent load balancing. When employing non-transparent load balancing, Alteon redirects the traffic and returns it to the client and changes one or more source or destination parameters in the process.

The following is an example of a supported non-transparent load balancing scenario.

Redirecting Traffic with a Non-Transparent Server

When redirecting traffic with a non-transparent server, the client traffic is redirected to a VAS server group. The VAS server changes the destination IP and destination port to that of the VAS server and sends the traffic to the internet. The return traffic is then redirected back to the VAS server and the server translates its source IP and source port back to the original before returning to the client.

Figure 57: Redirecting Traffic with a Non-Transparent Server



To redirect traffic with a non-transparent server

1. Configure Filter 10 to redirect traffic to Real Server Group 10 (VAS server).

`>> # /cfg/slb/filt 10`	(Select the menu for Filter 10)
`>> Filter 10# sip 1.1.0.0`	(From a specific source IP address)
`>> Filter 10# smask 255.255.0.0`	(From a specific source IP mask)
`>> Filter 10# dip any`	(To any network destination address)
`>> Filter 10# dmask 0.0.0.0`	(For any subnet range)
`>> Filter 10# proto tcp`	(For TCP protocol traffic)
`>> Filter 10# rport 8080`	(To real server port 8080)
`>> Filter 10# dport http`	(To any destination port)
`>> Filter 10# ipver v4`	(Set filter IP version to IP Version 4)
`>> Filter 10# action redirect`	(Redirect matching traffic)
`>> Filter 10# group 10`	(Redirect to Real Server Group 10)
`>> Filter 10# vlan any`	(To any VLAN)
`>> Filter 10# ena`	(Enable the filter)

2. Configure Filter 10 to enable the Redirect to Proxy option.

`>> # /cfg/slb/filt 10/adv`	(Select the Advanced menu for Filter 10)
`>> Filter 10 Advanced# redir`	(Select the Redirection Advanced menu for Filter 10)
`>> Filter 10 Advanced# rtproxy ena`	(Enable redirect to proxy server)

MAC-Based Filters for Layer 2 Traffic

Filters can be configured based on MAC addresses to capture non-IP frames. The benefits of a MAC-based filtering solution is that filters can be applied to allow or deny non-IP traffic such as ARP or AppleTalk. In early Alteon versions, filtering allowed for MAC address criteria, but only IP traffic was supported.

- To configure a filter for non-IP traffic, specify only the source MAC (smac) and destination MAC (dmac) addresses. Do not enter source or destination IP addresses on a MAC-based filter. MAC-based filtering of non-IP frames is supported for non-cached filters only. Even if caching is enabled on this type of filter, it does not create a session entry.
- To configure a MAC-based filter, specify only smac and dmac criteria without any IP-related parameters. The only filtering actions supported for MAC-based filters are **allow** and **deny**.

MAC-based filters are supported for VLAN-based filters (see VLAN-Based Filtering, page 373), and 802.1p bit filtering (see Filtering on 802.1p Priority Bit in a VLAN Header, page 376).

Example MAC-Based Filters for Layer 2 Traffic

`>> # /cfg/slb/filt 23`	(Select the menu for Filter 23)
`Filter 23# smac any`	(From any source MAC address)
`>> Filter 23# dmac 00:60:cf:40:56:00`	(To this MAC destination address)
`>> Filter 23# action deny`	(Deny matching traffic)
`>> Filter 23# ena`	(Enable this filter)

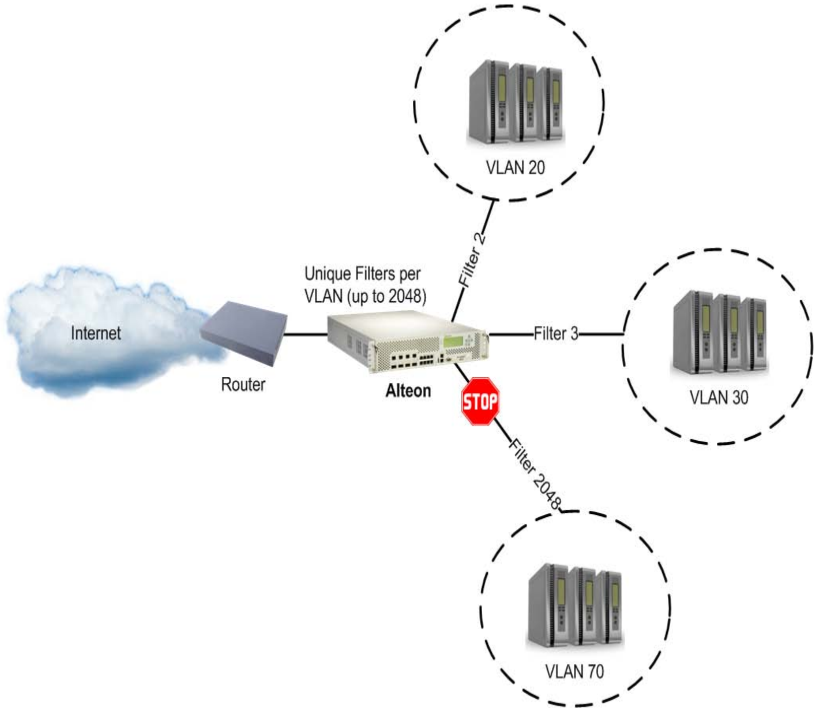
VLAN-Based Filtering

Filters are applied per Alteon, per port, or per VLAN. VLAN-based filtering allows a single Alteon to provide differentiated services for multiple customers, groups, or departments. For example, you can define separate filters for Customers A and B on the same Alteon on two different VLANs. If VLANs are assigned based on data traffic, for example, ingress traffic on VLAN 1, egress traffic on VLAN 2, and management traffic on VLAN 3, filters can be applied accordingly to the different VLANs.

Example VLAN-Based Filtering

In the example in Figure 58 - Example VLAN-Based Filtering Configuration, page 374, Filter 2 is configured to allow local clients on VLAN 20 to browse the Web, and Filter 3 is configured to allow local clients on VLAN 30 to Telnet anywhere outside the local intranet. Filter 2048 is configured to deny ingress traffic from VLAN 70.

Figure 58: Example VLAN-Based Filtering Configuration



To configuring VLAN-based filtering

This procedure is based on Figure 58 - Example VLAN-Based Filtering Configuration, page 374.

Note: While this example is based on IP traffic, VLAN-based filtering can also be used for non-IP traffic by specifying smac and dmac criteria instead of sip and dip.

1. Configure Filter 2 to allow local clients to browse the Web and then assign VLAN 20 to the filter.

   The filter must recognize and allow TCP traffic from VLAN 20 to reach the local client destination IP addresses if originating from any HTTP source port.

`>> # /cfg/slb/filt 2`	(Select the menu for Filter 2)
`>> Filter 2# sip any`	(From any source IP address)
`>> Filter 2# dip 205.177.15.0`	(To base local network destination address)
`>> Filter 2# dmask 255.255.255.0`	(For entire subnet range)
`>> Filter 2# proto tcp`	(For TCP protocol traffic)
`>> Filter 2# sport http`	(From any source HTTP port)
`>> Filter 2# dport any`	(To any destination port)
`>> Filter 2# action allow`	(Allow matching traffic to pass)

`>> Filter 2# vlan 20`	(Assign VLAN 20 to Filter 2)
`>> Filter 2# ena`	(Enable the filter)

All clients from other VLANs are ignored.

2. Configure Filter 3 to allow local clients to telnet anywhere outside the local intranet and then assign VLAN 30 to the filter.

   The filter must recognize and allow TCP traffic to reach the local client destination IP addresses if originating from a Telnet source port.

`>> # /cfg/slb/filt 3`	(Select the menu for Filter 3)
`>> Filter 3# sip any`	(From any source IP address)
`>> Filter 3# dip 205.177.15.0`	(To base local network destination address)
`>> Filter 3# dmask 255.255.255.0`	(For entire subnet range)
`>> Filter 3# proto tcp`	(For TCP protocol traffic)
`>> Filter 3# sport telnet`	(From a Telnet port)
`>> Filter 3# dport any`	(To any destination port)
`>> Filter 3# action allow`	(Allow matching traffic to pass)
`>> Filter 3# name allow clients to telnet`	(Provide a descriptive name for the filter)
`>> Filter 3# vlan 30`	(Assign VLAN 30 to Filter 3)
`>> Filter 3# ena`	(Enable the filter)

3. Configure Filter 2048 to deny traffic and then assign VLAN 70 to the filter. As a result, ingress traffic from VLAN 70 is denied entry to Alteon.

`>> # /cfg/slb/filt 2048`	(Select the menu for Filter 2048)
`>> Filter 2048# sip any`	(From any source IP address)
`>> Filter 2048# dip 205.177.15.0`	(To base local network destination address)
`>> Filter 2048# dmask 255.255.255.0`	(For entire subnet range)
`>> Filter 2048# proto tcp`	(For TCP protocol traffic)
`>> Filter 2048# sport http`	(From a Telnet port)
`>> Filter 2048# dport any`	(To any destination port)
`>> Filter 2048# action deny`	(Allow matching traffic to pass)
`>> Filter 2048# vlan 70`	(Assign VLAN 70 to Filter 2048)
`>> Filter 2048# ena`	(Enable the filter)

4. Assign VLAN-based filters to an SLB port.

   Before the filters can be used, they must be assigned to an SLB port.

`>> # /cfg/slb/port 10`	(Select the menu for the port in use)
`>> SLB Port 10# add 2`	(Add Filter 2 to SLB Port 10)

```
>> SLB Port 10# add 3          (Add Filter 3 to SLB Port 10)
>> SLB Port 10# add 2048       (Add Filter 2048 to SLB Port 10)
```

Filtering on 802.1p Priority Bit in a VLAN Header

Alteon lets you filter based on the priority bits in a packet's VLAN header. The priority bits are defined by the 802.1p standard within the IEEE 802.1Q VLAN header. The 802.1p bits, if present in the packet, specify the priority that should be given to packets during forwarding. Packets with higher (non-zero) priority bits should be given forwarding preference over packets with numerically lower priority bit value.

802.1p Priorities

The IEEE 802.1p standard uses eight levels of priority, 0 through 7, with priority 7 being assigned to highest priority network traffic such as OSPF or RIP routing table updates, priorities 5 though 6 being for delay-sensitive applications such as voice and video, and lower priorities for standard applications. A value of zero indicates a "best effort" traffic prioritization, and this is the default when traffic priority has not been configured on your network. Alteon can only filter packets based on the 802.1p values already present in the packets. It does not assign or overwrite the 802.1p values in the packet.

Classifying Packets Based on 802.1p Priority Bits

Traffic is easily classified based on its 802.1p priority by applying a filter based on the priority bit value. The *Filtering Advanced* menu provides the option to filter based on the priority bit value. The filter matches if it finds the corresponding 802.1p bit value in the packet. If the packet does not have the 802.1p bits, the filter does not match.

To configuring a filter to classify traffic

1. Configure a filter and an action.

```
 >> # /cfg/slb/filt <x> /ena          (Enable the filter)
>> Filter 1 # action allow            (Set filter action)
```

2. Go to the *Filtering Advanced* menu and select the 802.1p priority value.

```
 >> # /cfg/slb/filt <x>
>> Filter <x># adv/8021p/             (Select the 802.1p advanced menu)
>> 802.1p Advanced# match ena         (Enable matching of 802.1p value)
>> # 802.1p Advanced# value 1         (Set the 802.1p priority value to match)
```

Apply a Bandwidth Management (BWM) contract to the prioritized filter. You can apply an 802.1p-prioritized filter to a BWM contract to establish the rule for how the traffic that matches the defined 802.1p priority value. For more information on configuring a BWM contract, see Contracts, page 761.

```
 >> # /cfg/slb/filt <x> /adv/cont 1
```

Persistence for Filter Redirection

The persistence feature ensures that all connections from a specific client session reach the same real server. Alteon provides the following options for persistence when using filter redirection:

- **Layer 3/4 persistence**—The hash is based on Layer 3/4 session parameters. You can choose from a number of options for the *hash input* (also called *tunable hash*): source IP address, destination IP address, both source and destination IP addresses, or source IP address and source port.
- **HTTP Layer 7 persistence**—The hash is based on any HTTP header value.
- **Persistence binding per filter**—The **/cfg/slb/filt <filter number>/adv/redir/pbind** command enables persistent binding for redirection. It is applicable when using redirect filters for SLB instead of virtual services. When enabled, persistence is maintained across multiple sessions from the same client (same source IP).

  Persistence-based SLB enables the network administrator to configure the network to redirect requests from a client to the same real server that initially handled the request. For example, when a server has data associated with a specific user that is not dynamically shared with other servers at the site.

  Persistence binding per filter is similar to client IP-based persistence for virtual services, where the **cip**, **dip**, **rport**, and **dport** values force sessions with values that match the filter to be redirected to the same server in the group.

Notes

- When either Layer 3/4 or Layer 7 persistence is required, the group metric must be set to **hash** or **minmiss**.
- HTTP Layer 7 persistence, when configured, overwrites the Layer 3/4 persistence setting.
- Persistence binding per filter cannot be enabled with Layer 7 content lookup (**/cfg/slb/filt <filter number>/adv/layer7/l7lkup**) because **pbind** server selection uses Layer 3 and 4 criteria, while the **l7lkup** command can use Layer 7 SLB strings attached to the server.
- If Firewall Load Balancing (FWLB) is enabled, the FWLB filter which hashes on the source and destination IP addresses overrides the tunable hash filter. For more information, see Firewall Load Balancing, page 657.

To configure Layer 3/4 persistence (tunable hash) for filter redirection

1. Configure hashing based on source IP address:

`>> # /cfg/slb/filt 10/ena`	(Enable the filter)
`>> Filter 10 # action redir`	(Specify the redirection action)
`>> Filter 10 # proto tcp`	(Specify the protocol)
`>> Filter 10 # group 1`	(Specify the group of real servers)
`>> Filter 10 # rport 3128`	(Specify the redirection port)
`>> Filter 10 # vlan any`	(Specify the VLAN)
`>> Filter 10 # adv`	(Select the *Advanced Filter* menu)
`>> TCP advanced menu # thash sip`	(Select source IP address for hashing)

   Hashing on the 24-bit source IP address ensures that client requests access the same cache.

2. Set the metric for the real server group to **minmiss** or **hash**.

   The source IP address is passed to the real server group for either of the two metrics.

`>> # /cfg/slb/group 1`	(Select the group of real servers)
`>> Real server group 1 # metric minmiss`	(Set the metric to minmiss or hash)

To configure HTTP Layer 7 persistence for filter redirection

1. Configure hashing based on User-Agent HTTP header:

`>> # /cfg/slb/filt 10/ena`	(Enable the filter)
`>> Filter 10 # action redir`	(Specify the redirection action)
`>> Filter 10 # proto 80`	(Specify the protocol)
`>> Filter 10 # group 1`	(Specify the group of real servers)
`>> Filter 10 # vlan any`	(Specify the VLAN)
`>> Filter 10 # adv`	(Select the *Advanced Filter* menu)
`>> Filter 10 Advanced # layer7`	(Select *the Layer 7 Advanced Filter* menu)
`>> Layer 7 Advanced # httphash headerhash User-Agent 20`	(Specify the header name and the length of the value to use)

2. Set the metric for the real server group to **minmiss** or **hash**.

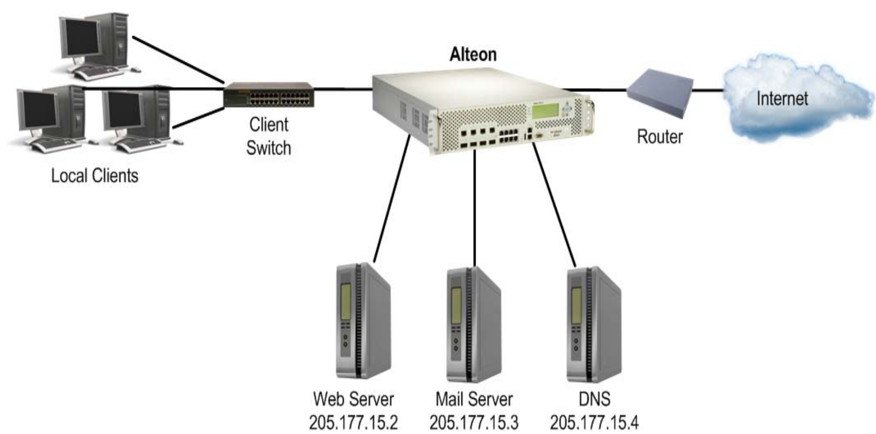
   The source IP address is passed to the real server group for either of the two metrics.

`>> # /cfg/slb/group 1`	(Select the group of real servers)
`>> Real server group 1 # metric minmiss`	(Set the metric to minmiss or hash)

Filter-Based Security

This section includes an example for configuring filters for providing the best security. Radware recommends that you configure filters to deny all traffic except for those services that you specifically want to allow. Consider the example network in Figure 59 - Filter-Based Security Configuration Example, page 379:

Figure 59: Filter-Based Security Configuration Example



In this example, the network is made of local clients on a collector Alteon, a Web server, a mail server, a domain name server, and a connection to the Internet. All the local devices are on the same subnet. The administrator wants to install basic security filters to allow only the following traffic:

- External HTTP access to the local Web server
- External SMTP (mail) access to the local mail server
- Local clients browsing the World Wide Web
- Local clients using Telnet to access sites outside the intranet
- DNS traffic

All other traffic is denied and logged by the default filter.

Note: Since IP address and port information can be manipulated by external sources, filtering does not replace the necessity for a well-constructed network firewall.

To configure a filter-based security solution

Notes

- In this example, all filters are applied only to the port that connects to the Internet. If intranet restrictions are required, filters can be placed on ports connecting to local devices.

- Filtering is not limited to the few protocols and TCP or UDP applications shown in this example. See Well-Known Application Ports, page 175 for a list of well-known applications ports.

1. Before you begin, you must be logged into the CLI as the administrator.
2. Assign an IP address to each of the network devices.

   For this example, the network devices have the following IP addresses on the same IP subnet:

Table 31: Web Cache Example Real Server IP Addresses

Network Device	IP address
Local Subnet	205.177.15.0 - 205.177.15.255
Web Server	205.177.15.2
Mail Server	205.177.15.3
Domain Name Server	205.177.15.4

3. Create a default filter to deny and log unwanted traffic.

   The default filter is defined as Filter 2048 in order to give it the lowest order of precedence:

`>> # /cfg/slb/filt 2048`	(Select the default filter)
`>> Filter 2048# sip any`	(From any source IP addresses)
`>> Filter 2048# dip any`	(To any destination IP addresses)
`>> Filter 2048# proto any`	(For any protocols)
`>> Filter 2048# action deny`	(Deny matching traffic)
`>> Filter 2048# name deny unwanted traffic`	(Provide a descriptive name for the filter)
`>> Filter 2048# ena`	(Enable the default filter)
`>> Filter 2048# adv/log enable`	(Log matching traffic to syslog)

Note: Because the proto parameter is *not tcp or udp*, the source port (sport) and destination port (dport) values are ignored and may be excluded from the filter configuration.

4. Create a filter that allows external HTTP requests to reach the Web server.

   The filter must recognize and allow TCP traffic with the Web server's destination IP address and HTTP destination port:

`>> Filter 2048# /cfg/slb/filt 1`	(Select the menu for Filter 1)
`>> Filter 1# sip any`	(From any source IP address)
`>> Filter 1# dip 205.177.15.2`	(To Web server destination IP address)
`>> Filter 1# dmask 255.255.255.255`	(Set mask for exact destination address)
`>> Filter 1# proto tcp`	(For TCP protocol traffic)
`>> Filter 1# sport any`	(From any source port)
`>> Filter 1# dport http`	(To an HTTP destination port)
`>> Filter 1# action allow`	(Allow matching traffic to pass)

`>> Filter 1# name allow matching traffic`	(Provide a descriptive name for the filter)
`>> Filter 1# ena`	(Enable the filter)

5. Create a pair of filters to allow incoming and outgoing mail to and from the mail server.

   Filter 2 allows incoming mail to reach the mail server, and Filter 3 allows outgoing mail to reach the Internet:

`>> Filter 1# /cfg/slb/filt 2`	(Select the menu for Filter 2)
`>> Filter 2# sip any`	(From any source IP address)
`>> Filter 2# dip 205.177.15.3`	(To mail server destination IP address)
`>> Filter 2# dmask 255.255.255.255`	(Set mask for exact destination address)
`>> Filter 2# proto tcp`	(For TCP protocol traffic)
`>> Filter 2# sport any`	(From any source port)
`>> Filter 2# dport smtp`	(To a SMTP destination port)
`>> Filter 2# action allow`	(Allow matching traffic to pass)
`>> Filter 2# ena`	(Enable the filter)
`>> Filter 2# /cfg/slb/filt 3`	(Select the menu for Filter 3)
`>> Filter 3# sip 205.177.15.3`	(From mail server source IP address)
`>> Filter 3# smask 255.255.255.255`	(Set mask for exact source address)
`>> Filter 3# dip any`	(To any destination IP address)
`>> Filter 3# proto tcp`	(For TCP protocol traffic)
`>> Filter 3# sport smtp`	(From a SMTP port)
`>> Filter 3# dport any`	(To any destination port)
`>> Filter 3# action allow`	(Allow matching traffic to pass)
`>> Filter 3# ena`	(Enable the filter)

6. Create a filter that allows local clients to browse the Web.

   The filter must recognize and allow TCP traffic to reach the local client destination IP addresses if traffic originates from any HTTP source port:

`>> Filter 3# /cfg/slb/filt 4`	(Select the menu for Filter 4)
`>> Filter 4# sip any`	(From any source IP address)
`>> Filter 4# dip 205.177.15.0`	(To base local network destination address)
`>> Filter 4# dmask 255.255.255.0`	(For entire subnet range)
`>> Filter 4# proto tcp`	(For TCP protocol traffic)
`>> Filter 4# sport http`	(From any source HTTP port)
`>> Filter 4# dport any`	(To any destination port)
`>> Filter 4# action allow`	(Allow matching traffic to pass)
`>> Filter 4# name allow clients Web browse`	(Provide a descriptive name for the filter)
`>> Filter 4# ena`	(Enable the filter)

7. Create a filter that allows local clients to telnet anywhere outside the local intranet.

   The filter must recognize and allow TCP traffic to reach the local client destination IP addresses if originating from a Telnet source port:

`>> Filter 4# /cfg/slb/filt 5`	(Select the menu for Filter 5)
`>> Filter 5# sip any`	(From any source IP address)
`>> Filter 5# dip 205.177.15.0`	(To base local network destination address)
`>> Filter 5# dmask 255.255.255.0`	(For entire subnet range)
`>> Filter 5# proto tcp`	(For TCP protocol traffic)
`>> Filter 5# sport telnet`	(From a Telnet port)
`>> Filter 5# dport any`	(To any destination port)
`>> Filter 5# action allow`	(Allow matching traffic to pass)
`>> Filter 5# ena`	(Enable the filter)

8. Create a series of filters to allow Domain Name System (DNS) traffic. DNS traffic requires four filters; one pair is needed for UDP traffic (incoming and outgoing) and another pair for TCP traffic (incoming and outgoing).

   a. For UDP:

`>> Filter 5# /cfg/slb/filt 6`	(Select the menu for Filter 6)
`>> Filter 6# sip any`	(From any source IP address)
`>> Filter 6# dip 205.177.15.4`	(To local DNS Server)
`>> Filter 6# dmask 255.255.255.255`	(Set mask for exact destination address)
`>> Filter 6# proto udp`	(For UDP protocol traffic)
`>> Filter 6# sport any`	(From any source port)
`>> Filter 6# dport domain`	(To any DNS destination port)
`>> Filter 6# action allow`	(Allow matching traffic to pass)
`>> Filter 6# ena`	(Enable the filter)
`>> Filter 6# /cfg/slb/filt 7`	(Select the menu for Filter 7)
`>> Filter 7# sip 205.177.15.4`	(From local DNS Server)
`>> Filter 7# smask 255.255.255.255`	(Set mask for exact source address)
`>> Filter 7# dip any`	(To any destination IP address)
`>> Filter 7# proto udp`	(For UDP protocol traffic)
`>> Filter 7# sport domain`	(From a DNS source port)
`>> Filter 7# dport any`	(To any destination port)
`>> Filter 7# action allow`	(Allow matching traffic to pass)
`>> Filter 7# ena`	(Enable the filter)

   b. Similarly, for TCP:

`>> Filter 7# /cfg/slb/filt 8`	(Select the menu for Filter 8)
`>> Filter 8# sip any`	(From any source IP address)
`>> Filter 8# dip 205.177.15.4`	(To local DNS Server)

`>> Filter 8# dmask 255.255.255.255`	(Set mask for exact destination address)
`>> Filter 8# proto tcp`	(For TCP protocol traffic)
`>> Filter 8# sport any`	(From any source port)
`>> Filter 8# dport domain`	(To any DNS destination port)
`>> Filter 8# action allow`	(Allow matching traffic to pass)
`>> Filter 8# ena`	(Enable the filter)
`>> Filter 8# /cfg/slb/filt 9`	(Select the menu for Filter 9)
`>> Filter 9# sip 205.177.15.4`	(From local DNS Server)
`>> Filter 9# smask 255.255.255.255`	(Set mask for exact source address)
`>> Filter 9# dip any`	(To any destination IP address)
`>> Filter 9# proto tcp`	(For TCP protocol traffic)
`>> Filter 9# sport domain`	(From a DNS source port)
`>> Filter 9# dport any`	(To any destination port)
`>> Filter 9# action allow`	(Allow matching traffic to pass)
`>> Filter 9# ena`	(Enable the filter)

9. Assign the filters to the port that connects to the Internet.

`>> Filter 9# /cfg/slb/port 5`	(Select the SLB port 5 to the Internet)
`>> SLB Port 5# add 1-9`	(Add filters 1 through 9 to port 5)
`>> SLB Port 5# add 2048`	(Add the default filter to port 5)
`>> SLB Port 5# filt enable`	(Enable filtering for port 5)

Alteon lets you add and remove a contiguous block of filters with a single command.

10. Apply and verify the configuration.

```
>> SLB Port 5# apply
>> SLB Port 5# cur
```

Note: After port filtering is enabled or disabled and you apply the change, session entries are deleted immediately.

Examine the resulting information. If any settings are incorrect, make appropriate changes.

11. Save your new configuration changes.

```
>> SLB Port 5# save
```

12. Check the SLB information.

```
>> SLB Port 5# /info/slb/dump
```

13. Check that all SLB parameters are working as expected. If necessary, make any appropriate configuration changes and then check the information again.

Note: Changes to filters on a given port do not take effect until the port's session information is updated (every two minutes or so). To make filter changes take effect immediately, clear the session binding table for the port (see the **`/oper/slb/clear`** command in the *Alteon Application Switch Operating System Command Reference)*.

Network Address Translation

Network Address Translation (NAT) is an Internet standard that enables Alteon to use one set of IP addresses for internal traffic and a second set of addresses for external traffic. Alteon uses filters to implement NAT.

NAT serves two main purposes:

- Provides a type of firewall by hiding internal IP addresses, increasing network security.
- Enables a company to use more internal IP addresses. Since they are used internally only, there is no possibility of conflict with public IP addresses used by other companies and organizations.

In the NAT examples in this section, a company has configured its internal network with private IP addresses. A private network is one that is isolated from the global Internet and is, therefore, free from the usual restrictions requiring the use of registered, globally unique IP addresses.

With NAT, private networks are not required to remain isolated. Alteon NAT capabilities allow internal, private network IP addresses to be translated to valid, publicly advertised IP addresses and back again. NAT can be configured in one of the following two ways:

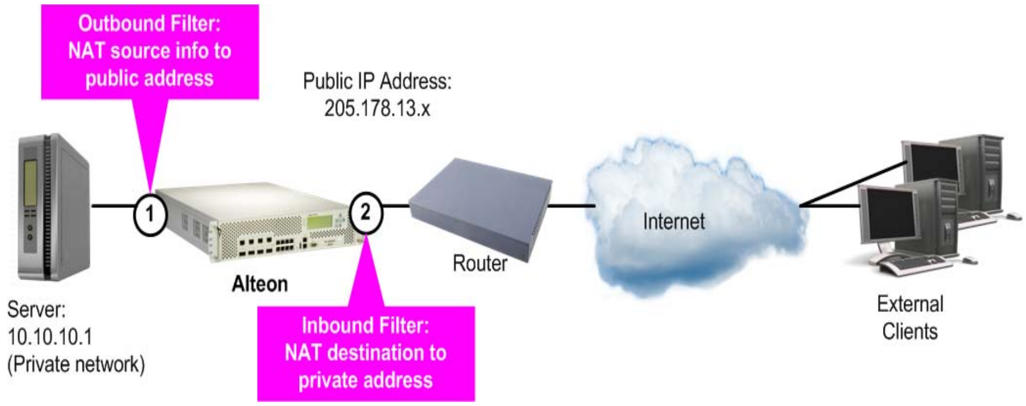
- Static NAT provides a method for direct mapping of one predefined IP address (such as a publicly available IP address) to another (such as a private IP address).
- Dynamic NAT provides a method for mapping multiple IP addresses (such as a group of internal clients) to a single IP address (to conserve publicly advertised IP addresses).

Static NAT

In the following example for static NAT (non-proxy), there are two filters: one for the external client-side port, and one for the internal, server-side port. The client-side filter translates incoming requests for the publicly advertised server IP address to the server's internal private network address. The filter for the server-side port reverses the process, translating the server's private address information to a valid public address.

In Figure 60 - Static NAT Example, page 385, clients on the Internet require access to servers on the private network:

Figure 60: Static NAT Example



To configure static NAT

`>> # /cfg/slb/filt 10`	(Select the menu for outbound filter)
`>> Filter 10# action nat`	(Perform NAT on matching traffic)
`>> Filter 10# nat source`	(Translate source information)
`>> Filter 10# sip 10.10.10.0`	(From the clients private IP address)
`>> Filter 10# smask 255.255.255.0`	(For the entire private subnet range)
`>> Filter 10# dip 205.178.13.0`	(To the public network address)
`>> Filter 10# dmask 255.255.255.0`	(For the same subnet range)
`>> Filter 10# ena`	(Enable the filter)
`>> Filter 10# adv/proxyadv/proxy disable`	(Override any proxy IP settings. Static NAT is used for this filter.)
`>> Enter new NAT IP address:`	(Set the NAT Address)
`>> Filter 10 Advanced# /cfg/slb/filt 11`	(Select the menu for inbound filter)
`>> Filter 11# action nat`	(Use the same settings as outbound)
`>> Filter 11# nat dest`	(Reverse the translation direction)
`>> Filter 11# sip 10.10.10.0`	(Use the same settings as outbound)
`>> Filter 11# smask 255.255.255.0`	(Use the same settings as outbound)
`>> Filter 11# dip 205.178.13.0`	(Use the same settings as outbound)
`>> Filter 11# dmask 255.255.255.0`	(Use the same settings as outbound)
`>> Filter 11# ena`	(Enable the filter)
`>> Filter 11# adv/proxy disable`	(Override any proxy IP settings)
`>> Filter 11 Advanced# /cfg/slb/port 1`	(Select server-side port)
`>> SLB port 1# add 10`	(Add the outbound filter)
`>> SLB port 1# filt enable`	(Enable filtering on port 1)

`>> SLB port 1# /cfg/slb/port 2`	(Select the client-side port)
`>> SLB port 2# add 11`	(Add the inbound filter)
`>> SLB port 2# filt enable`	(Enable filtering on port 2)
`>> SLB port 2# apply`	(Apply configuration changes)
`>> SLB port 2# save`	(Save configuration changes)

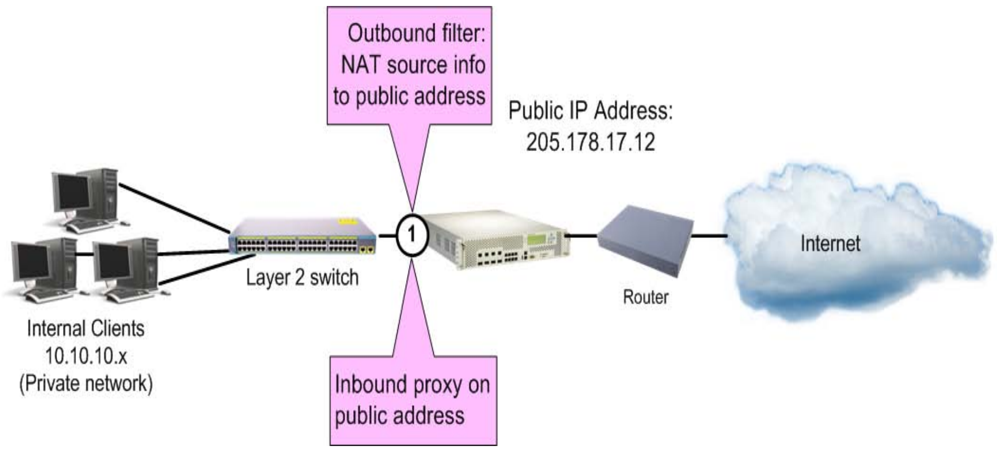
Notes

- Within each filter, the smask and dmask values are identical.
- All parameters for both filters are identical except for the NAT direction. For Filter 10, **nat source** is used. For Filter 11, **nat dest** is used.
- Filters for static (non-proxy) NAT should take precedence over dynamic NAT filters (see Dynamic NAT, page 386). Static filters should be given lower filter numbers.
- After port filtering is enabled or disabled and you apply the change, session entries are deleted immediately.

Dynamic NAT

Dynamic NAT is a many-to-one solution. Multiple clients on the private subnet take advantage of a single external IP address, thus conserving valid IP addresses. In the example in Figure 61 - Dynamic NAT Example, page 386, clients on the internal private network require TCP/UDP access to the Internet:

Figure 61: Dynamic NAT Example



You may directly connect the clients to Alteon if the total number of clients is less than or equal to the ports.

Note: Dynamic NAT can also be used to support ICMP traffic for PING.

This example requires a NAT filter to be configured on the port that is connected to the internal clients. When the NAT filter is triggered by outbound client traffic, the internal private IP address information on the outbound packets is translated to a valid, publicly advertised IP address on Alteon. In addition, the public IP address must be configured as a proxy IP address on the Alteon port that is connected to the internal clients. The proxy performs the reverse translation, restoring the private network addresses on inbound packets.

To configure dynamic NAT

`>> # /cfg/slb/filt 14`	(Select the menu for client filter)
`>> Filter 14# invert ena`	(Invert the filter logic)
`>> Filter 14# dip 10.10.10.0`	(If the destination is not private)
`>> Filter 14# dmask 255.255.255.0`	(For the entire private subnet range)
`>> Filter 14# sip any`	(From any source IP address)
`>> Filter 14# action nat`	(Perform NAT on matching traffic)
`>> Filter 14# nat source`	(Translate source information)
`>> Filter 14# ena`	(Enable the filter)
`>> Filter 14# adv/proxyadv/proxy enable`	(Enable client proxy on this filter)
`>> Filter 14 Proxy Advanced# proxyip 205.178.17.12`	(Set the filter's proxy IP address)
`>> Filter 14 Advanced# /cfg/slb/port 1`	(Select SLB port 1)
`>> SLB port 1# add 14`	(Add the filter 14 to port 1)
`>> SLB port 1# filt enable`	(Enable filtering on port 1)
`>> SLB port 1# proxy ena`	(Enable proxies on this port)
`>> SLB port 1# apply`	(Apply configuration changes)
`>> SLB port 1# save`	(Save configuration changes)

For more information on proxy IP address, see Client Network Address Translation (Proxy IP), page 190.

Notes

- The **invert** option in this example filter makes this specific configuration easier, but is not a requirement for dynamic NAT.
- Filters for dynamic NAT should be given a higher numbers than any static NAT filters (see Static NAT, page 384).
- After port filtering is enabled or disabled and you apply the change, session entries are deleted immediately.

FTP Client NAT

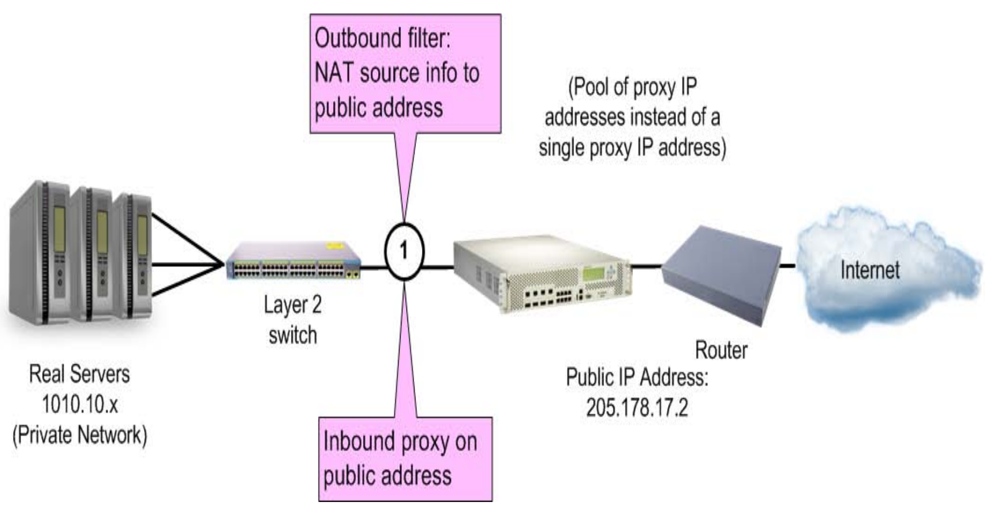
Alteon provides NAT services to many clients with private IP addresses. An FTP enhancement lets you perform true FTP NAT for dynamic NAT.

Because of the way FTP works in active mode, a client sends information on the control channel (information that reveals their private IP address) out to the Internet. However, the filter only performs NAT translation on the TCP/IP header portion of the frame, preventing a client with a private IP address from performing active FTP.

Alteon can monitor the control channel and replace the client 's private IP address with a proxy IP address defined on Alteon. When a client in active FTP mode sends a port command to a remote FTP server, Alteon analyzes the data part of the frame and modifies the **port** command as follows:

- The real server (client) IP address is replaced by a public proxy IP address.
- The real server (client) port is replaced with a proxy port.

Figure 62: FTP Client NAT Example



You may directly connect the real servers to Alteon if the total number of servers is less than or equal to the ports.

To configure active FTP client NAT

Note: The passive mode does not need to use this feature.

1. Make sure that a proxy IP address is enabled on the filter port.
2. Make sure that a source NAT filter is set up for the port:

`>> # /cfg/slb/filt 14`	(Select the menu for client filter)
`>> Filter 14# invert ena`	(Invert the filter logic)
`>> Filter 14# dip 10.10.10.0`	(If the destination is not private)
`>> Filter 14# dmask 255.255.255.0`	(For the entire private subnet range)

`>> Filter 14# sip any`	(From any source IP address)
`>> Filter 14# action nat`	(Perform NAT on matching traffic)
`>> Filter 14# nat source`	(Translate source information)
`>> Filter 14# ena`	(Enable the filter)
`>> Filter 14# adv/proxyadv/proxy enable`	(Allow proxy IP translation)
`>> Filter 14 Proxy Advanced# proxyip 205.178.17.12`	(Set the filter's proxy IP address)
`>> Proxy IP Address# /cfg/slb/port 1`	(Select SLB port 1)
`>> SLB port 1# add 14`	(Add the filter to port 1)
`>> SLB port 1# filt enable`	(Enable filtering on port 1)
`>> SLB port 1# proxy ena`	(Enable proxies on this port)
`>> SLB port 1# apply`	(Apply configuration changes)
`>> SLB port 1# save`	(Save configuration changes)

Note: After port filtering is enabled or disabled and you apply the change, session entries are deleted immediately.

For more information on proxy IP address, see Client Network Address Translation (Proxy IP), page 190.

3. Enable active FTP NAT using the following command:

```
>> # /cfg/slb/filt <filter number> /adv/layer7/ftpa ena
```

4. Apply and save the configuration.

Overlapping NAT

Alteon supports overlapping or duplicate source IP addresses on different VLANs in a source NAT filter. This is done by extending the session table lookup algorithm to include the session VLAN.

When there is an overlapping source IP address for different VLANs, Alteon creates different sessions. For the source NAT, Alteon substitutes the source IP address with the configured proxy IP address. A proxy IP address for the VLAN must be configured for this to function properly.

When there is an overlapping NAT, Alteon does not use the routing table to route the packet back to the sender in Layer 3 mode, due to the overlapping source address. Instead, Alteon uses the VLAN gateway to forward the packet back to the sender. While VLAN gateway configuration is necessary to make this feature function properly, Layer 2 mode is also supported.

To configure overlapping NAT

1. Configure a gateway per VLAN. Default Gateway 5 or above must be used for the VLAN gateway, as gateways 1 through 4 are reserved for default gateways.

```
>> Main# /cfg/l3/gw 5
>> Default Gateway 5# addr <IP address>
>> Default Gateway 5# vlan 100
```

2. Configure the source NAT filter. Select the appropriate filter. In this example, Filter 2 is used.

```
>> Main# /cfg/slb/filt 2/action na
```

3. Enable overlapping NAT.

```
>> Main# /cfg/slb/adv/pvlantag enable
```

SIP NAT and Gleaning Support

The IP end points on a network are typically assigned private addresses. Voice calls from and to the public network need to reach end points on the private network. As a result, NAT is required to allow proper routing of media to end points with private addresses.

The Session Initiation Protocol (SIP) carries the identification of the IP end point (IP address and port) within the body of the message. The voice media which gets directed to the private IP address identified in the signaling message cannot be routed and results in a one-way path. Therefore, Alteon allows you to translate the address (using NAT) for the Session Description Protocol (SDP) and create sessions for the media communication.

How SIP NAT Works

All occurrences of the internal client's private IP address and port in the outgoing SIP message is replaced with the translated address. This procedure is reversed when the SIP messages come from an external source, in which case the public IP is replaced with the private client's IP and port. Alteon translates the IP address and port.

Setting Up SIP NAT

To set up SIP NAT, configure a NAT filter and enable SIP parsing. The SIP NAT modifies the signaling to reflect the public IP addresses and ports. These pinholes and NAT bindings are assigned dynamically based on stateful inspection. The SIP NAT performs the necessary translation of the IP addresses embedded in the SIP messages and updates the SIP message before sending the packet out.

To support SIP NAT and gleaning

1. Enable VMA.
2. Configure a NAT filter.

Note: Dynamic NAT is supported only.

```
>> Main# /cfg/slb/filt 14
>> Filter 14# action nat
>> Filter 14# nat source
```

3. Enable SIP parsing.

```
>> Main# /cfg/slb/filt 14
>> Filter 14# adv
>> Filter 14 Advanced# Layer7
>> Layer 7 Advanced# sip
>> Layer 7 SIP# sipp ena
```

4. Set a BWM contract for the SIP RTP sessions.

```
>> Layer 7 SIP# rtpcont <contract #>
```

5. Apply and save the configuration.

```
>> Layer 7 SIP# apply
>> Layer 7 SIP# save
```

Note: When MCS proxy authentication is enabled, the MCS PC client creates message digests using the client's private address. These digests are sent back to the MCS server for authentication during the *invite* stage. Call setup fails with MCS proxy authentication enabled as Alteon does not regenerate these message digests with the public address.

Matching TCP Flags

This section describes the ACK filter criteria, which provides greater filtering flexibility. Alteon supports packet filtering based on any of the following TCP flags.

Table 32: Supported TCP Flags

Flag	Description
URG	Urgent
ACK	Acknowledgement
PSH	Push
RST	Reset
SYN	Synchronize
FIN	Finish

Any filter may be set to match against more than one TCP flag at the same time. If there is more than one flag enabled, the flags are applied with a logical AND operator. For example, by setting Alteon to filter **SYN** and **ACK**, Alteon filters all SYN-ACK frames.

Notes

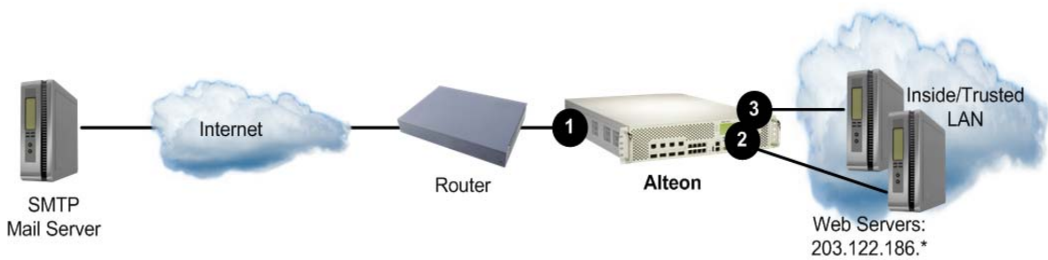
- TCP flag filters must be cache-disabled. Exercise caution when applying cache-enabled and cache-disabled filters to the same port. For more information, see Cached Versus Non-Cached Filters, page 362.
- With IPv6, TCP health checks end with an RST flag instead of FIN as in IPv4.

Configuring the TCP Flag Filter

By default, all TCP filter options are disabled. TCP flags are *not* inspected unless one or more TCP options are enabled.

Consider the network as illustrated in Figure 63 - TCP Flag Filter Configuration Example, page 392.:

Figure 63: TCP Flag Filter Configuration Example



In this network, the Web servers inside the LAN must be able to transfer mail to any SMTP-based mail server out on the Internet. At the same time, you want to prevent access to the LAN from the Internet, except for HTTP.

SMTP traffic uses well-known TCP port 25. The Web servers originates TCP sessions to the SMTP server using TCP destination port 25, and the SMTP server acknowledges each TCP session and data transfer using TCP source port 25.

Creating a filter with the ACK flag closes one potential security hole. Without the filter, Alteon permits a TCP SYN connection request to reach any listening TCP destination port on the Web servers inside the LAN, as long as it originated from TCP source port 25. The server would listen to the TCP SYN, allocate buffer space for the connection, and reply to the connect request. In some SYN attack scenarios, this could cause the server's buffer space to fill, crashing the server or at least making it unavailable.

A filter with the ACK flag enabled prevents external devices from beginning a TCP connection (with a TCP SYN) from TCP source port 25. Alteon drops any frames that have the ACK flag turned off.

To configure TCP flag filters

This procedure is based on Figure 63 - TCP Flag Filter Configuration Example, page 392.

1. Configure an allow filter for TCP traffic from the LAN that allows the Web servers to pass SMTP requests to the Internet.

`>> # /cfg/slb/filt 10`	(Select a filter for trusted SMTP requests)
`>> Filter 10# sip 203.122.186.0`	(From the Web servers' source IP address)
`>> Filter 10# smask 255.255.255.0`	(For the entire subnet range)
`>> Filter 10# sport any`	(From any source port)
`>> Filter 10# proto tcp`	(For TCP traffic)
`>> Filter 10# dip any`	(To any destination IP address)
`>> Filter 10# dport smtp`	(To well-known destination SMTP port)
`>> Filter 10# action allow`	(Allow matching traffic to pass)
`>> Filter 10# ena`	(Enable the filter)

2. Configure a filter that allows SMTP traffic from the Internet to pass through Alteon only if the destination is one of the Web servers, and the frame is an acknowledgment (SYN-ACK) of a TCP session.

`>> Filter 10# /cfg/slb/filt 15`	(Select a filter for Internet SMTP ACKs)
`>> Filter 15# sip any`	(From any source IP address)
`>> Filter 15# sport smtp`	(From well-known source SMTP port)
`>> Filter 15# proto tcp`	(For TCP traffic)
`>> Filter 15# dip 203.122.186.0`	(To the Web servers' IP address)
`>> Filter 15# dmask 255.255.255.0`	(To the entire subnet range)
`>> Filter 15# dport any`	(To any destination port)
`>> Filter 15# action allow`	(Allow matching traffic to pass)
`>> Filter 15# ena`	(Enable the filter)
`>> Filter 15# adv/tcp`	(Select the advanced TCP menu)
`>> Filter 15 Advanced# ack ena`	(Match acknowledgments only)
`>> Filter 15 Advanced# syn ena`	(Match acknowledgments only)

3. Configure a filter that allows SMTP traffic from the Internet to pass through Alteon only if the destination is one of the Web servers, and the frame is an acknowledgment (ACK-PSH) of a TCP session.

`>> Filter 15# /cfg/slb/filt 16`	(Select a filter for Internet SMTP ACKs)
`>> Filter 16# sip any`	(From any source IP address)
`>> Filter 16# sport smtp`	(From well-known source SMTP port)
`>> Filter 16# proto tcp`	(For TCP traffic)
`>> Filter 16# dip 203.122.186.0`	(To the Web servers' IP address)

`>> Filter 16# dmask 255.255.255.0`	(To the entire subnet range)
`>> Filter 16# dport any`	(To any destination port)
`>> Filter 16# action allow`	(Allow matching traffic to pass)
`>> Filter 16# ena`	(Enable the filter)
`>> Filter 16# adv/tcp`	(Select the advanced TCP menu)
`>> Filter 16 Advanced# ack ena`	(Match acknowledgments only)
`>> Filter 16 Advanced# psh ena`	(Match acknowledgments only)

4. Configure a filter that allows trusted HTTP traffic from the Internet to pass through Alteon to the Web servers.

`>> Filter 16 Advanced# /cfg/slb/filt 17`	(Select a filter for incoming HTTP traffic)
`>> Filter 17# sip any`	(From any source IP address)
`>> Filter 17# sport http`	(From well-known source HTTP port)
`>> Filter 17# proto tcp`	(For TCP traffic)
`>> Filter 17# dip 203.122.186.0`	(To the Web servers' IP address)
`>> Filter 17# dmask 255.255.255.0`	(To the entire subnet range)
`>> Filter 17# dport http`	(To well-known destination HTTP port)
`>> Filter 17# action allow`	(Allow matching traffic to pass)
`>> Filter 17# ena`	(Enable the filter)

5. Configure a filter that allows HTTP responses from the Web servers to pass through Alteon to the Internet.

`>> Filter 17# /cfg/slb/filt 18`	(Select a filter for outgoing HTTP traffic)
`>> Filter 18# sip 203.122.186.0`	(From the Web servers' source IP address)
`>> Filter 18# smask 255.255.255.0`	(From the entire subnet range)
`>> Filter 18# sport http`	(From well-known source HTTP port)
`>> Filter 18# proto tcp`	(For TCP traffic)
`>> Filter 18# dip any`	(To any destination IP address)
`>> Filter 18# dport http`	(To well-known destination HTTP port)
`>> Filter 18# action allow`	(Allow matching traffic to pass)
`>> Filter 18# ena`	(Enable the filter)

6. Configure a default filter which denies all other traffic. This filter is required.

`>> Filter 18# /cfg/slb/filt 2048`	(Select a default filter)
`>> Filter 2048# sip any`	(From any source IP address)
`>> Filter 2048# dip any`	(To any destination IP address)
`>> Filter 2048# action deny`	(Block matching traffic)
`>> Filter 2048# name deny matching traffic`	(Provide a descriptive name for the filter)
`>> Filter 2048# ena`	(Enable the filter)

7. Apply the filters to the appropriate ports.

`>> Filter 2048# /cfg/slb/port 1`	(Select the Internet-side port)
`>> SLB port 1# add 15`	(Add the SMTP ACK filter to the port)
`>> SLB port 1# add 16`	(Add the incoming HTTP filter)
`>> SLB port 1# add 17`	(Add the incoming HTTP filter)
`>> SLB port 1# add 2048`	(Add the default filter to the port)
`>> SLB port 1# filt ena`	(Enable filtering on the port)
`>> SLB port 1# /cfg/slb/port 2`	(Select the first Web server port)
`>> SLB port 2# add 10`	(Add the outgoing SMTP filter to the port)
`>> SLB port 2# add 18`	(Add the outgoing HTTP filter to the port)
`>> SLB port 2# add 2048`	(Add the default filter to the port)
`>> SLB port 2# filt ena`	(Enable filtering on the port)
`>> SLB port 2# /cfg/slb/port 3`	(Select the other Web server port)
`>> SLB port 3# add 10`	(Add the outgoing SMTP filter to the port)
`>> SLB port 3# add 18`	(Add the outgoing HTTP filter to the port)
`>> SLB port 3# add 2048`	(Add the default filter to the port)
`>> SLB port 3# filt ena`	(Enable filtering on the port)
`>> SLB port 3# apply`	(Apply the configuration changes)
`>> SLB port 3# save`	(Save the configuration changes)

Note: After port filtering is enabled or disabled and you apply the change, session entries are deleted immediately.

Matching ICMP Message Types

This section describes the ICMP message types. The Internet Control Message Protocol (ICMP) is used for reporting TCP/IP processing errors. There are numerous types of ICMP messages, as shown in Table 33 - ICMP Supported Message Types, page 395. Although ICMP packets can be filtered using the `proto icmp` option, by default, Alteon ignores the ICMP message type when matching a packet to a filter. To perform filtering based on specific ICMP message types, ICMP message type filtering must be enabled.

Table 33: ICMP Supported Message Types

Type #	Message Type	Description
0	echorep	ICMP echo reply
3	destun	ICMP destination unreachable
4	quench	ICMP source quench

Table 33: ICMP Supported Message Types

Type #	Message Type	Description
5	redir	ICMP redirect
8	echoreq	ICMP echo request
9	rtradv	ICMP router advertisement
10	rtrsol	ICMP router solicitation
11	timex	ICMP time exceeded
12	param	ICMP parameter problem
13	timereq	ICMP timestamp request
14	timerep	ICMP timestamp reply
15	inforeq	ICMP information request
16	inforep	ICMP information reply
17	maskreq	ICMP address mask request
18	maskrep	ICMP address mask reply

To enable or disable ICMP message type filtering

```
>> # /cfg/slb/filt <filter number> /adv
>> Filter 1 Advanced# icmp any|<number>|<type; "icmp list" for list>
```

For any given filter, only one ICMP message type can be set at any one time. The **any** option disables ICMP message type filtering. The **list** option displays a list of the available ICMP message types that can be entered.

Note: ICMP message type filters must be cache-disabled. Exercise caution when applying cache-enabled and cache-disabled filters to the same port. For more information, see Cached Versus Non-Cached Filters, page 362.

Multicast Filter Redirection

Multicast Filter Redirection is used to redirect multicast packets based on filtering criteria. Before packets get redirected to the filter-specified server, Alteon substitutes the destination MAC address with the server MAC address. The modified packets are then sent to the port where the specified server is connected. Multicast packets are redirected without substituting the destination MAC address.

Since the destination MAC address and destination IP address need to be in same cast category, the redirected multicast or broadcast packets should keep the multicast type destination MAC address. In redirection filter processing, Alteon checks cast type of destination MAC address in the received packet. If the received packet is a unicast packet, the destination MAC address is substituted to the specified server's MAC address. Then the redirected unicast packet is sent to the port to where the server is connected. If the received packet is a multicast packet, the destination MAC address is not substituted. Then the redirected multicast packet is sent to the port that the server connected to.

IPv6 Filtering

Alteon IPv6 support includes support for filter classification and action up to Layer 4. Layer 7 classification and actions are not supported on IPv6 filters. IPv6 filtering operates in a similar fashion to IPv4 filtering.

Notes

- For NAT filters, the advanced PIP address configured within an IPv6 filter must also be IPv6.
- For an IPv6 redirection filter, the server group to which the filter redirects must contain only IPv6 servers.

Connectivity is maintained in IPv6 through the regular exchange of Neighbors Solicitation (NSol) packets. These packets are sent to find the link layer address of a neighbor in the link and to find the reachability of a neighboring node. It is usually necessary to configure an additional ALLOW filter for these multicast packets so that link neighbors can be learnt. If this is not done, no packets are allowed because link neighbors cannot be learnt. Filter inversion also must take these NSol packets into consideration.

Not all *Advanced* menu commands that are available for configuring IPv4 filters are available for configuring IPv6 filters. You can use the following *Advanced* menu commands to configure IPv6 filters:

Table 34: IPv6 Filter Configuration Commands

<table>
<tr><th>Command Menu</th><th>Supported Commands</th></tr>
<tr><td>/cfg/slb/filt <filter number> /adv</td><td>• cont <BW Contract, 1-1024>
• revcont <BW Contract, 1-1024>
• tmout <even number of minutes, 4-32768>
• idsgrp <real server group number, 1-1024>|none
• idshash sip|dip|both
• thash auto|sip|dip|both|sip+sport|dip32
• mcvlan <Vlan id>
• goto <filter ID>
• reverse disable|enable (or just d|e)
• cache disable|enable (or just d|e)
• log disable|enable (or just d|e)
• mirror disable|enable (or just d|e)
• nbind disable|enable (or just d|e)</td></tr>
<tr><td>/cfg/slb/filt <filter number> /adv/ip</td><td>• length <IP packet length (in bytes), 64-65535 | any></td></tr>
<tr><td>/cfg/slb/filt <filter number> /adv/tcp</td><td>All TCP menu commands.</td></tr>
</table>

Table 34: IPv6 Filter Configuration Commands

Command Menu	Supported Commands
`/cfg/slb/filt <filter Number> /adv/ 8021p`	All *802.1p* menu commands.
`/cfg/slb/filt <filter Number> /adv/ proxyadv`	All *Proxy* menu commands.
`/cfg/slb/filt <filter Number> /adv/ redir`	All *Redirection* menu commands.
`/cfg/slb/filt <filter Number> /adv/ security/ratelim`	All *Rate Limiting* menu commands.

The following example creates two IPv6 filters for Port 1. Filter 1 allows the exchange Neighbors Solicitation packets, and Filter 2 allows the movement of bridged HTTP traffic.

Example IPv6 Filtering Example

1. Globally enable Layer 4 load balancing. Layer 4 load balancing must be enabled to allow filter processing to take place.

```
>> Main# /cfg/slb/on
```

2. Create Filter 1 to allow the passage of Neighbors Solicitation packets.

```
>> Main# /cfg/slb/filt 1/ena                 (Enable Filter 1)
>> Filter 1# action allow                    (Specify an ALLOW filter)
>> Filter 1# ipver v6                        (Specify an IPv6 filter)
>> Filter 1# sip 2001:0:0:0:0:0:0:0          (Specify source IP)
>> Filter 1# smask 64                        (Specify IPv6 source prefix)
>> Filter 1# dip ff00:0:0:0:0:0:0:0          (Specify destination IP)
>> Filter 1# dmask 8                         (Specify IPv6 destination prefix)
>> Filter 1# vlan any                        (Specify VLAN settings)
```

3. Create Filter 2 to allow the movement of bridged HTTP traffic.

s

```
>> Main# /cfg/slb/filt 2/ena                 (Enable Filter 2)
>> Filter 2# action allow                    (Specify an ALLOW filter)
>> Filter 2# ipver v6                        (Specify an IPv6 filter)
>> Filter 2# sip 2001:0:0:0:0:0:0:1          (Specify source IP)
>> Filter 2# smask 128                       (Specify IPv6 source prefix)
>> Filter 2# dip 2001:0:0:0:0:0:0:8          (Specify destination IP)
>> Filter 2# dmask 128                       (Specify IPv6 destination prefix)
>> Filter 2# proto tcp                       (Specify filter protocol)
>> Filter 2# sport any                       (Specify source port)
```

`>> Filter 2# dport http`	(Specify destination port)
`>> Filter 2# vlan any`	(Specify VLAN settings)

4. Add the two filters to Port 1.

`>> Main# /cfg/slb/port 1`	(Select Port 1)
`>> Port 1# filt ena`	(Enable port filtering)
`>> Port 1# add 1-2`	(Add Filters 1 and 2 to Port 1)

Content Class Filters for Layer 7 Traffic

Alteon filters serve as traffic classifiers for Layers 2 through 4. The integration of the Application Acceleration module with Alteon filters extends this functionality to Layer 7, and provides complete service transparency for users.

The following topics are discussed in this section:

- Content Class Overview, page 399
- Defining a Content Class, page 400
- Assigning a Content Class to Filters, page 401

Content Class Overview

The content class is a matching object used for Layer 7 content switching rules. You can define a set of matching criteria that are based on the application type. For example, with an HTTP class, you can define matching criteria based on HTTP protocol elements such as URL, HTTP headers, and so on. Each element can have multiple matching values, enabling advanced matching decisions to be evaluated. For example, "if (URL=my-site.com OR URL=my-site2.com) AND (Header=User-Agent: Internet-Explorer)".

Content classes can be nested using logical expressions. This enables you to use one class as part of the matching criteria for another class. For example, Class A includes a list of 100 mobile phone browser types. Classes B, C, and D need to match specific URLs for all the mobile phones from Class A. To configure this, Class A is defined as a logical expression matching the criteria of Classes B, C, and D. When you need to add additional mobile phone browsers to the list, you add them to Class A, and they are then propagated to Classes B, C, and D.

For more information, see Content-Intelligent Server Load Balancing, page 219.

Notes

- Alteon supports Layer 7 content switching using an additional legacy configuration model that is based on Layer 7 strings. For related examples based on using Layer 7 strings see Appendix B - Content-Intelligent Server Load Balancing Not Using Layer 7 Content Switching Rules, page 809.
- To support IP fragment traffic when Layer 7 content switching is defined based on strings, use the **forceproxy** command under `/cfg/slb/virt/service/dbind` to force traffic through the Application Services Engine.

  For more information, see `/cfg/slb/virt<server number>/service <virtual port or application name>/dbind/forceproxy` option in the *Alteon Application Switch Operating System Command Reference*.

In earlier versions of Alteon, filters are tied to content rules. The content rules act as a link to virtual services. Alteon version 29 lets you assign content classes to Layer 7 filtering, freeing content rules for use in a classification library.

Defining a Content Class

This section describes how to define a new content class.

To configure a content class

1. Select the **cntclss** option.

```
>> Main# /cfg/slb/layer7/slb/cntclss
```

2. Set an ID and class type for the content class.

```
>> vADC 1 - Server Load balance Resource# cntclss
Enter Class id: myclass
```

The *Content Class* menu displays.

```
[HTTP Content Class myclass Menu]
     name     - Set the Descriptive HTTP content class name
     hostname - URL Hostname lookup Menu
     path     - URL Path lookup Menu
     filename - URL File Name lookup Menu
     filetype - URL File Type lookup Menu
     header   - Header lookup Menu
     cookie   - Cookie lookup Menu
     text     - Text lookup Menu
     xmltag   - XML tag lookup Menu
     logexp   - Set logical expression between classes
     copy     - Copy HTTP content class
     del      - Delete HTTP content class
     cur      - Display current HTTP content class
```

3. Define the following class classes:
   - URL hostname
   - URL path
   - URL file name
   - URL file type
   - header
   - cookie
   - general text
   - XML tag

Assigning a Content Class to Filters

This section describes how to assign a content class to one or more filters.

To assign a content class to one or more filters

1. Select the **cntclss** option.

```
>> Main# /cfg/slb/filt <filter number>/cntclss
```

2. Add the the content class to one or more filters.

```
>> Filter 10  # cntclss
Current cntclss ID: 1-5,40
Enter new cntclss ID: 1-6,40
```

Return-to-Sender

Enabling Return-to-Sender (RTS) allows Alteon to associate the session with the MAC address of the WAN router. This ensures that the returning traffic takes the same ISP path as the incoming traffic. RTS is enabled on the incoming WAN ports (port 2 and 7) to maintain persistence for the returning traffic. Data leaves Alteon from the same WAN link that it used to enter, thus maintaining persistency.

Note: As of version 29.0, the RTS method has been superseded by Transparent Load Balancing. For best results, Radware recommends that you use Transparent Load Balancing. For more information, see Transparent Load Balancing, page 364.

Chapter 16 – ADC-VX Management

This chapter discusses how to use ADC-VX™ in an Alteon environment, and includes the following topics:

What is ADC-VX?

ADC-VX is a specialized Application Delivery Controller (ADC) hypervisor that runs multiple virtual ADC instances on dedicated ADC hardware, Radware's OnDemand Switch platforms. ADC-VX is built on a unique architecture that virtualizes the OnDemand Switch resources—including CPU, memory, network, and acceleration resources. This specialized hypervisor runs fully functional virtual ADC instances, each of which delivers ADC functionality just like a dedicated physical ADC. Each virtual ADC instance contains a complete and separated environment of resources, configurations and management.

ADC Form Factors

ADC-VX supports three different ADC form factors:

- **Dedicated ADC**—The traditional Alteon hardware ADC.
- **vADC**—A virtualized instance of the Alteon operating system (AlteonOS).
- **Alteon VA**—A software-based ADC supporting AlteonOS functionality and running on the VMware virtual infrastructure. For more information, see the *Radware Alteon ADC-VA Release Notes* and *Radware Alteon ADC-VA Quick Install Guide*.

You can save and back up configurations from and to different form factors. For more information, see Importing the Active ADC Configuration, page 429, and Backing Up the Active vADC Configuration, page 433.

vADCs

A vADC is a virtualized instance of the AlteonOS that behaves in the same manner as a traditional Alteon hardware ADC, with the exception that while it is bound to a specific hardware resource, the amount of resources allocated to the vADC may vary based on the user’s or application's resource needs. This enables you to run multiple independent and private vADCs that vary in their processing power.

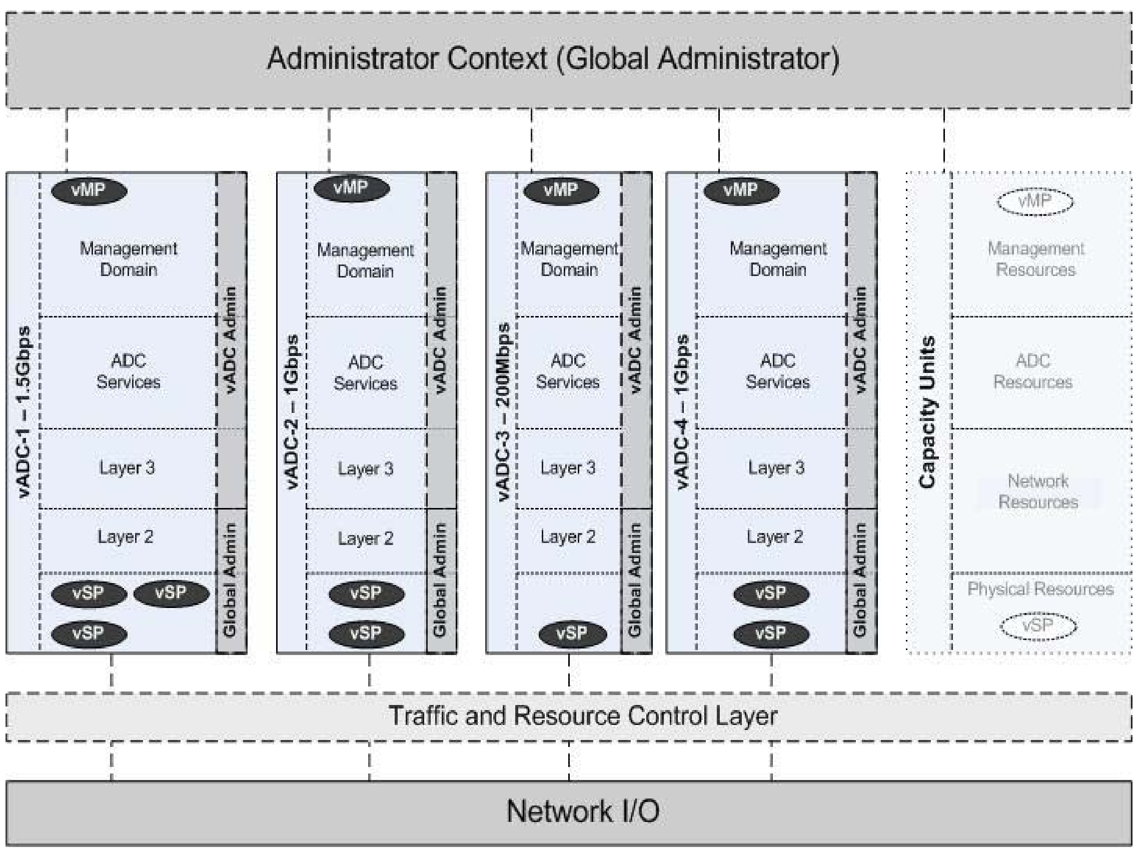
Each vADC comprises a vSP (Virtualized Switch Processor) and a vMP (Virtualized Management Processor), providing the vADCs with their own set of resources, network infrastructure, and services that are completely independent of neighboring vADCs. This enables multiple users to run vADCs and allocate resources to these vADCs without introducing any risk to the other vADCs within the shared physical environment.

vADC management is divided between two management roles:

- The *Global Administrator* creates, initially configures, and monitors vADCs. In addition, one of the main tasks of the Global Administrator is to dynamically allocate CPU and throughput resources by assigning capacity units and adjusting throughput limits to a vADC. For more details on capacity units and throughput, see Allocating and Removing Processing Power (Capacity Units) and Throughput Resources, page 406. For more details on the Global Administrator's tasks, see Global Administrator, page 405.
- The *vADC Administrator* is responsible for the day-to-day configuration and maintenance of vADCs using the same tasks as with traditional ADCs, except for those vADC tasks that only the Global Administrator performs. For more details on the vADC Administrator's tasks, see vADC Administrator, page 409.

The following is an illustration of a network architecture configured to use ADC-VX:

Figure 64: Network Architecture Configured to use ADC-VX



vADC Management

As opposed to traditional ADC management, vADC management is divided between two management roles:

- Global Administrator, page 405
- vADC Administrator, page 409

This section also discusses the following topic:

- Resource Management, page 410

Global Administrator

The Global Administrator is a superuser that works at a management level above and separate from a vADC Administrator. The Global Administrator manages the physical Alteon resources and uses the physical devices in a data center, is responsible for creating vADC instances, and manages and monitors both system and vADC resource allocation and utilization. The Global Administrator does not manage Layer 3 or SLB functionality, but rather they are managed by the vADC Administrator. The Global Administration environment is only accessible through the out-of-band management ports.

The basic tasks and responsibilities of the Global Administrator include the following:

- Managing vADCs, page 405
- Monitoring Health and Resource Usage, page 406
- Allocating and Removing Processing Power (Capacity Units) and Throughput Resources, page 406

The following are additional tasks the Global Administrator performs:

- Assigning Initial User Access, page 406
- Configuring and Maintaining Management Ports, page 406
- Delegating System Services, page 407
- Synchronizing vADCs, page 407

Managing vADCs

The Global Administrator creates and deletes vADCs. The number of vADCs and their overall capacity and throughput are based on the installed vADC and throughput licenses. Throughput can be allocated to vADCs in increments of 1 Mbps. The maximum number of vADCs that can be created is platform-specific from 24–256.

For an example procedure for creating and configuring vADC, see Creating a New vADC, page 419. For more details on creating vADCs, see the section on the **/cfg/vadc** menu in the *Alteon Application Switch Operating System Command Reference.*

For a discussion of allocating resources, see Allocating and Removing Processing Power (Capacity Units) and Throughput Resources, page 406.

Monitoring Health and Resource Usage

The Global Administrator regularly monitors the system for application resource consumption and average throughput. Each vADC has an accompanying dashboard that aggregates the status of the configured vADC. The dashboard is only accessible through the BBI. The Global Administrator uses the dashboard to verify the health, resource usage, and activity of the vADC. For more information on the dashboard, see the Resource Dashboard, page 411.

Allocating and Removing Processing Power (Capacity Units) and Throughput Resources

As a result of monitoring health and resource usage, the Global Administrator may want to readjust the amount of processing power and throughput resources allocated per application. These resources are allocated by assigning capacity units to the vADC.

A capacity unit represents the amount of processing power and throughput that is assigned to a vADC.

Table 35 lists the total number of capacity units that can be assigned to a vADC, and the maximum throughput per capacity unit.

Table 35: Capacity Unit Throughput Limits

Platform	Maximum number of CUs that can be assigned to a vADC	Maximum throughput (Mbps) per CU
5224	24	666

Capacity units can be assigned to vADCs regardless of throughput requirements and only for the purpose of increasing processing power. For example, an application that is assigned a policy that requires a large amount of processing power does not necessarily require more throughput. For such an application, you can increase the available processing power without having to adjust the allocated throughput.

You can assign multiple capacity units to a vADC from the available capacity units in the pool of global capacity units.

After initially assigning a capacity unit, you can add or remove throughput in 100 Mb increments up until the amount of available throughput, based on the total amount of your installed throughput license.

To adjust the number of capacity units, you must first shut down (disable) the vADC. After making the adjustment, for the change to take effect, you then power on (enable) the vADC.

For more details, see the **`/cfg/vadc/cu`** command in the *Alteon Application Switch Operating System Command Reference.* For an example procedure, see step 5.

Assigning Initial User Access

The Global Administrator assigns initial access to vADCs, including the vADC Administrator, using the **`/cfg/vadc/users/uid`** menu. For more information, see the *Alteon Application Switch Operating System Command Reference.*

Configuring and Maintaining Management Ports

The Global Administrator is responsible for the initial vADC settings, including user access methods. Additionally, the Global Administrator can control the access method in which a vADC is accessed, such as limiting access through SSH and/or HTTPS. These settings can be changed by the vADC Administrator if the Global Administrator allows for this.

For more details on configuring and maintaining management ports in the vADC environment, see the section on the **`/cfg/sys/mmgmt`** menu in the *Alteon Application Switch Operating System Command Reference*.

Delegating System Services

If the Global Administrator wants to enforce a global policy across vADCs, the Global Administrator can enforce specific service usage. For example, an organization that requires authentication using AAA servers, or requires information collection for security purpose, might want to both enforce (delegate) these settings globally and lock them for modification by the vADC Administrator. For each of these system services, the Global Administrator can either enable or disable them for modification.

The system services that the Global Administrator can delegate include:

- Syslog server
- AAA Services
  - RADIUS server
  - TACACS server
- Time Services (NTP)
- Timeout for idle CLI sessions
- vADC Management IP settings
- Management access protocols
- SMTP services

For more details, see the section on the `/cfg/vadc/sys` menu in the *Alteon Application Switch Operating System Command Reference*.

Synchronizing vADCs

Environments using ADC-VX usually take advantage of a least one additional Alteon for redundancy purposes. ADC-VX supports solution designs constructed with up to six peers for redundancy and risk distribution. A Global Administrator managing the system is required to define a vADC only once, while the system synchronizes all the settings to one of the peers. The system is aware of the location of all vADCs and their peers at all times and performs the configuration synchronization based on the location of the target vADC. Therefore, there is no need to keep track of or make modifications in multiple locations. The synchronization mechanism creates new vADCs, synchronizes changes, and adapts to any modification.

Each ADC-VX platform supports synchronization with up to five peers. Each system is aware of the location of each vADC at any given time. This enables the contextual synchronization of all changed configuration information to the relevant Alteon without manual intervention or any unnecessary operations. To use this feature, you perform the following tasks:

- Define the IP information of Alteons in the system. The IP address that is used for synchronization is the IP address of the Global Administrator management access.
- Assign each vADC with a peer ID using **`/cfg/vadc #/sys/sync`**.

For more details, see the section on the `/cfg/sys/sync` and `/oper/sync` commands in the *Alteon Application Switch Operating System Command Reference*.

Note: ADC-VX also supports bulk vADC peer configuration using the **`range`** command available under **`/cfg/sys/sync/peer #/range`**. For more details, see the *Alteon Application Switch Operating System Command Reference*.

Backing Up and Restoring vADCs

ADC-VX supports multiple backup and restore mechanisms for quick and efficient disaster recovery. vADCs are entities that can be exported and imported in their entirety, similar to virtual machines. The exported vADCs can be imported to any site or ADC-VX platform available for recovery or for simple service creation.

The Global Administrator has the following options for backing up and restoring vADC configurations:

- **Backup and recovery of vADC**—Backup of a vADC and, upon disaster, recovery of the backed up vADC to any location with an active ADC-VX platform, with a simple import action (no configuration necessary).
- **Export of vADC**—Export a vADC and template creation for quick service creation.
- **Global backup and restore**—All elements are backed up, including the Global Administration configuration (vADCs, allocated resource, system settings, and so on) and all vADC configurations files.
- **Selective vADC backup and restore**—Individual vADC configurations are backed up.
- **Global Administrator infrastructure backup and restore**—The Global Administrator configuration is backed up, but not the vADC configuration files.

For more details, see the section on the **`/cfg/ptcfg`** and **`/cfg/gtcfg`** commands in the *Alteon Application Switch Operating System Command Reference.*

Integrating vADCs into a Shared Network Design

A shared external interface is a connectivity option that is designed to simplify the integration of vADCs into existing environments and avoid risky and invasive changes to the existing infrastructure. Shared interfaces are dedicated tagged or untagged ports that can be assigned to one or more vADCs as a new interface type.

A shared interface consolidates multiple private vADC communications links with a shared physical network. Even though each vADC instance is virtualized, they appear and perform in the same manner as physical ADCs, having dedicated MAC addresses and establishing relationships with adjacent network ADCs.

To minimize risk when integrating vADCs into a network infrastructure, a shared interface enables you to integrate into the existing infrastructure without having to make configuration changes or to allocate new subnets or VLAN IDs. A shared external interface further benefits integration by enabling you to mirror the connectivity of physical ADCs with the a shared infrastructure.

When you assign a shared external interface to vADCs, the vADCs share a VLAN in the same way that ADCs in a physical network do. When you set a vADC to be part of a shared network, the vADC is assigned a virtual MAC address. Both the VLAN (subnet IP) and virtual MAC addresses are visible to the network and the Internet in the same way that the VLAN and physical MAC addresses are visible in a traditional ADC design.

You can also have a mixed environment where some of your vADCs are part of the shared network, while others are not. You may do this, for example, if you want to first test a new vADC configuration before integrating it into your shared network.

To configure a vADC to be part of a shared network, you set the `/cfg/l2/vlan/shared` command to enabled. For an example configuration, see Assigning a VLAN Shared Interface to a vADC, page 428.

vADC Administrator

The vADC Administrator manages Layer 3 and SLB functionality controlling the service and/or application policies and performance. Configuration and management of physical ADCs are handled only by the Global Administrator.

The basic tasks and responsibilities of the vADC Administrator include the following:

- Configuring vADCs, page 409
- Configuring and Maintaining Management Ports, page 409
- Delegating System Services, page 409
- Locking and Unlocking Delegated Services, page 410
- Monitoring and Maintaining vADCs, page 410
- Synchronizing vADCs, page 410

Configuring vADCs

The vADC is responsible for vADC configuration and management. This is done in the same manner as a traditional standalone ADC, except for those features and functions which are reserved for the Global Administrator. For more details on the Global Administrator tasks and responsibilities, see Global Administrator, page 405).

The vADC Administrator can override many of the Global Administrator settings for individual vADCs. For example, under the **/cfg/sys/mmgmt** menu, the vADC Administrator can set different IP and subnet addresses than were defined by the Global Administrator.

Configuring and Maintaining Management Ports

The Global Administrator is responsible for the initial vADC settings, including user access methods. Additionally, the Global Administrator can control the access method in which a vADC is accessed, such as limiting access through SSH and/or HTTPS. These settings can be changed by the vADC Administrator if the Global Administrator allows for this.

For more details on configuring and maintaining management ports in the vADC environment, see the section on the **/cfg/sys/mmgmt** menu in the *Alteon Application Switch Operating System Command Reference*.

Delegating System Services

When vADCs are first created by the Global Administrator, all vADCs inherit the system services settings as defined by the Global Administrator. If the Global Administrator has enabled the vADC Administrator to modify the settings on any of these system services, the vADC Administrator can change the settings for individual vADCs as required (for example, this is a way to gain privacy and segregation between vADCs).

There are two options for how a vADC Administrator delegates system services:

- Use the dedicated services that the vADC Administrator defines.
- Inherit the dedicated services that the Global Administrator defines. If the Global Administrator has locked the global system services, the vADC Administrator can only use the services as defined by the Global Administrator.

The system services that the vADC Administrator can change, if unlocked, include:

- Syslog server
- AAA Services
  — RADIUS server
  — TACACS server
- Time Services (NTP)
- Timeout for idle CLI sessions

- vADC Management IP settings
- Management access protocols
- SMTP services

For more details, see the section on the `/cfg/sys` menu in the *Alteon Application Switch Operating System Command Reference*.

Locking and Unlocking Delegated Services

This feature enables the Global Administrator to lock any service that was delegated to a vADC, preventing the vADC Administrator from changing them. Each delegated service can be individually locked, enabling the Global Administrator to have more flexibility and control when configuring policies for vADC Administrators.

Monitoring and Maintaining vADCs

The vADC Administrator monitors vADCs in essentially the same manner as a traditional ADC, except for those features and functions which are reserved for the Global Administrator. In addition to the standard data that are displayed in a traditional vADC, many of the information displays also include additional data about each of the vADC instances.

Synchronizing vADCs

Each vADC individually supports configuration synchronization. Unlike the synchronization mechanism used by the Global Administrator, which is responsible for synchronizing elements such as VLANs and throughput limits, this mechanism is controlled by the vADC administrator and synchronizes elements such as filters, SLB groups, virtual IPs, and all the vADC SLB settings.

If the vADC Administrator needs to synchronize vADC configurations, the synchronization is done in the same manner as traditional ADCs using the `/oper/slb/sync` command. For more details, see the *Alteon Application Switch Operating System Command Reference*.

Resource Management

ADC-VX manages vADC resources by limiting the resource consumption of vADCs. You can enable or disable this feature.

- **Enabled** (default for Alteon 5224)—Limits the resources available to vADCs. See Limiting Resource Consumption of vADCs, page 411.
- **Disabled** (default for Alteon 5412)—Enables sharing of any extra available resources between vADCs. See Sharing Idle Resource Consumption with Other vADCs, page 410.

The Global Administrator can switch between these two modes. When changing modes, all vADCs remain active and operational. Any connections beyond the allowed maximum resource consumption are gracefully timed out rather than discarded.

Sharing Idle Resource Consumption with Other vADCs

In share (disabled) mode, although resources are guaranteed to all vADCs, any resources that are used from the idle pool can be allocated permanently to other vADCs. This behavior may decrease the performance of an application as the idle resource is no longer available to the vADC associated with that application.

For example, if an SP or MP uses unassigned CUs, these unassigned CUs are used equally across the vADCs that require the resources.

To share resource consumption of vADCs

Access the *System* menu and disable the **limitcu** command.

```
>> /cfg/sys/limitcu/disable
```

Limiting Resource Consumption of vADCs

Limit (enabled) mode is a resource management mode for handling idle resources. Unlike share mode, in which idle resources can be used by any active vADC, in limit mode idle resources remain unused and vADCs can use only resources assigned specifically to them.

In this mode, resource consumption is static and is allocated to a full system of 24 vADCs.

To limit resource consumption of vADCs

Access the *System* menu and enable the **limitcu** command.

```
>> /cfg/sys/limitcu/enable
```

Resource Dashboard

Each vADC has an accompanying dashboard that monitors the processing power and throughput usage relative to the total allocated resources. The dashboard provides a centralized view of this data so the Global Administrator can preemptively identify potential application and user issues and needs by verifying the health, resource usage, and activity of the vADC.

Note: The dashboard is only accessible through the BBI.

The dashboard displays data on throughput and CPU usage, enabling the Global Administrator to identify allocation issues and dependencies between the throughput and CPU usage. The charts on the dashboard provide the Global Administrator with answers to the following questions:

- Are there any vADCs (applications and/or services) that do not have enough resources to fulfill their tasks successfully or optimally?
- If there are not enough resources, is there a problem with the system or application that needs to be addressed, or is it just displaying uncharacteristic behavior?

This section includes the following topics:

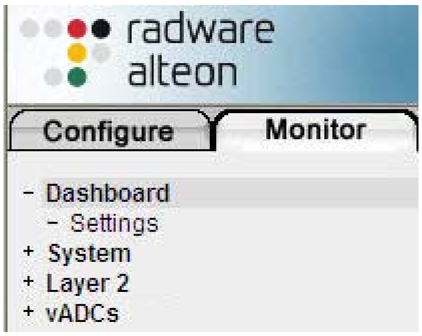
- Accessing the Dashboard, page 412
- Dashboard Charts, page 413
- Settings Menu, page 417

Accessing the Dashboard

The following is the procedure for accessing the resource dashboard.

To access the dashboard

- From the *Monitor* tab, select **Dashboard**.



The following is an example dashboard display of multiple vADCs, as set for viewing through the *Settings* menu (see Settings Menu, page 417)

Figure 65: Example Dashboard Display for Multiple vADCs

Dashboard Charts

The dashboard displays two charts:

- **Resource utilization (CPU and memory)**—This chart consists of two metrics: CPU and memory consumption per vADC.
- **Service usage of the set limit (in percents)**—This chart displays throughput, SSL, and compression consumption per vADC.

You can choose one of two chart types to display for each dashboard:

- Bar chart
- Line chart

Chart Filters

You can select one of the following filters based on operating capacity:

- Customize, default view
- vADCs operating at 90% capacity
- vADCs operating at 80% capacity
- vADCs operating at 70% capacity
- Time (real-time, 1 hour, 24 hours)

Tool Tips

All charts include tool tips which provide more detailed information for a given vADC.

For example, the tool tip for the Throughput Utilization chart may state “Throughput utilization is 500Mb/1Gb”, meaning that the vADC is using on average 500 Mb out of the allocated 1 Gb.

Chart Behavior

Table 36 describes the behavior of each chart according to the selected chart type:

Table 36: Chart Views

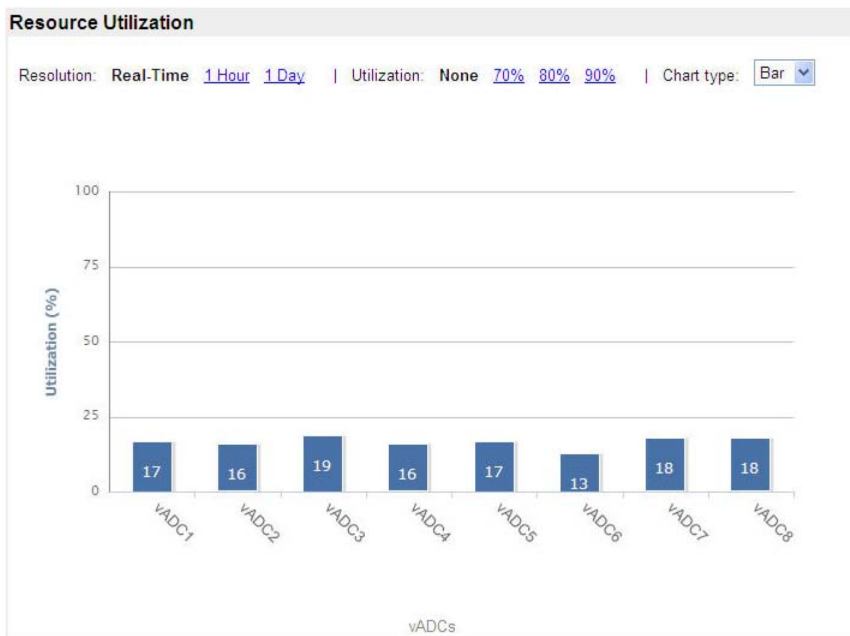
<table>
<tr><th>Chart View</th><th>Chart Type</th><th>Behavior</th></tr>
<tr><td rowspan="2">Resource Utilization Chart</td><td>Bar</td><td>When using filters:
• The real-time filter displays real time data.
• The hour displays the maximum value of the last hour.
• The day filter displays the maximum value within the last 24 hours.</td></tr>
<tr><td colspan="2">The following is a sample resource utilization bar chart:

</td></tr>
</table>

Table 36: Chart Views (cont.)

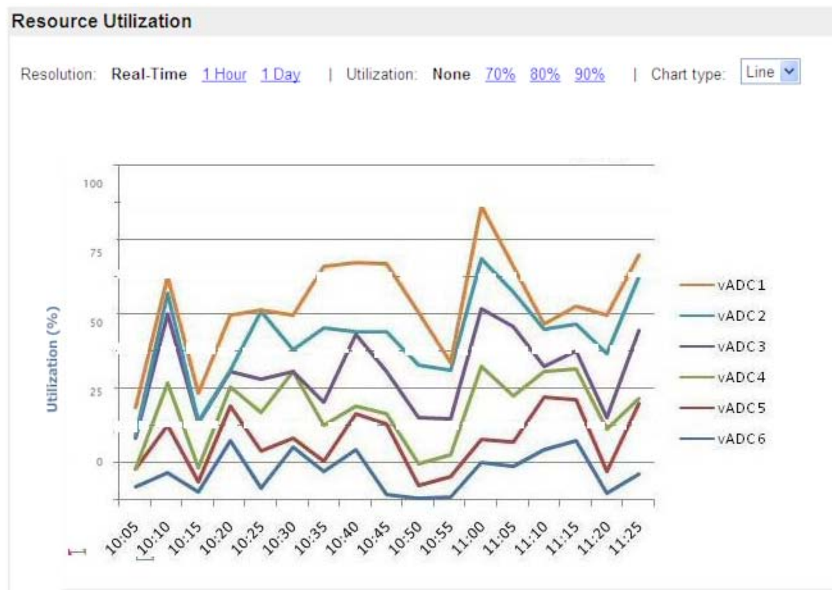
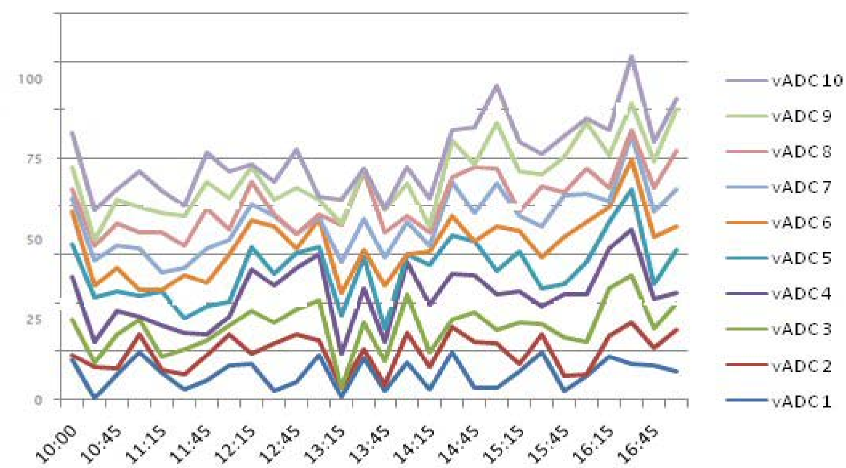
Chart View	Chart Type	Behavior
	Line	This displays the CPU utilization. Multiple lines in different colors are used to represent the different vADCs.
	The following is a sample resource utilization line chart: 	

Table 36: Chart Views (cont.)

Chart View	Chart Type	Behavior
Service Utilization Chart	Bar	When using the tabs: • The *System Throughput* tab displays the amount of throughput that is used in relation to the limit set by the Global Administrator. • The *SSL CPS* tab displays the number of SSL CPSs consumed in relation to the limit set by the Global Administrator. • The *Comp. Throughput* tab displays the amount of data going through the compression engine in relation to the limit set by the Global Administrator. When using filters: • The real-time filter displays real-time data. • The hour displays the maximum value of the last hour. • The day filter displays the maximum value of the last 24 hours. To provide context, the tool tip displays the frequency of the value from the last time period.
	The following is a sample resource throughput bar chart: 	

Table 36: Chart Views (cont.)

Chart View	Chart Type	Behavior
Service Utilization Chart	Line	The tool tip displays detailed data per vADC.
(continued)	The following is a sample resource throughput line chart: 	

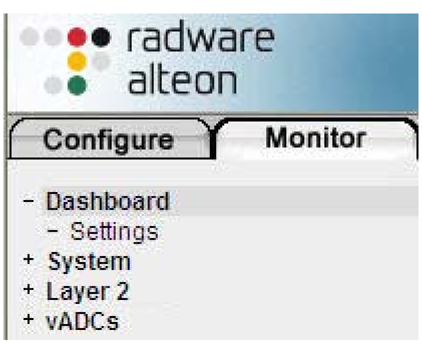
Settings Menu

The *Settings* menu is used to configure the following dashboard settings:

- Sampling interval
- Default chart type
- vADC chart selection

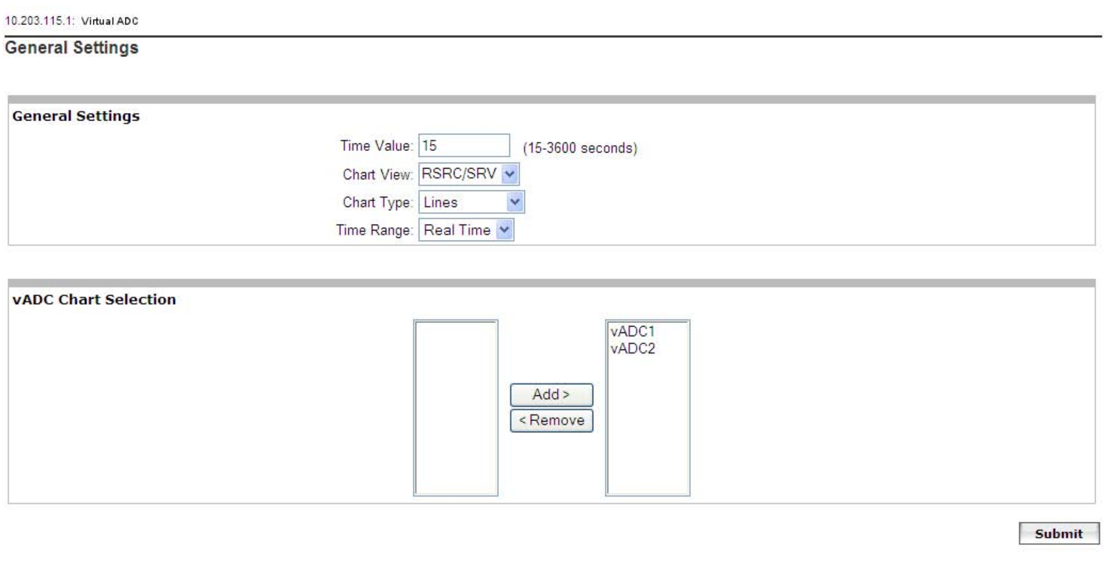
To configure the dashboard settings

1. Expand the *Dashboard* option to display the *Settings* menu.



The following panel displays:

Figure 66: Dashboard Settings Example



2. Change the following settings as required:

Parameter	Description
Time Value	This sets the display increments of the real-time chart. Range: 15—3600 seconds Default: 15 seconds
Chart View	The chart view affects the way information is selected. Values: • Thr/Util—Each chart is independent and selects data for display based on its specific category. For example, the Top 10 Chart displays the top 10 vADCs in the resource utilization category, and the top 10 vADCs in the throughput category. • Throughput—Services is the selection key. The Top 10 chart displays the top 10 vADCs that consume the most throughput relative to their throughput limit. • Util—Resource utilization is the selection key. The Top 10 chart displays the top 10 vADCs that consume the most resources relative to their allocated resource capacity. Default: RSRC/SRV
Chart Type	Options: • Bars—Displays an aggregate view of all the collected data based on the selected time period. • Lines—Displays consumption across time based on all the collected data. Default: Lines For more information on the chart types, see Dashboard Charts, page 413.

Parameter	Description
Time Range	Each chart displays a set of sampled data collected across a period of up to a month. You can change the view based on set time periods in the context of the chart. Options: • Real Time • 1 Hour • 1 Day Default: Real Time
vADC Chart Selection	You can create a customized chart that only displays selected vADCs. Options: • Add vADCs to the chart • Remove vADCs from the chart By default, 10 vADCs are displayed. You can optionally select specific vADCs to display. Default Mode: Display the first 10 vADCs created

3. Click **Submit** to save your changes.

Basic ADC-VX Procedures

This section includes basic procedures for common ADC-VX operations.

- Creating a New vADC, page 419
- Resizing vADC Resources, page 427
- Assigning a VLAN Shared Interface to a vADC, page 428

Creating a New vADC

There are two options for creating vADCs:

- Creating a Basic vADC with the Creation Dialog, page 420
- Creating a vADC Using the vADC Menu, page 423

This section also includes:

- Enabling a Newly Created vADC, page 426

For the purposes of illustration, the example procedures in this section illustrate a vADC created for a new Marketing Portal, which includes the following configuration:

- The new vADC is set with four VLANs.
- Only one VLAN is limited for a specific subnet (in the example, 100), while VLANs 101, 102, and 200 can use any IP subnet as required by the vADC Administrator.

For more details on the vADC Creation Dialog and the *vADC Configuration* menu, see the section on the **/cfg/vadc** menu in the *Alteon Application Switch Operating System Command Reference*.

Creating a Basic vADC with the Creation Dialog

This example creates a basic vADC through the vADC Creation Dialog. The Creation Dialog is invoked whenever you create a new vADC using the **/cfg/vadc** menu:

```
>> Global - Configuration# vadc
Enter vADC Number [1-]: 20

Do you wish to use vADC creation dialog? [y/n]: y
Do you wish to import a configuration file? [y/n] n

Enter vADC name: "Marketing Portal"
Enter throughput limit in Mbps: 1000
Do you want to edit the default acceleration settings? [y/n]: y
Enter SSL CPS limit: 400
Enter Compression limit: 200
Enter Cache RAM allocation:
2 Capacity Unit is Assigned



Enter VLAN Number to be added: 100-102, 200

Do you want to configure Allowed Networks? [y/n]: y

Enter VLAN Number: 100
Enter allowed IP version[v4,v6]: v4
Enter allowed IP network: 192.168.20.0
Enter subnet: 255.255.255.0

Do you want to assign additional IP network to the allowed list [y/n]? n

Enter vADC management IP address(v4 or v6): 10.1.1.1
Enter vADC management subnet mask: 255.255.255.0
Enter vADC management default gateway(v4 or v6): 10.1.1.100

Do you wish to use a different vADC ID for peer? [y/n]: n

Do you wish to use a different vADC name for peer?[y/n]: n


Enter vADC Peer management address(v4 or v6): 10.1.1.2
Enter vADC management subnet mask: 255.0.0.0
Enter vADC Peer management gateway address(v4 or v6): 10.1.1.100

Do you wish to enable vADC ? [y/n]:

>> Global - Configuration# apply
------------------------------------------------------------------
Apply complete; don't forget to 'save' updated configuration.
```

To enable delegated services

After creating a basic vADC with the Creation Dialog, the Global Administrator can configure additional settings using the vADC menu system. Under the **/cfg/vadc/sys** menu, for example, the Global Administrator can enable or disable certain system delegated services in order to set the global usage policy, such as centralized logging and SMTP.

In this example, the Global Administrator may want to set a global usage policy that results in all vADCs being required to use the organization's AAA server. To do so, the Global Administrator can impose and lock certain delegated services so that the vADC Administrator is not able to reconfigure them.

1. In the following steps, the syslog and RADIUS servers are enabled:

```
/cfg/vadc 2/sys

>> vADC 2# sys
------------------------------------------------------------
[vADC system services Menu]
     mmgmt    - Management Port Menu
     peer     - Sync Peer Management Port Menu
     sync     - Assign target appliance for configuration sync
     haid     - Set HA-ID value
     syslog   - System Syslog Servers
     radius   - System RADIUS Servers
     tacacs   - System TACACS Servers
     access   - System Access Menu
     idle     - System timeout for idle CLI sessions
     smtp     - System SMTP host
     cur      - Display current vADC system parameters
>> Global - vADC system services# syslog
------------------------------------------------------------
[Global - vADC 1 sys/syslog Menu]
     delegate - Enable/Disable service delegation from global to vADC
     lock     - Lock access for vADC Administrator
     unlock   - Unlock access for vADC Administrator
     cur      - Display current settings
>> Global - vADC sys/syslog# delegate
Current Settings: disabled
Enter new Settings [d/e]:e
```

```
(continued)
>> Global - vADC sys/syslog# ..
------------------------------------------------------------
[vADC system services Menu]
     mmgmt    - Management Port Menu
     peer     - Sync Peer Management Port Menu
     sync     - Assign target appliance for configuration sync
     haid     - Set HA-ID value
     syslog   - System Syslog Servers
     radius   - System RADIUS Servers
     tacacs   - System TACACS Servers
     access   - System Access Menu
     idle     - System timeout for idle CLI sessions
     smtp     - System SMTP host
     cur      - Display current vADC system parameters

>> Global - vADC system services# radius
------------------------------------------------------------
[vADC sys/RADIUS Menu]
     delegate - Enable/Disable service delegation from global to vADC
     lock     - Lock access for vADC Administrator
     unlock   - Unlock access for vADC Administrator
     cur      - Display current settings
>> Global - vADC sys/RADIUS# delegate
Current Settings: disabled
Enter new Settings [d/e]:e
>> Global - vADC sys/RADIUS# apply
```

2. The following **cur** commands display the status of vADC 1 with syslog and RADIUS servers enabled:

   — Display for the Global Administrator

```
>> Global - System# syslog/cur
Current syslog configuration:
  hst1 212.150.48.1, severity 7, facility 7
  hst2 0.0.0.0, severity 7, facility 0
  hst3 0.0.0.0, severity 7, facility 0
  hst4 0.0.0.0, severity 7, facility 0
  hst5 0.0.0.0, severity 7, facility 0, console enabled
  syslogging all features
>> Global - System# radius/cur
Current RADIUS settings:
 RADIUS authentication currently ON
 Primary RADIUS Server 192.168.1.2
 Secondary RADIUS Server 0.0.0.0
 Primary Radius Server Secret is empty
 Secondary Radius Server Secret is empty
 Current RADIUS Server 192.168.1.2
 RADIUS port 1645, retries 3, timeout 3
 Secure backdoor access disabled
```

— Display for the vADC Administrator

```
>> vADC 1 - Syslog# cur
Current syslog configuration:
  Current  Syslog Status: Enabled
>> vADC 1# sys/radius/cur
Current RADIUS status: Enabled
```

Creating a vADC Using the vADC Menu

The following is an example procedure for creating a vADC using the *vADC* menu.

To create a vADC using the *vADC* menu

1. Create a basic vADC using the **/cfg/vadc** menu.

```
>> Global - Main# /cfg/
------------------------------------------------------------
[Configuration Menu]
     sys       - System-wide Parameter Menu
     port      - Port Menu
     vadc      - vADC Management Menu
     dashboard - Dashboard Menu
     l2        - Layer 2 Menu
     dump      - Dump current configuration to script file
     ptcfg     - Backup current configuration to FTP/TFTP server
     gtcfg     - Restore current configuration from FTP/TFTP server

>> Global - Main# /cfg/vadc 2
```

2. Enter a name for the vADC in order to access it again using the *vADC* menu.

```
/cfg/vadc 2/name

>> vADC 4# name "Marketing Portal"
Current vADC name:
New vADC name:     Marketing Portal

>> vADC 4# apply
------------------------------------------------------------------
Apply complete; don't forget to 'save' updated configuration.
```

3. The initial Management IP is the address assigned to the vADC for initial access. This address can be changed by the vADC Administrator based on the vADC's specific requirements:

```
>> Global - vADC 4 sys/mmgmt# addr 10.203.114.54
Current vADC IP address:     0.0.0.0
New pending vADC 4 IP address: 10.203.114.53

>> Global - vADC 4 sys/mmgmt# mask 255.255.0.0
Current vADC subnet mask:     0.0.0.0
New pending vADC 4 subnet mask: 255.255.0.0

>> Global - vADC 4 sys/mmgmt# gw 10.203.1.1
Current vADC default gateway:     0.0.0.0
New pending vADC 4 default gateway: 10.203.1.1

>> Global - vADC 4 sys/mmgmt# unlock
Current status: locked
New status:     unlocked
```

4. Assign to a vADC the exact application throughput requirement.

Note: When assigning a vADC with the required throughput, no capacity units are assigned. You must do this separately.

```
>> vADC 4# limit 1000
Current Settings:
        vADC 4 throughput assignment: 625 Mbps, ssl assignment: 4200 CPS,
compression assignment: 0 Mbps
New Settings:
        vADC 4 throughput assignment: 1000 Mbps, ssl assignment: 4200 CPS,
compression assignment: 0 Mbps
>> vADC 4# apply
------------------------------------------------------------------
Apply complete; don't forget to 'save' updated configuration.
```

5. When assigning capacity units, you need to consider the total allocated throughput. If the throughput allocated is 1 Gbps, Alteon does not allow you to assign only one capacity unit, but instead requires you to assign at least two capacity units.

```
>> vADC 4# cu 2

Current Settings:
        vADC 4 Assigned Capacity Units:
New Settings:
        vADC 4 Assigned Capacity Units: 2
```

6. Each vADC requires at least one VLAN assigned to it. A vADC supports any type of interface represented by a VLAN ID. Alteon uses VLAN IDs to represent any type of link, and such links can be associated with a vADC (trunk, dedicated link, VLAN tag on a dot1q trunk, team, shared interface, and so on).

   For an example of assigning a VLAN shared interface to a vADC, see Assigning a VLAN Shared Interface to a vADC, page 428.

You can add VLANs using one of the following syntaxes:

— *vlan1 vlan2 vlan3* (one by one)

— *vlan1-vlan3 vlan4* (range and list)

```
>> vADC 4# add 101-102 104

Current vADC 4 Layer2 interfaces:
Pending new vADC 4 Layer2 interfaces:  101 102 104

>> vADC 4# add 103

Current vADC 4 Layer2 interfaces:
Pending new vADC 4 Layer2 interfaces:  101-104
>> Global - vADC allowed IP networks# add
Enter allowed network number: 1
Current VLAN Number: 0
Pending new VLAN Number: 100
Enter new VLAN Number [1-4090]: 100
Enter new IP version[v4, v6]: v4
Current Network IP address: 0.0.0.0
Enter new Network IP address: 192.168.1.0
Current Network Mask: 0.0.0.0
Enter new Network Mask: 255.255.255.0

Current Settings:
    vADC 1 allowed networks:
    No allowed IP networks configured.

New Settings:
    vADC 1 allowed networks:
    Current IPv4 allowed networks:
      Id    Vlan  NetAddress                           NetMask
      ----  ----  ---------------                      ---------------
      1     100   192.168.1.0                          255.255.255.0
```

Enabling a Newly Created vADC

After creating a new vADC either through the Creation Dialog or the vADC menu, you must enable it for it to be functional, as shown in the following example:

To enable a newly created vADC

```
>> Global - Configuration# vadc 4
------------------------------------------------------------
[vADC 1 Menu]
     sys       - Enable system services
     add       - Add Vlan
     rem       - Remove Vlan
     name      - vADC Name
     cu        - Update Capacity Units
     limit     - Maximum throughput allowed
     allow     - Allocate allowed IP networks
     users     - vADC Users Menu
     swf       - Enable/Disable software features
     ena       - Enable vADC
     dis       - Disable vADC
     del       - Delete vADC
     cur       - Display current vADC configuration

>> vADC 1# ena
Current status: disabled
New status:     enabled

>> vADC 1#
```

The following example displays all vADCs:

Resizing vADC Resources

You can resize vADC resources by changing the number of capacity units, as shown in the following example.

To resize vADC resources

(In order to resize resources, you must first disable the vADC)

```
>> vADC 1# dis
Current status: enabled
New status:     disabled

>> vADC 1# apply

------------------------------------------------------------
Apply complete; don't forget to 'save' updated configuration.
```

(Change the number of allocated capacity units)

```
>> vADC 1# cu 5

Current Settings:
        vADC 1 Assigned Capacity Units: 3
New Settings:
        vADC 1 Assigned Capacity Units: 5

>> vADC 1# apply

>> vADC 1# ena
Current status: disabled
New status:     enabled

>> vADC 1# apply

------------------------------------------------------------
Apply complete; don't forget to 'save' updated configuration.

>> vADC 1#
```

Assigning a VLAN Shared Interface to a vADC

When assigning a VLAN that is a shared interface to a vADC, the shared interface must be a dedicated port. For more information on shared interfaces, see Integrating vADCs into a Shared Network Design, page 408.

To assign a VLAN shared interface to a vADC

```
>> vADC 1# /cfg/port
Enter port (1-16):      15
------------------------------------------------------------
[Port 15 Menu]
     gig      - SFP Gig Phy Menu
     pvid     - Set default port VLAN id
     alias    - Set port alias
     name     - Set port name
     rmon     - Enable/Disable RMON for port
     tag      - Enable/disable VLAN tagging for port
     iponly   - Enable/disable allowing only IP related frames
     ena      - Enable port
     dis      - Disable port
     cur      - Display current port configuration

>> Port 15# ena
Current status: enabled
New status:     enabled

>> Global - Configuration# /cfg/l2/vlan 300

VLAN number 300 with name "VLAN 300" created.
------------------------------------------------------------
[VLAN 300 Menu]
     name     - Set VLAN name
     stg      - Assign VLAN to a Spanning Tree Group
     add      - Add port to VLAN
     rem      - Remove port from VLAN
     def      - Define VLAN as list of ports
     learn    - Enable/disable smac learning
     shared   - Enable/disable VLAN sharing between vADCs
     ena      - Enable VLAN
     dis      - Disable VLAN
     del      - Delete VLAN
     cur      - Display current VLAN configuration

>> VLAN 300# add 15
Port 15 is an UNTAGGED port and its current PVID is 1.
Confirm changing PVID from 1 to 300 [y/n]: y
Current ports for VLAN 300:      empty
Pending new ports for VLAN 300:  15
```

```
>> VLAN 300# shared
Current Enabled VLAN sharing: disabled
Enter new Enabled VLAN sharing [d/e]: e

>> VLAN 300# ena
Current status: disabled

>> vADC 1# add 300
Current vADC 1 Layer2 interfaces: 100
Pending new vADC 1 Layer2 interfaces:  300

>> vADC 1# apply
```

The following example displays information for a shared interface:

```
>> Global - Layer 2# vlan
VLAN          Name              VADCs      Status  Learn Shared Ports
----   -------------------- -------------  ------  ----- ------ -----
1      Default VLAN                          ena    ena   dis   1-14 16
3      VLAN 3                                ena     ena   dis   empty
100    VLAN 100                  1           ena     ena   dis   16
300    VLAN 300                  1           ena     ena   ena   15
```

Importing the Active ADC Configuration

The vADC Administrator and the Global Administrator can import configurations from one ADC form factor to another:

- The vADC Administrator import tasks include:
    - — Restoring the Active Configuration of an Existing vADC, page 429
- The Global Administrator import tasks include:
    - — Performing a Complete System Recovery, page 430
    - — Importing vADC Configuration Files to an Existing vADC, page 430
    - — Creating a New vADC from Configuration Files of a Physical ADC, page 432

For both administrators, the file can contain a full ADC configuration or a partial ADC configuration.

Restoring the Active Configuration of an Existing vADC

The vADC Administrator can restore the active configuration of an existing vADC.

To restore the active configuration of an existing vADC

- Access the *Active Switch Configuration Restoration* menu and configure the following parameters:

```
Configuration# gtcfg <hostname> <filename> <-tftp | username password> [-mgmt
| -data] <scp>
```

Performing a Complete System Recovery

The Global Administrator can perform a complete system recovery (administrator configuration and vADC files) and restore all current settings.

To perform a complete system recovery

1. Access the *Active Switch Configuration Restoration* menu.

```
>> /cfg/gtcfg
```

2. When prompted, configure the following parameters:

```
Select import option [all/vadc/padc]: all
Enter hostname or IP address of FTP/TFTP/SCP server:
Enter name of file on FTP/TFTP/SCP server:
Enter username for FTP/SCP server or hit return for TFTP server:
```

Importing vADC Configuration Files to an Existing vADC

The Global Administrator can import vADC configuration files to an existing vADC and define the type of file to import. Import options include the following:

- **all**—Performs a complete system recovery (AC and vADC files) and will restore all current settings.
- **vadc**—Imports vADC configuration files to an existing vADC and define the type of file to recover. Sub-options include:
  - **all**—Creates a new vADC from the settings of the recovery file or replace an existing one.
  - **vadmin**—Creates a vADC Administrator level backup file containing the configuration information available to the vADC administrator. This option requires a vADC to exist in the system.
- **padc**—Creates a new vADC from the configuration files of a physical, standalone ADC or to replace one or all existing vADCs with the configuration files of a physical, standalone ADC.

This section includes the following procedures:

- To create a new vADC from the settings of the recovery file, page 430
- To create a vADC Administrator level backup file, page 431

To create a new vADC from the settings of the recovery file

1. Access the *Active Switch Configuration Restoration* menu.

```
>> /cfg/gtcfg
Select import option [all/vadc/padc]: vadc
Select vADC recovery type [all/vadmin]: vadmin
Enter vADC number: [1-28]: 1
```

If the selected vADC 1 already exists, the following message displays:

```
vADC 1 already exists in the system, do you wish to replace it? [y/n]: y
```

2. Enter **y** to replace the existing vADC.
3. When prompted, configure the following parameters:

```
Enter hostname or IP address of FTP/TFTP/SCP server:
Enter name of file on FTP/TFTP/SCP server:
Enter username for FTP/SCP server or hit return for TFTP server:
```

Example Creating a New vADC from the Settings of the Recovery File

```
>> Global - Configuration# /c/gtcfg
Select Import option [all/vadc/padc]:vadc
Select vADC recovery type [all/vadmin]:all
Enter vADC number: [1-28]: 1
Enter hostname or IP address of FTP/TFTP/SCP server: 192.168.1.1
Enter name of file on FTP/TFTP/SCP server: OCS Service vADC
Enter username for FTP/SCP server or hit return for TFTP server: radware
Enter password for username on FTP/SCP server:
Enter "scp" or hit return for FTP server:
Include private keys? [y/n]: y
Enter passphrase:
Reconfirm passphrase:
Connecting to 192.168.1.1...
```

To create a vADC Administrator level backup file

1. Access the *Active Switch Configuration Restoration* menu.

```
>> /cfg/gtcfg
Select import option [all/vadc/padc]: vadc
Select vADC recovery type [all/vadmin]: vadmin
Enter vADC number: [1-28]: 1
```

If the selected vADC 1 already exists, the following message displays:

```
vADC 1 already exists in the system, do you wish to replace it? [y/n]: y
```

2. Enter **y** to replace the existing vADC.
3. When prompted, configure the following parameters:

```
Enter hostname or IP address of FTP/TFTP/SCP server:
Enter name of file on FTP/TFTP/SCP server:
Enter username for FTP/SCP server or hit return for TFTP server:
```

Creating a New vADC from Configuration Files of a Physical ADC

The Global Administrator can create a new vADC from the configuration files of a physical, standalone ADC, or to replace one or all existing vADCs with the configuration files of a physical, standalone ADC.

To create a new vADC from the configuration files of a physical, standalone ADC

1. Access the *Active Switch Configuration Restoration* menu.

```
>> /cfg/gtcfg
```

2. When prompted, configure the following parameters:

```
Select import option [all/vadc/padc]: padc
Enter hostname or IP address of FTP/TFTP/SCP server:
Enter name of file on FTP/TFTP/SCP server:
Enter username for FTP/SCP server or hit return for TFTP server:
Enter password for username on FTP/SCP server:
Enter "scp" or hit return for FTP server:
Include private keys? [y/n]: y
Enter passphrase:
Reconfirm passphrase:
Enter vADC number: [1-28]: 1
```

If the selected vADC 1 already exists, the following message displays:

```
vADC 1 already exists in the system, do you wish to replace it? [y/n]: y
```

3. Enter **y** to replace the existing vADC.
4. When prompted, configure the following parameters:

```
Enter hostname or IP address of FTP/TFTP/SCP server:
Enter name of file on FTP/TFTP/SCP server:
Enter username for FTP/SCP server or hit return for TFTP server:
Enter vADC number: [1-28]: 1
```

The following message displays:

```
vADC 1 doesn't exist. Do you wish to create vADC 1? [y/n]: y
```

5. Enter **y** to create a new vADC.
6. When prompted, configure the following parameters:

```
Enter vADC name: Employee Portal
Enter throughput limit in Mbps: 1000
Do you want to configure edit the default acceleration settings? [y/n]: n
```

To replace an existing vADC with the configuration files of a physical, standalone ADC

1. Access the *Active Switch Configuration Restoration* menu.

```
>> /cfg/gtcfg
```

2. When prompted, configure the following parameters:

```
Select import option [all/vadc/padc]: padc
Enter hostname or IP address of FTP/TFTP/SCP server:
Enter name of file on FTP/TFTP/SCP server:
Enter username for FTP/SCP server or hit return for TFTP server:
Enter password for username on FTP/SCP server:
Enter "scp" or hit return for FTP server:
Include private keys? [y/n]: y
Enter passphrase:
Reconfirm passphrase:
Enter vADC number: [1-28]: 1
```

If the selected vADC 1 already exists, the following message displays:

```
vADC 1 is active do you wish to replace its current settings? [y/n] y
```

3. Enter **y** to replace the settings of the existing vADC.
4. When prompted, configure the following parameters:

```
Enter hostname or IP address of FTP/TFTP/SCP server:
Enter name of file on FTP/TFTP/SCP server:
Enter username for FTP/SCP server or hit return for TFTP server:
```

Backing Up the Active vADC Configuration

The vADC Administrator can back up the vADC Administrator level configuration of an existing vADC to a specified destination on the file server.

The Global Administrator can back up both the Global and vADC Administrator level configurations of one or all existing vADCs to a destination on the file server.

This section includes the following topics:

- Backing Up the vADC Administrator Level Configuration, page 434
- Backing Up the Complete System, page 434
- Backing Up vADC Configuration Files from an Existing vADC, page 434
- Backing Up the Entire Administrator Environment, page 435

Backing Up the vADC Administrator Level Configuration

The vADC Administrator can upload the vADC Administrator level configuration of an existing vADC.

To upload the vADC Administrator level configuration of an existing vADC

1. Access the *Active Switch Configuration Restoration* menu.

```
>> /cfg/ptcfg
```

2. When prompted, configure the following parameters:

```
Enter hostname or IP address of FTP/TFTP/SCP server:
Enter name of file on FTP/TFTP/SCP server:
Enter username for FTP/SCP server or hit return for TFTP server:
```

Backing Up the Complete System

The Global Administrator can back up the complete system (administrator environment and vADC files).

To backup the complete system

1. Access the *Active Switch Configuration Restoration* menu.

```
>> /cfg/ptcfg/all
```

2. When prompted, configure the following parameters:

```
Enter hostname or IP address of FTP/TFTP/SCP server:
Enter name of file on FTP/TFTP/SCP server:
Enter username for FTP/SCP server or hit return for TFTP server:
```

Backing Up vADC Configuration Files from an Existing vADC

The Global Administrator can back up vADC configuration files from an existing vADC and define the type of file to back up.

To backup all vADC configuration files from an existing vADC

1. Access the *Active Switch Configuration Restoration* menu.

```
>> /cfg/ptcfg/vadc
Select backup option [all/global/vadc]:all
```

   Choosing this option backs up the entire vADC, including both the Global and vADC administration settings, such as CUs, VLANs, IP interfaces, licenses, SLB, acceleration features, and so on.

2. When prompted, configure the following parameters:

```
Enter vADC number: [1-28, all]:
Enter hostname or IP address of FTP/TFTP/SCP server:
Enter name of file on FTP/TFTP/SCP server:
Enter username for FTP/SCP server or hit return for TFTP server:
```

To backup a vADC Administrator level backup file from an existing vADC

This option creates a vADC Administrator level backup file containing the configuration information available to the vADC administrator.

1. Access the *Active Switch Configuration Restoration* menu.

```
>> /cfg/ptcfg/vadc
Select backup option [all/global/vadc]:vadc
```

2. When prompted, configure the following parameters:

```
Enter vADC number: [1-28, all]:
Enter hostname or IP address of FTP/TFTP/SCP server:
Enter name of file on FTP/TFTP/SCP server:
Enter username for FTP/SCP server or hit return for TFTP server:
```

Backing Up the Entire Administrator Environment

The Global Administrator can back up the entire Administrator environment.

To backup the entire Administrator environment

1. Access the *Active Switch Configuration Restoration* menu.

```
>> /cfg/ptcfg/vadc
Select backup option [all/global/vadc]:global
```

2. When prompted, configure the following parameters:

```
Enter hostname or IP address of FTP/TFTP/SCP server:
Enter name of file on FTP/TFTP/SCP server:
Enter username for FTP/SCP server or hit return for TFTP server:
Enter vADC number: [1-28, all]:
```

Image Management

Alteon can support completely separate and unrelated ADC virtual instances ranging from 10 to 28, whose images and configurations are managed by the Global Administrator. ADC management also includes image management, enabling the Global Administrator to manage both standalone and virtual modes. You can upgrade, patch, migrate, and stage new ADC environments without high operational costs. With image management, you can

- Load new images
- Selectively upgrade system components
- Switch quickly and easily between standalone and virtual ADC modes

This section includes the following topics:

- Image Management in a Standalone ADC, page 438
- ADC-VX Image Management, page 442
- Switching Between System Modes, page 450

What Is An Image

An image is a file that contains specific pre-installed and pre-configured applications necessary to implement one or more of the Alteon form factors.

A set of image files are available for download, letting you upgrade only specific elements of the system. The image is pre-loaded to the system, supporting both ADC-VX and standalone ADC deployment without the need to change software images. For downloading procedures, see the Radware *Alteon Installation and Maintenance Guide*.

The following are the available image types:

Table 37: Image Formats

Image Format	File Name	Description
AlteonOS	AlteonOS-<version>-<platform>.img For example: AlteonOS-29.0.0.0-4408.img	This is the default image you can download when installing an Alteon system. It includes ADC-VX and the ADC application. This image lets you upgrade the entire system or just one of its elements. It is installed on the virtual (vADC) and standalone Alteons, and is used for USB recovery and standalone ADC upgrades. This image upgrades the entire system infrastructure and ADC for both the vADC and standalone mode. For more information on default images, see Default Image, page 437.
ADC Application Image	AlteonOS-<version>-<platform>-ADC.img For example: AlteonOS-29.0.0.0-5000-ADC.img	This image is an upgrade image and is used to install or and upgrade a specific vADC version within an active ADC-VX system. In ADC-VX mode, you can boot to standalone mode from any version installed as an ADC application image. **Note:** This image can only be installed when an image is first installed and set as the default image.

Table 37: Image Formats

Image Format	File Name	Description
ADC-VX Infrastructure Update Image	AlteonOS-<version>-<platform>-VX.img For example: AlteonOS-29.0.0.0-5000-VX.img	This image is an upgrade image for the ADC-VX infrastructure. It is only issued when an update is available to the ADC-VX infrastructure. **Note:** This image can only be installed when an image is first installed and set as the default image.
USB Recovery System Image	Recovery-AlteonOS-<version>-<platform>.zip For example: Recovery-AlteonOS-29.0.0.0-4416.zip	This image is a USB recovery image for the system image. It is used for the entire system, not for only one element (standalone mode, vADC mode, or ADC-VX infrastructure).

Default Image

The default image is the ADC image used in the following scenarios:

- When switching from standalone to ADC-VX
- When creating a new vADC in ADC-VX mode

To assign a default image in ADC-VX

1. Access the *Active Switch Configuration Boot* menu.

```
>> ADC-VX - Main# boot
[Boot Options Menu]
     single   - Switch between ADC-VX and Standalone
     vadc     - Restart selected vADC process
     image    - Select software image to use on next boot
     dimage   - Select default image
     conf     - Select config block to use on next boot
     gtimg    - Download new software image via FTP/TFTP/SCP
     reset    - Reset switch
     cur      - Display current boot options
```

2. Enter **dimage** to select the new default image from a list of existing images.

```
>> ADC-VX - Boot Options# dimage
ADC Application Images:
ID        Version               Downloaded           Image status    vADC IDs
--        -------               ----------           ------------    --------
 1         17:41:28 Sun Jan 13, 2013     Incompatible        -
 2        28.1.0.0      12:45:39 Wed Mar 31, 2013      Active          6
 3        28.1.0.2      17:41:28 Sun Jan 13, 2013      Active          7
 4        28.1.0.3      12:45:39 Wed Mar 31, 2013      Active         10-12
 5        28.1.0.4      17:41:28 Sun Jan 13, 2013      Active         15-20
 6        28.1.0.5      12:45:39 Wed Mar 31, 2013      Idle           28
 7        28.1.0.6      17:41:28 Sun Jan 13, 2013      Idle           1-5
 8        28.1.0.7      12:45:39 Wed Mar 31, 2013      Idle
 9        28.3.0.0      17:41:28 Sun Jan 13, 2013       Active        22
10        28.4.0.0      12:45:39 Wed Mar 31, 2013      Idle            -

Select default image (1-10): 8
```

Note: If you delete the default image, the system automatically selects the latest version number and assigns it as the default image.

What Is Multi-Image Management?

Multi-image management is the part of ADC-VX that enables the Global Administrator to

- Separately control vADC and ADC-VX infrastructure images.
- Maintain backward compatibility between the ADC-VX infrastructure and ADC software.
- Upgrade or patch one or more vADCs with a single action.
- Avoid multiple reloads of the same software image.

Image Management in a Standalone ADC

With image management, the Global Administrator role includes managing enhanced image banks. You can load up to 10 ADC images, which are also used for vADC assignments, and up to four ADC-VX infrastructure images. Global administrators can view and manage ADC-VX and standalone deployment images.

Image Bank

The image bank can store up to 10 ADC application images and ADC-VX infrastructure images. When booting the system or loading an image, the image bank displays all available images and their statuses. You can only load one image of each AlteonOS version.

Loading Images

In standalone mode, you can

- Upgrade the entire system with an AlteonOS image
- Upgrade an ADC application image

To load an AlteonOS image

This procedure upgrades both ADC-VX and ADC application images with a single operation, whether the system is in standalone or ADC-VX mode.

1. Access the *Active Switch Configuration Boot* menu.

```
>> Standalone ADC - Main# boot
[Boot Options Menu]
     virtual  - Switch mode from Standalone to ADC-VX
     image    - Select software image to use on next boot
     conf     - Select config block to use on next boot
     gtimg    - Download new software image via FTP/TFTP/SCP
     reset    - Reset switch [WARNING: Restarts Spanning Tree]
     cur      - Display current boot options
```

2. Enter **gtimg** to load the AlteonOS image.

```
>> Standalone ADC - Boot Options#gtimg
Enter image type [all|vx|adc]: all

ADC-VX Infrastructure Images:
ID        Version                Downloaded            Image status
--        -------                ----------            ------------
 1        28.1.0.5      17:41:28 Sun Jan 13, 2013       Idle
 2        28.1.0.0      12:45:39 Wed Mar 31, 2013       Idle
 3        28.1.0.1      17:41:28 Sun Jan 13, 2013       Idle
 4        28.1.0.2      12:45:39 Wed Mar 31, 2013       Idle

Enter Image ID to be replaced (1-4): 2

ADC Application Images:
ID        Version                Downloaded            Image status    vADC IDs
--        -------                ----------            ------------    --------
 1         17:41:28 Sun Jan 13, 2013     Incompatible       -
 2        28.1.0.0      12:45:39 Wed Mar 31, 2013       Active            6
 3        28.1.0.2      17:41:28 Sun Jan 13, 2013       Active            7
 4        28.1.0.3      12:45:39 Wed Mar 31, 2013       Active          10-12
 5        28.1.0.4      17:41:28 Sun Jan 13, 2013       Active          15-20
 6        28.1.0.5      12:45:39 Wed Mar 31, 2013       Idle             28
 7        28.1.0.6      17:41:28 Sun Jan 13, 2013       Idle             1-5
 8        28.1.0.7      12:45:39 Wed Mar 31, 2013       Idle
 9        28.3.0.0      17:41:28 Sun Jan 13, 2013        Active          22
10        28.4.0.0      12:45:39 Wed Mar 31, 2013       Idle              -

Enter Image ID to be replaced (1-10): 2

Enter hostname or IP address of FTP/TFTP/SCP server: 10.210.31.39
Enter name of file on FTP/TFTP/SCP server: AAS-28.1.0.12--IF-AlteonOS
Enter username for FTP/SCP server or hit return for TFTP server:
```

To load an ADC application image

This procedure uploads an ADC application image for the active standalone ADC, or as an image for one or more vADCs in ADC-VX mode.

1. Access the *Active Switch Configuration Boot* menu.

```
>> Standalone ADC - Main# boot
[Boot Options Menu]
     virtual  - Switch mode from Standalone to ADC-VX
     image    - Select software image to use on next boot
     conf     - Select config block to use on next boot
     gtimg    - Download new software image via FTP/TFTP/SCP
     reset    - Reset switch [WARNING: Restarts Spanning Tree]
     cur      - Display current boot options
```

2. Enter **gtimg** to load the ADC application image.

```
>> Standalone ADC - Boot Options#gtimg
Enter image type [all|vx|adc]: adc

ADC Application Images:
ID        Version               Downloaded           Image status    vADC IDs
--        -------               ----------           ------------    --------
 1         17:41:28 Sun Jan 13, 2013     Incompatible       -
 2        28.1.0.0      12:45:39 Wed Mar 31, 2013      Active           6
 3        28.1.0.2      17:41:28 Sun Jan 13, 2013      Active           7
 4        28.1.0.3      12:45:39 Wed Mar 31, 2013      Active         10-12
 5        28.1.0.4      17:41:28 Sun Jan 13, 2013      Active         15-20
 6        28.1.0.5      12:45:39 Wed Mar 31, 2013      Idle            28
 7        28.1.0.6      17:41:28 Sun Jan 13, 2013      Idle           1-5
 8            -                    -                     -              -
 9        28.3.0.0      17:41:28 Sun Jan 13, 2013       Active         22
10        28.4.0.0      12:45:39 Wed Mar 31, 2013      Idle             -

Enter Image ID to be replaced (1-10): 5

Enter hostname or IP address of FTP/TFTP/SCP server: 10.210.31.39
Enter name of file on FTP/TFTP/SCP server: AAS-28.1.0.12--IF-AlteonOS
Enter username for FTP/SCP server or hit return for TFTP server:
```

Managing Images for ADC-VX

You can add ADC-VX images to the image bank while in standalone mode.

In standalone mode, the Global Administrator can prepare the system for the switch to ADC-VX mode by loading the desired ADC-VX infrastructure image. This image is completely independent from the ADC application image.

To add an ADC-VX infrastructure image

This procedure uploads an ADC-VX infrastructure image to the image bank.

1. Access the *Active Switch Configuration Boot* menu.

```
>> Standalone ADC - Main# boot
[Boot Options Menu]
     virtual  - Switch mode from Standalone to ADC-VX
     image    - Select software image to use on next boot
     conf     - Select config block to use on next boot
     gtimg    - Download new software image via FTP/TFTP/SCP
     reset    - Reset switch [WARNING: Restarts Spanning Tree]
     cur      - Display current boot options
```

2. Enter **gtimg** to load the ADC-VX infrastructure image.

```
>> Standalone ADC - Boot Options#gtimg
Enter image type [all|vx|adc]: vx

ADC-VX Infrastructure Images:
ID        Version                Downloaded           Image status
--        -------                ----------           ------------
 1        28.1.0.5      17:41:28 Sun Jan 13, 2013       Idle
 2        28.1.0.0      12:45:39 Wed Mar 31, 2013       Idle
 3        28.1.0.1      17:41:28 Sun Jan 13, 2013       Idle
 4        28.1.0.2      12:45:39 Wed Mar 31, 2013       Idle

Enter Image ID to be replaced (1-4): 2

Enter hostname or IP address of FTP/TFTP/SCP server: 10.210.31.39
Enter name of file on FTP/TFTP/SCP server: AAS-28.1.0.12--IF-AlteonOS
Enter username for FTP/SCP server or hit return for TFTP server:
```

Image Statuses

The image status displays the current ADC-VX setup. The following are the image statuses:

Caution: You should not remove images that are currently being used by vADCs.

Table 38: Image Statuses

Status Option	Description
Incompatible	The image is only compatible with standalone mode and not in use.
Active	The currently active image in the system.
Assigned	The image is assigned to a vADC that is not active.
Idle	The image is idle and not assigned to a vADC or any other system component.

Note: ADC-VX is not compatible with image versions earlier than version 28.1. Therefore, images that are inherited from a standalone ADC from an earlier version are displayed in the image bank as **incompatible**.

ADC-VX Image Management

Images used in ADC-VX mode are completely independent of other ADC images, enabling you to easily upgrade or patch specific vADCs without affecting certified image versions or existing configurations.

Loading Images

Only the Global Administrator can load images. Because the system only holds one image for each ADC-VX at a time, you do not need to load the same image more than once. The same image can be used by multiple vADCs.

You can only replace an active image after the Global Administrator authorizes the switch.

In the ADC-VX mode, you can load the following images:

- AlteonOS
- ADC application image
- ADC-VX infrastructure image

For more information, see Table 37 - Image Formats, page 436.

To load an AlteonOS image

1. Access the *Active Switch Configuration Boot* menu.

```
>> Global - Main# /boot
------------------------------------------------------------
[Boot Options Menu]
     single   - Switch between ADC-VX  and Standalone
     vadc     - Restart selected vADC process
     dimage   - Select default image
     image    - Select software image to use on next boot
     conf     - Select config block to use on next boot
     gtimg    - Download new software image via FTP/TFTP/SCP
     reset    - Reset switch
     cur      - Display current boot options
     logen    - Enable/Disable Enhanced Log Size
```

2. Enter **gtimg** to load the AlteonOS image.

```
>> Global - Boot Options#gtimg
Enter image type [all|vx|adc]: adc
Enter image ID to be replaced: (1-10)
```

To load an ADC Application image to a vacant image bank

1. Access the *Active Switch Configuration Boot* menu.

```
>> Global - Main# /boot
------------------------------------------------------------
[Boot Options Menu]
     single   - Switch between ADC-VX  and Standalone
     vadc     - Restart selected vADC process
     dimage   - Select default image
     image    - Select software image to use on next boot
     conf     - Select config block to use on next boot
     gtimg    - Download new software image via FTP/TFTP/SCP
     reset    - Reset switch
     cur      - Display current boot options
     logen    - Enable/Disable Enhanced Log Size
```

2. Enter **gtimg** to load the ADC application image.

```
>> Global - Boot Options#gtimg
Enter image type [all|vx|adc]: adc

ADC Application Images:
ID        Version                 Downloaded            Image status    vADC IDs
--        -------                 ----------            ------------    --------
 1         17:41:28 Sun Jan 13, 2013     Incompatible        -
 2        28.1.0.0      12:45:39 Wed Mar 31, 2013       Active             6
 3        28.1.0.2      17:41:28 Sun Jan 13, 2013       Active             7
 4        28.1.0.3      12:45:39 Wed Mar 31, 2013       Active           10-12
 5        28.1.0.4      17:41:28 Sun Jan 13, 2013       Active           15-20
 6        28.1.0.5      12:45:39 Wed Mar 31, 2013       Idle             28
 7        28.1.0.6      17:41:28 Sun Jan 13, 2013       Idle             1-5
 8           -                   -                        -               -
 9        28.3.0.0      17:41:28 Sun Jan 13, 2013        Active          22
10        28.4.0.0      12:45:39 Wed Mar 31, 2013       Idle              -

Enter image ID to be replaced: (1-10) 8

Enter hostname or IP address of FTP/TFTP/SCP server: 10.210.31.39
Enter name of file on FTP/TFTP/SCP server: AAS-28.1.0.12--IF-AlteonOS
Enter username for FTP/SCP server or hit return for TFTP server:
```

Loading Infrastructure Images

The following describes how to load ADC-VX infrastructure images.

To add ADC-VX infrastructure settings

1. Access the *Active Switch Configuration Boot* menu.

```
>> Global - Main# /boot
------------------------------------------------------------
[Boot Options Menu]
     single   - Switch between ADC-VX  and Standalone
     vadc     - Restart selected vADC process
     dimage   - Select default image
     image    - Select software image to use on next boot
     conf     - Select config block to use on next boot
     gtimg    - Download new software image via FTP/TFTP/SCP
     reset    - Reset switch
     cur      - Display current boot options
     logen    - Enable/Disable Enhanced Log Size
```

2. Enter **gtimg**, and enter **vx** to add the ADC-VX infrastructure settings.

```
>> Global - Boot Options# gtimg
Enter image type [all|vx|adc]: vx

ADC-VX Infrastructure Images:
ID        Version                Downloaded              Image status
--        -------                ----------              ------------
 1        28.1.0.3      17:41:28 Sun Jan 13, 2013         Idle
 2        28.1.0.0      12:45:39 Wed Mar 31, 2013         Active
 3        28.1.0.1      17:41:28 Sun Jan 13, 2013         Idle
 4        28.1.0.2      12:45:39 Wed Mar 31, 2013         Idle
```

3. At the prompt, select the image ID for the new infrastructure image.

```
Enter image ID: (1-4) 1
Enter hostname or IP address of FTP/TFTP/SCP server: 10.210.31.39
Enter name of file on FTP/TFTP/SCP server: AAS-28.1.0.12--IF-AlteonOS
Enter username for FTP/SCP server or hit return for TFTP server:
```

Loading vADC Images

ADC application images are used by vADCs and standalone ADCs. Assigning an application image does not interfere with neighboring vADCs or vADCs currently running with the same image version. Application images are reusable and can be assigned in bulk, one by one, or for the entire system.

Upgrading a Single vADC

vADCs can use any of the 10 ADC application images loaded on the system.

To upgrade a single vADC

1. Access the *Active Switch Configuration Boot* menu.
2. Enter **image**, and select the image type used for the upgrade.

```
>> Global - Boot Options# image
Enter image type [vx|adc]: adc

ADC Application Images:
ID        Version               Downloaded             Image status    vADC IDs
--        -------               ----------             ------------    --------
 1         17:41:28 Sun Jan 13, 2013     Incompatible        -
 2        28.1.0.0      12:45:39 Wed Mar 31, 2013       Active            6
 3        28.1.0.2      17:41:28 Sun Jan 13, 2013       Active            7
 4        28.1.0.3      12:45:39 Wed Mar 31, 2013       Active          10-12
 5        28.1.0.4      17:41:28 Sun Jan 13, 2013       Active          15-20
 6        28.1.0.5      12:45:39 Wed Mar 31, 2013       Idle             28
 7        28.1.0.6      17:41:28 Sun Jan 13, 2013       Idle             1-5
 8           -                    -                       -               -
 9        28.3.0.0      17:41:28 Sun Jan 13, 2013        Active          22
10        28.4.0.0      12:45:39 Wed Mar 31, 2013       Idle              -

Enter vADC ID: (1-28) 1
Enter image ID: (1-10) 10
Image 10 instead of image 7 will be used by vADC # next vADC restart
```

3. Restart the vADC process.

```
>> Global - Boot Options# /boot/vadc 1
WARNING: There are unapplied/unsaved configuration changes.
Confirm Operation without apply/save changes [y/n]: y
vADC 1 set to restart. Are you sure? [y/n]: y
```

Upgrading a Group of vADCs

You can upgrade a group of vADCs by entering their ID numbers separated by a comma, or entering a range of vADCs. For example, enter **1-10, 25** to upgrade vADCs 1 to 10 and vADC 25. After upgrading, restart all relevant vADCs for the changes to apply.

To upgrade a group of vADCs

1. Access the *Active Switch Configuration Boot* menu.
2. Enter **image**, and select the image type used for the upgrade.

```
>> Global - Boot Options# image
Enter image type [vx|adc]: adc

ADC Application Images:
ID         Version                 Downloaded             Image status    vADC IDs
--         -------                 ----------             ------------    --------
 1          17:41:28 Sun Jan 13, 2013     Incompatible          -
 2         28.1.0.0      12:45:39 Wed Mar 31, 2013       Active            6
 3         28.1.0.2      17:41:28 Sun Jan 13, 2013       Active            7
 4         28.1.0.3      12:45:39 Wed Mar 31, 2013       Active          10-12
 5         28.1.0.4      17:41:28 Sun Jan 13, 2013       Active          15-20
 6         28.1.0.5      12:45:39 Wed Mar 31, 2013       Idle             28
 7         28.1.0.6      17:41:28 Sun Jan 13, 2013       Idle             1-5
 8            -                   -                        -               -
 9         28.3.0.0      17:41:28 Sun Jan 13, 2013        Active          22
10         28.4.0.0      12:45:39 Wed Mar 31, 2013       Idle              -

Enter vADC ID: (1-28) 1,4 10-15
Enter image ID: (1-10) 10
Image 10 instead of image 7 will be used by vADC 1,4,10-15 next vADC restart
```

3. Restart the vADC processes.

```
>> Global - Boot Options# /boot/vadc
Enter vADC Number [1-28]: 1,4 10-15
WARNING: There are unapplied/unsaved configuration changes.
Confirm Operation without apply/save changes [y/n]: y
vADCs 1-5, 28 set to restart. Are you sure? [y/n]: y
```

Upgrading All vADCs

You can upgrade all vADCs by entering the entire range of existing vADCs. For example, enter **1-28**. After upgrading, restart all vADCs for the changes to apply.

To upgrade all vADCs

1. Access the *Active Switch Configuration Boot* menu.
2. Enter **image**, and select the image type used for the upgrade.

```
>> Global - Boot Options# image
Enter image type [vx|adc]: adc

ADC Application Images:
ID        Version                Downloaded            Image status    vADC IDs
--        -------                ----------            ------------    --------
 1         17:41:28 Sun Jan 13, 2013     Incompatible        -
 2        28.1.0.0      12:45:39 Wed Mar 31, 2013       Active             6
 3        28.1.0.2      17:41:28 Sun Jan 13, 2013       Active             7
 4        28.1.0.3      12:45:39 Wed Mar 31, 2013       Active           10-12
 5        28.1.0.4      17:41:28 Sun Jan 13, 2013       Active           15-20
 6        28.1.0.5      12:45:39 Wed Mar 31, 2013       Idle              28
 7        28.1.0.6      17:41:28 Sun Jan 13, 2013       Idle              1-5
 8           -                  -                        -                -
 9        28.3.0.0      17:41:28 Sun Jan 13, 2013        Active           22
10        28.4.0.0      12:45:39 Wed Mar 31, 2013       Idle               -

Enter vADC ID: (1-28) 1-28
Enter image ID: (1-10) 10
Image 10 instead of image 7 will be used by vADC 1-28 next vADC restart
```

3. Restart the vADC processes.

```
>> Global - Boot Options# /boot/vadc
Enter vADC Number [1-28]: 1-28
WARNING: There are unapplied/unsaved configuration changes.
Confirm Operation without apply/save changes [y/n]: y
vADCs 1-28 set to restart. Are you sure? [y/n]: y
```

Upgrading the ADC-VX Infrastructure

The ADC-VX infrastructure is backward- and forward-compatible with AlteonOS. Because of this, when upgrading the ADC-VX infrastructure software, you are not required to re-certify the AlteonOS for multiple applications.

To upgrade the ADC-VX infrastructure

1. Access the *Active Switch Configuration Boot* menu.
2. Enter **image**, and select the image type used for the upgrade.

```
>> Global - Boot Options# image
Enter image type [vx|adc]: vx

ADC-VX Infrastructure Images:
ID        Version                Downloaded            Image status
--        -------                ----------            ------------
 1        28.1.0.3      17:41:28 Sun Jan 13, 2013       Idle
 2        28.1.0.0      12:45:39 Wed Mar 31, 2013       Active
 3        28.1.0.1      17:41:28 Sun Jan 13, 2013       Idle
 4        28.1.0.2      12:45:39 Wed Mar 31, 2013       Idle

Enter image ID: (1-4) 3

ADC-VX infrastructure image 3 will become active after a system restart

Do you wish to restart the system? [y|n]n
```

Note: If you select **no**, you must restart the system manually.

ADC Application Image Status Options

The image status options display the current ADC-VX setup.

Caution: You should not remove images that are currently being used by vADCs.

Table 39: Image Status Options

Status Option	Description
Incompatible	Image is only compatible with standalone mode and not in use.
Active	The currently active image in the system
Assigned	Image is assigned to a vADC that is not active
Idle	Image is idle and not assigned to a vADC or any other system component

Note: Images inherited from a standalone ADC that are not compatible with ADC-VX display in the ADC application repository as incompatible.

Switching Between System Modes

The factory-installed Alteon image supports both ADC-VX and standalone modes. You can switch between these two modes using a single command.

There are two options for switching between modes:

- **Standalone to ADC-VX**—The administrator selects an ADC-VX infrastructure image from which to boot.
- **ADC-VX to Standalone**—The administrator selects an ADC application image.

Regardless of the mode which is booted, the system does not delete old configuration files.

Caution: If you remove all infrastructure images, the image switching process cannot be initiated.

Switching from Standalone to ADC-VX Mode

Switching from standalone to ADC-VX mode includes both the software and the configuration files. The following boot options are available::

- Boot with factory defaults
- Boot with the last known configuration

When booting with the last known configuration, the image IDs stored in the configuration file are used. If the image bank is empty, the assigned default image is used. The last known ADC-VX configuration includes both AC settings and vADCs.

To switch from standalone to ADC-VX mode

1. Access the *Active Switch Configuration Boot* menu.

```
>> Standalone ADC - Main# boot
[Boot Options Menu]
     virtual  - Switch mode from Standalone to ADC-VX
     dimage   - Select default image
     image    - Select software image to use on next boot
     conf     - Select config block to use on next boot
     gtimg    - Download new software image via FTP/TFTP/SCP
     reset    - Reset switch [WARNING: Restarts Spanning Tree]
     cur      - Display current boot options
```

2. Enter **virtual**, and select **2**.

```
>> Standalone ADC - Boot Options# virtual
Boot options:
1.Factory defaults
2.Last known ADC-VX configuration
Select ADC-VX boot option (1-2):2
Boot with current 28.1.0.0 ADC-VX infrastructure image? [y|n] y
```

The system now boots up with the following settings:

— The ADC-VX infrastructure boots with the pre-installed version (for example version 28.1.0.0) and the vADCs are loaded based on the image IDs originally set for them.

— The standalone configuration file is still available to the system but is not visible to the system administrator.

This procedure prevents combining the configuration import and operational mode transformation.

Switching from ADC-VX to Standalone Mode

When you switch from ADC-VX to standalone mode, ADC-VX images and ADC-VX configuration files are not deleted from their respective banks as a result of the switch.

This option imports the vADC Administrator level settings and the related network settings available to the Global Administrator (VLANs and port association).

Note: Always use the settings available to the vADC, including the management address, management access mode, syslog service, and so on.

To switch a vADC to a standalone ADC

1. Access the *Active Switch Configuration Boot* menu.

```
>> Global - Main# /boot
------------------------------------------------------------
[Boot Options Menu]
     single   - Switch between ADC-VX  and Standalone
     vadc     - Restart selected vADC process
     dimage   - Select default image
     image    - Select software image to use on next boot
     conf     - Select config block to use on next boot
     gtimg    - Download new software image via FTP/TFTP/SCP
     reset    - Reset switch
     cur      - Display current boot options
     logen    - Enable/Disable Enhanced Log Size
```

2. Enter **single** to switch to standalone mode.

```
>> Global - Boot Options# single
Confirm Use last known standalone ADC configuration? [y/n]: y

ADC Application Images:
ID         Version                 Downloaded            Image status
--         -------                 ----------            ------------
 1          17:41:28 Sun Jan 13, 2013     Incompatible
 2         28.1.0.0      12:45:39 Wed Mar 31, 2013       Active
 3         28.1.0.2      17:41:28 Sun Jan 13, 2013      Assigned
 4         28.1.0.3      12:45:39 Wed Mar 31, 2013      Assigned
 5         28.1.0.4      17:41:28 Sun Jan 13, 2013       Idle
 6         28.1.0.5      12:45:39 Wed Mar 31, 2013       Idle
 7         28.1.0.6      17:41:28 Sun Jan 13, 2013       Idle
 8         28.1.0.7      12:45:39 Wed Mar 31, 2013       Idle
 9         28.3.0.0      17:41:28 Sun Jan 13, 2013      Assigned
10         28.4.0.0      12:45:39 Wed Mar 31, 2013       Idle

Select standalone ADC image (1-10) : 7
```

HA ID Management

ADC-VX is a virtual environment in which vADCs can be isolated, share physical links, connect to shared areas of the network, and connect with other ADC form factors. This virtual environment handles all network layers, transitions between standalone to virtual environments and application resiliency.

Starting with version 28.1, ADC-VX supports

- Establishing a high availability relationship between vADCs with different IDs
- Establishing a high availability relationship between vADCs and standalone or virtual appliances
- Sharing a single link between up to 64 vADCs

What is an HA ID?

An HA ID is a unique identifier that you use to assign vADC MAC addresses. You use HA IDs for vADCs with different IDs, establishing relationships, and for when an overlapping MAC address is generated over a shared link.

An HA ID is used to generate a unique MAC similar to the way a vADC ID is used to generate virtual router MACs. Once an HA ID is assigned, a unique virtual router MAC is created for each vADC on the shared interface. vADCs automatically adjust their virtual router MAC allocation based on the HA ID.

HA ID Settings

The HA ID is set by the Global Administrator and is transparent to the vADC administrator. HA IDs are automatically assigned to vADCs during creation. By default, they are identical to the vADC ID and can be modified by the Global Administrator.

Table 40 describes the HA ID settings.

Table 40: HA ID Settings

HA ID	Description
0	This HA ID is required when creating an HA pair between a vADC and any other form factor through a shared interface.
1–63	This range of IDs is used to create a unique virtual router MAC together with the virtual router ID.

Modifying HA IDs

The Global Administrator can modify the HA ID of vADCs.

To modify an HA ID

1. Access the *Active Switch Configuration vADC System Services* menu.

```
>> Global - Main# /cfg/vadc 3/sys
------------------------------------------------------------
[Global - vADC 3 system services Menu]
     mmgmt    - Management Port Menu
     peer     - Sync Peer Management Port Menu
     sync     - Assign target appliance for configuration sync
     haid     - Set HA-ID value
     syslog   - System Syslog Servers
     radius   - System RADIUS Servers
     tacacs   - System TACACS Servers
     access   - System Access Menu
     idle     - System timeout for idle CLI sessions
     smtp     - System SMTP host
     cur      - Display current vADC system parameters
```

2. Enter **haid** to set the HA ID value.

```
>> Global - vADC 3 system services# haid
Enter HA-ID value [0-63]: 1
Current HA-ID value: 3
New HA-ID value: 1
```

Chapter 17 – Application Redirection

Application redirection improves network bandwidth and provides unique network solutions. Filters can be created to redirect traffic to cache or application servers, improving the speed of repeated client access to common Web or application content and freeing valuable network bandwidth.

The following topics are discussed in this chapter:

Note: To access application redirection functionality, the optional Layer 4 software must be enabled. For more information, see the section on Filtering and Layer 4 in the *Alteon Application Switch Operating System Command Reference*.

Overview

Most of the information downloaded from the Internet is not unique, as clients will often access a Web page many times for additional information or to explore other links. Duplicate information also gets requested as the components that make up Internet data at a particular Web site (pictures, buttons, frames, text, and so on) are reloaded from page to page. When you consider this scenario in the context of many clients, it becomes apparent that redundant requests can consume a considerable amount of your available bandwidth to the Internet.

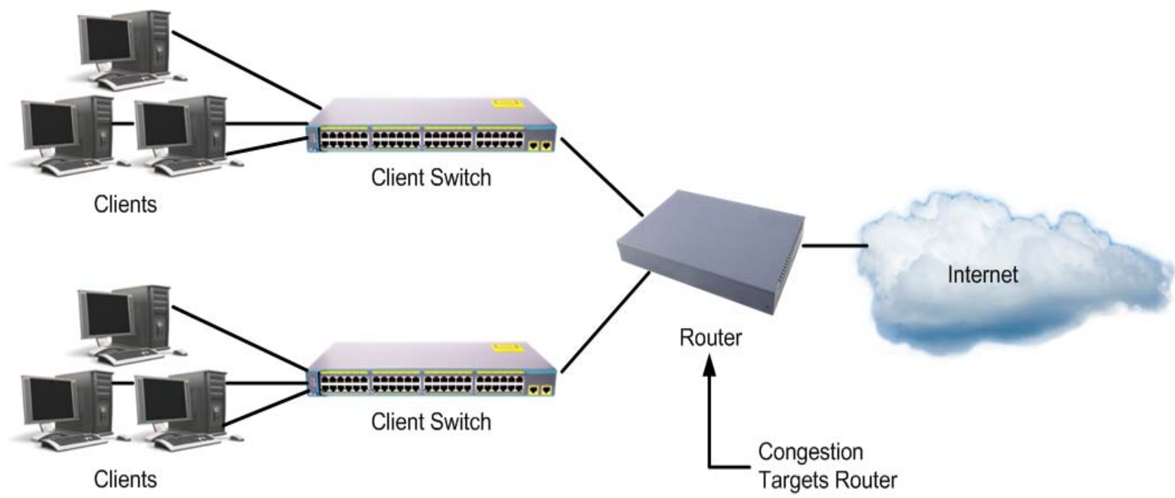
Application redirection can help reduce the traffic congestion during peak loads. When application redirection filters are properly configured, outbound client requests for Internet data are intercepted and redirected to a group of application or cache servers on your network. The servers duplicate and store inbound Internet data that has been requested by your clients. If the servers recognize a client's outbound request as one that can be filled with cached information, the servers supply the information rather than send the request across the Internet.

In addition to increasing the efficiency of your network, accessing locally cached information can be much faster than requesting the same information across the Internet.

Cache Redirection Environment

Consider the network illustrated in Figure 67 - Network without Application Redirection, page 456, where client HTTP requests begin to regularly overload the Internet router.

Figure 67: Network without Application Redirection

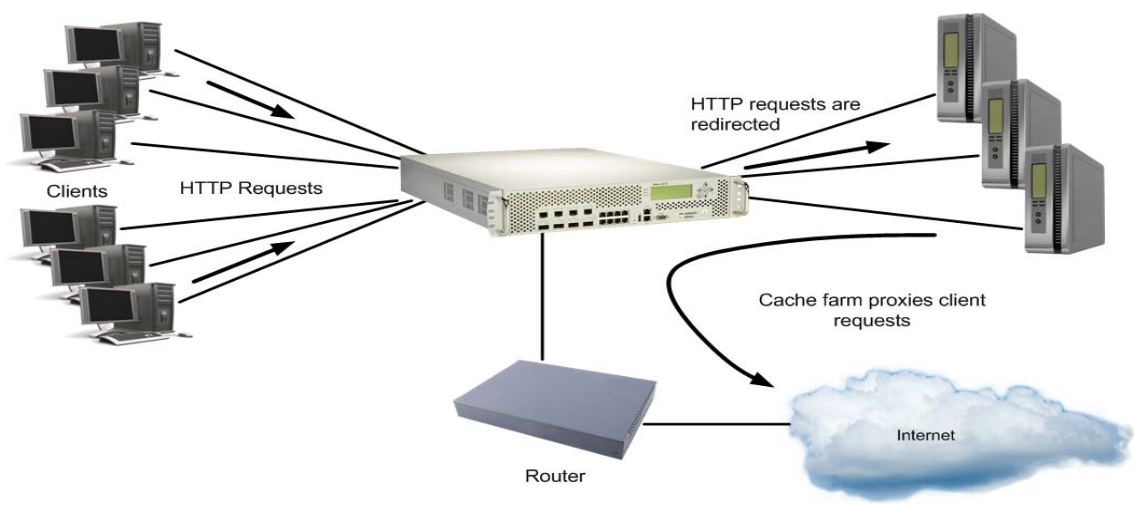


This network needs a solution that addresses the following key concerns:

- The solution must be readily scalable.
- The administrator should not need to reconfigure all the clients' browsers to use proxy servers.

If you have more clients than ports, then connect the clients to a Layer 2 switch, as shown in Figure 68 - Network with Application Redirection, page 456:

Figure 68: Network with Application Redirection



Adding Alteon with optional Layer 4 software addresses the following issues:

- Cache servers can be added or removed dynamically without interrupting services.
- Performance is improved by balancing the cached request load across multiple servers. More servers can be added at any time to increase processing power.
- The proxy is transparent to the client.
- Frames that are not associated with HTTP requests are normally passed to the router.

Additional Application Redirection Options

Application redirection can be used in combination with other Layer 4 options, such as load-balancing metrics, health checks, real server group backups, and more. For more details, see Implementing Server Load Balancing, page 167.

Cache Redirection Example

The following is an example cache redirection configuration.

Example Cache Redirection Configuration

The following is required prior to configuration:

- You must connect to the CLI as the administrator.
- Layer 4 (SLB) software must be enabled.

Note: For details about these procedures or any of the menu commands described in this example, see the *Alteon Application Switch Operating System Command Reference*.

In this example, Alteon is placed between the clients and the border gateway to the Internet. Alteon is configured to intercept all Internet bound HTTP requests (on default TCP port 80), and redirect them to the cache servers. Alteon distributes HTTP requests equally to the cache servers based on the destination IP address of the requests. If the cache servers do not have the requested information, then the cache servers behave like the client and forward the request out to the Internet.

Note: Filters are not limited to the few protocols and TCP or UDP applications shown in this example. See Well-Known Application Ports, page 175 for a list of well-known applications ports.

1. Assign an IP address to each of the cache servers.

   Similar to SLB, the cache real servers are assigned an IP address and placed into a real server group. The real servers must be in the same VLAN and must have an IP route to Alteon that will perform the cache redirection. In addition, the path from Alteon to the real servers must not contain a router. The router would stop HTTP requests from reaching the cache servers and instead direct them back out to the Internet.

   More complex network topologies can be used if configuring IP proxy addresses (see IP Proxy Addresses for NAT, page 464).

For this example, the three cache real servers have the following IP addresses on the same IP subnet:

Table 41: Cache Redirection Example—Real Server IP Addresses

Cache Server	IP address
Server A	200.200.200.2
Server B	200.200.200.3
Server C	200.200.200.4

2. Install transparent cache software on all three cache servers.
3. Define an IP interface on Alteon. Alteon must have an IP interface on the same subnet as the three cache servers because, by default, Alteon only remaps destination MAC addresses.

   To configure an IP interface for this example, enter these commands:

`>> # /cfg/l3/if 1`	(Select IP interface 1)
`>> IP Interface 1# addr 200.200.200.100`	(Assign IP address for the interface)
`>> IP Interface 1# ena`	(Enable IP interface 1)

Note: The IP interface and the real servers must be in the same subnet. This example assumes that all ports and IP interfaces use default VLAN 1, requiring no special VLAN configuration for the ports or IP interface.

4. Define each real server. For each cache real server, you must assign a real server number, specify its actual IP address, and enable the real server. For example:

`>> # /cfg/slb/real 1`	(Server A is Real Server 1)
`>> Real server 1# rip 200.200.200.2`	(Assign Server A IP address)
`>> Real server 1# ena`	(Enable Real Server 1)
`>> Real server 1# /cfg/slb/real 2`	(Server B is Real Server 2)
`>> Real server 2# rip 200.200.200.3`	(Assign Server B IP address)
`>> Real server 2# ena`	(Enable Real Server 2)
`>> Real server 2# /cfg/slb/real 3`	(Server C is Real Server 3)
`>> Real server 3# rip 200.200.200.4`	(Assign Server C IP address)
`>> Real server 3# ena`	(Enable Real Server 3)

5. Define a real server group. This places the three cache real servers into one service group.

`>> Real server 3# /cfg/slb/group 1`	(Select Real Server Group 1)
`>> Real server group 1# add 1`	(Add Real Server 1 to Group 1)
`>> Real server group 1# add 2`	(Add Real Server 2 to Group 1)
`>> Real server group 1# add 3`	(Add Real Server 3 to Group 1)

6. Set the real server group metric to **minmisses**. This setting helps minimize cache misses in the event real servers fail or are taken out of service.

```
>> Real server group 1# metric minmisses
```

7. Verify that server processing is disabled on the ports supporting application redirection.

   Note: Do not use the **server** setting on a port with application redirection enabled. Server processing is used only with SLB. To disable server processing on the port, use the commands on the `/cfg/slb/port` menu, as described in the *Alteon Application Switch Operating System Command Reference*.

8. Create a filter that will intercept and redirect all client HTTP requests.

   The filter must intercept all TCP traffic for the HTTP destination port and must redirect it to the proper port on the real server group.

`>> SLB port 6# /cfg/slb/filt 2`	(Select the menu for Filter 2)
`>> Filter 2# sip any`	(From any source IP addresses)
`>> Filter 2# dip any`	(To any destination IP addresses)
`>> Filter 2# proto tcp`	(For TCP protocol traffic)
`>> Filter 2# sport any`	(From any source port)
`>> Filter 2# dport http`	(To an HTTP destination port)
`>> Filter 2# action redir`	(Set the action for redirection)
`>> Filter 2# rport http`	(Set the redirection port)
`>> Filter 2# group 1`	(Select Real Server Group 1)
`>> Filter 2# ena`	(Enable the filter)

The rport (redirection) parameter must be configured whenever TCP/UDP protocol traffic is redirected. The rport parameter defines the real server TCP or UDP port to which redirected traffic is sent. The port defined by the rport parameter is used when performing Layer 4 health checks of TCP services.

Also, if NAT and proxy addresses are used on Alteon (see step 3), the rport parameter must be configured for all application redirection filters. Make sure to use the proper port designation with rport. If the transparent proxy operation resides on the host, the well-known port 80 (or HTTP) is probably required. If the transparent proxy occurs in Alteon, make sure to use the service port required by the specific software package.

For more information on IP proxy addresses, see IP Proxy Addresses for NAT, page 464.

9. Create a default filter. In this case, the default filter will allow all non-cached traffic to proceed normally.

`>> Filter 2# /cfg/slb/filt 2048`	(Select the default filter)
`>> Filter 2048# sip any`	(From any source IP addresses)
`>> Filter 2048# dip any`	(To any destination IP addresses)
`>> Filter 2048# proto any`	(For any protocols)
`>> Filter 2048# action allow`	(Set the action to allow traffic)
`>> Filter 2048# ena`	(Enable the default filter)

Note: When the proto parameter is not TCP or UDP, sport and dport are ignored.

10. Assign the filters to the client ports. Assuming that the redirected clients are connected to physical ports 5 and 6, both ports are configured to use the previously created filters.

`>> Filter 2048# /cfg/slb/port 5`	(Select the Client Port 5)
`>> SLB Port 5# add 2`	(Add Filter 2 to Port 5)
`>> SLB Port 5# add 2048`	(Add the default filter to Port 5)
`>> SLB Port 5# filt enable`	(Enable filtering for Port 5)
`>> SLB Port 5# /cfg/slb/port 6`	(Select the client Port 6)
`>> SLB Port 6# add 2`	(Add Filter 2 to Port 6)
`>> SLB Port 6# add 2048`	(Add the default filter to Port 6)
`>> SLB Port 6# filt enable`	(Enable filtering for Port 6)

11. Activate Layer 4 services. Apply and verify the configuration.

`>> SLB Port 6# /cfg/slb`	(Select the *Server Load Balancing* menu)
`>> Layer 4# on`	(Activate Layer 4 software services)
`>> Layer 4# apply`	(Make your changes active)
`>> Layer 4# cur`	(View current settings)

SLB must be turned on in order for the application redirection to work properly.

12. Examine the resulting information from the **cur** command. If any settings are incorrect, make appropriate changes.
13. Save your new configuration changes.

```
>> Layer 4# save
```

14. Check the SLB information.

```
>> Layer 4# /info/slb
```

Check that all SLB parameters are working as expected. If necessary, make any appropriate configuration changes and then check the information again.

Note: Changes to filters on a given port only affect new sessions. To make filter changes take effect immediately, clear the session binding table for the port. See the **/oper/slb/clear** command in the *Alteon Application Switch Operating System Command Reference*).

Delayed Binding for Cache Redirection

To configure delayed binding for cache redirection only

```
>> # /cfg/slb/filt <filter number> /adv/layer7/l7lkup ena
```

For more information on delayed binding, see Delayed Binding, page 203.

RTSP Cache Redirection

Alteon supports cache redirection for Real Time Streaming Protocol (RTSP). RTSP cache redirection is similar to HTTP cache redirection. Multimedia presentations consume a lot of Internet bandwidth. The quality of these presentations depends upon the real-time delivery of the data. To ensure the high quality of multimedia presentations, several caching servers are needed to cache the multimedia data locally. This data is then made available quickly from the cache memory as required.

RTSP cache redirection redirects cached data transparently and balances the load among the cache servers. If there is no cache server, the request is directed to the origin server. Internet Service Providers (ISPs) use this feature to cache the multimedia data of a customer site locally. Since the requests for this data are directed to the local cache, they are served faster.

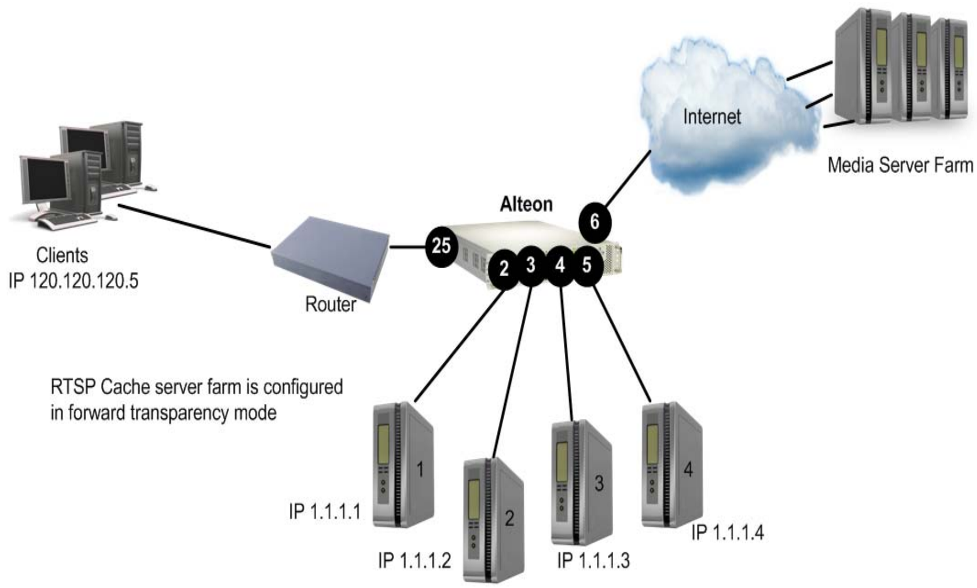
This section explains Layer 4 support for RTSP Streaming Cache Redirection. For detailed information on two prominent commercial RTSP servers (Real Player and QuickTime), see Real Time Streaming Protocol SLB, page 291.

You can also configure Alteon to redirect client requests based on URL content. For information on Layer 7 RTSP Streaming Cache Redirection, see RTSP Streaming Cache Redirection, page 477.

Example RTSP Cache Redirection Configuration

This example is based on Figure 69 - RTSP Cache Redirection Configuration, page 462:

Figure 69: RTSP Cache Redirection Configuration



1. Before configuring RTSP, do the following:
   — Connect each cache server to Alteon
   — Configure the IP addresses on all devices connected to Alteon
   — Configure the IP interfaces on Alteon
2. Configure RTSP cache servers and the IP addresses on Alteon.

`>> # /cfg/slb/real 1`	
`>> Real server 1# rip 1.1.1.1`	(Configure RTSP Cache Server 1)
`>> Real server 1# ena`	(Enable RTSP Cache Server 1)
`>> Real server 1# /cfg/slb/real 2`	
`>> Real server 2# rip 1.1.1.2`	(Configure RTSP Cache Server 2)
`>> Real server 2# ena`	(Enable RTSP Cache Server 2)
`>> Real server 2# /cfg/slb/real 3`	
`>> Real server 3# rip 1.1.1.3`	(Configure RTSP Cache Server 3)
`>> Real server 3# ena`	(Enable RTSP Cache Server 3)
`>> Real server 3# /cfg/slb/real 4`	
`>> Real server 4# rip 1.1.1.4`	(Configure RTSP Cache Server 4)
`>> Real server 4# ena`	(Enable RTSP Cache Server 4)

3. Define a group to load balance the RTSP cache servers.

`>> # /cfg/slb/group 1`	
`>> Real Server Group 1# add 1`	(Add RTSP Cache Server 1 to Group 1)
`>> Real Server Group 1# add 2`	(Add RTSP Cache Server 2 to Group 1)

`>> Real Server Group 1# add 3`	(Add RTSP Cache Server 3 to Group 1)
`>> Real Server Group 1# add 4`	(Add RTSP Cache Server 4 to Group 1)

4. Define the group metric for the RTSP cache servers. RTSP supports all the standard load-balancing metrics.

```
>>Real Server Group 1# metric leastconn
```

5. Configure an RTSP redirection filter to cache data and balance the load among the cache servers.

`>> # /cfg/slb/filt 1`	(Select the menu for Filter 1)
`>> Filter 1# action redir`	(Set the action for redirection)
`>> Filter 1# proto tcp`	(Enter TCP protocol)
`>> Filter 1# dport rtsp`	(Enter service port for RTSP)
`>> Filter 1# rport rtsp`	(Enter redirection port for RTSP)
`>> Filter 1# group 1`	(Select RTSP cache server Group 1)
`>> Filter 1# adv/proxyadv`	(Select advanced menu for Filter 1)
`>> Filter 1# Advanced# proxy disable`	(Disable proxy)

6. Configure a default allow filter to facilitate traffic.

`>> # /cfg/slb/filt 2048`	(Select a default allow filter 2048)
`>> Filter 2048# sip any`	(From any source IP addresses)
`>> Filter 2048# dip any`	(To any destination IP addresses)
`>> Filter 2048# ena`	(Enable a default allow filter)
`>> Filter 2048# action allow`	(Set the action to allow normal traffic)

7. Add and enable the redirection filter on the port to support basic cache redirection.

`>> # /cfg/slb/port 25`	(Select the menu for Port 25)
`>> SLB Port 25# add 1`	(Add RTSP filter 1 to Port 25)
`>> SLB Port 25# add 2048`	(Add default filter 2048 to Port 25)
`>> SLB Port 25# filt ena`	(Enable filtering on Port 25)

8. Apply and save the configuration.

```
>> SLB Port 25# apply
>> SLB Port 25# save
```

IP Proxy Addresses for NAT

Transparent proxies provide the following benefits when used with application redirection. Application redirection is enabled when a filter with the **redir** action is applied on a port.

- With proxy IP addresses configured on ports that use redirection filters, Alteon can redirect client requests to servers located on any subnet.
- Alteon can perform transparent substitution for all source and destination addresses, including destination port remapping. This provides support for comprehensive, fully-transparent proxies. No additional client configuration is needed.

The following procedure can be used for configuring proxy IP addresses:

1. Configure proxy IP addresses and enable proxy for the redirection ports.

   Each of the ports using redirection filters require proxy IP addresses. For more information on proxy IP addresses, see Client Network Address Translation (Proxy IP), page 190.
2. In this example, proxy IP addresses are configured.

`>> SLB port 3# /cfg/slb/pip`	(Select proxy IP address menu)
`>> Proxy IP address# type port`	(Use port-based proxy IP)
`>> Proxy IP Address# add 200.200.200.68`	(Set proxy IP address)
`>> Proxy IP Address# add 200.200.200.69`	(Set proxy IP address)
`>> Proxy IP Address# add 200.200.200.70`	(Set proxy IP address)
`>> Proxy IP Address# add 200.200.200.71`	(Set proxy IP address)
`>> Proxy IP Address# /cfg/slb/port 1`	(Select Port 1)
`>> SLB port 1# proxy ena`	(Enable Proxy Port 1)
`>> SLB port 1# /cfg/slb/port 2`	(Select Port 2)
`>> SLB port 2# proxy ena`	(Enable Proxy Port 2)
`>> SLB port 2# /cfg/slb/port 3`	(Select Port 3)
`>> SLB port 3# proxy ena`	(Enable Proxy Port 3)
`>> SLB port 3# /cfg/slb/port 4`	(Select Port 4)
`>> SLB port 4# proxy ena`	(Enable Proxy Port 4)
`>> SLB port 4# /cfg/slb/port 5`	(Select Port 5)
`>> SLB port 5# proxy ena`	(Enable Proxy Port 5)
`>> SLB port 5# /cfg/slb/port 6`	(Select Port 6)
`>> SLB port 6# proxy ena`	(Enable Proxy Port 6)

3. Configure the application redirection filters. Once proxy IP addresses are established, configure each application redirection filter (Filter 2 in this example) with the real server TCP or UDP port to which redirected traffic will be sent. In this case, the requests are mapped to a different destination port (8080). You must also enable proxies on the real servers:

`>> # /cfg/slb/filt 2`	(Select the menu for Filter 2)
`>> Filter 2# rport 8080`	(Set proxy redirection port)
`>> Filter 2# /cfg/slb/real 1/adv/proxy enable`	(Enable proxy on real servers)
`>> Real server 1# /cfg/slb/real 2/adv/proxy enable`	(Enable proxy on real servers)
`>> Real server 2# /cfg/slb/real 3/adv/proxy enable`	(Enable proxy on real servers)

Note: This configuration is not limited to the HTTP (Web) service. Other TCP/IP services can be configured in a similar fashion. For example, if this had been a DNS redirect, rport would be sent to well-known port 53 (or the service port you want to remap to). For a list of other well-known services and ports, see the Well-Known Application Ports, page 175.

4. Apply and save your changes.
5. Check server statistics to verify that traffic has been redirected based on filtering criteria.

```
>> # /info/slb/group <group number> /filter <filter number>
```

Excluding Non-Cacheable Sites

Some sites provide content that is not well suited for redirection to cache servers. Such sites might provide browser-based games or applications that keep real-time session information or authenticate by client IP address.

To prevent such sites from being redirected to cache servers, create a filter that allows this specific traffic to pass normally through Alteon. This filter must have a higher precedence (a lower filter number) than the application redirection filter.

For example, if you want to prevent a popular Web-based game site on subnet 200.10.10.* from being redirected, you could add the following to the previous example configuration:

`>> # /cfg/slb/filt 1`	(Select the menu for Filter 1)
`>> Filter 1# dip 200.10.10.0`	(To the site's destination IP address)
`>> Filter 1# dmask 255.255.255.0`	(For entire subnet range)
`>> Filter 1# sip any`	(From any source IP address)
`>> Filter 1# proto tcp`	(For TCP traffic)
`>> Filter 1# dport http`	(To an HTTP destination port)
`>> Filter 1# sport any`	(From any source port)
`>> Filter 1# action allow`	(Allow matching traffic to pass)
`>> Filter 1# ena`	(Enable the filter)
`>> Filter 1# /cfg/slb/port 5`	(Select SLB Port 5)
`>> SLB port 5# add 1`	(Add the filter to Port 5)

`>> SLB port 5# /cfg/slb/port 6`	(Select SLB Port 6)
`>> SLB port 6# add 1`	(Add the filter to Port 6)
`>> SLB port 6# apply`	(Apply configuration changes)
`>> SLB port 6# save`	(Save configuration changes)

Content-Intelligent Cache Redirection

Alteon lets you redirect cache requests based on different Layer 7 content for HTTP header information such as "Host:" header or "User-Agent" for browser-smart load balancing.

The No Cache/Cache-Control for cache redirection lets you offload the processing of non-cacheable content from cache servers by sending only appropriate requests to the cache server farm. When a Cache-Control header is present in a HTTP 1.1 request, it indicates a client's special request with respect to caching, such as to guarantee up-to-date data from the origin server. If this feature (Cache-Control: no cache directive) is enabled, HTTP 1.1 GET requests are forwarded directly to the origin servers.

Note: Origin server refers to the server originally specified in the request.

The HTTP 1.0 **Pragma: no-cache** header is equivalent to the HTTP 1.1 **Cache-Control** header. By enabling the Pragma: no-cache header, requests are forwarded to the origin server.

For cache redirection, at any given time one HTTP header is supported globally on Alteon. This section discusses the following types of cache redirection:

- URL-Based Cache Redirection, page 466
- HTTP Header-Based Cache Redirection, page 472
- Browser-Based Cache Redirection, page 474
- URL Hashing for Cache Redirection, page 475
- RTSP Streaming Cache Redirection, page 477

URL-Based Cache Redirection

URL parsing for cache redirection operates in a manner similar to URL-based server load balancing, except that in cache redirection a virtual server is the target of all IP/HTTP requests.

By separating static and dynamic content requests via URL parsing, Alteon enables you to send requests with specific URLs or URL strings to designated cache servers. The URL-based cache redirection option lets you offload overhead processing from the cache servers by only sending appropriate requests to the cache server farm.

Note: Both HTTP 1.0 and HTTP 1.1 requests are supported.

Each request is examined and handled as described below:

- If the request is a non-GET request such as HEAD, POST, PUT, or HTTP with cookies, it is not sent to the cache.
- If the request is an ASP or CGI request or a dynamically generated page, it is not sent to the cache.
- If the request contains a cookie, it can optionally bypass the cache.

Examples of matching string expressions are:

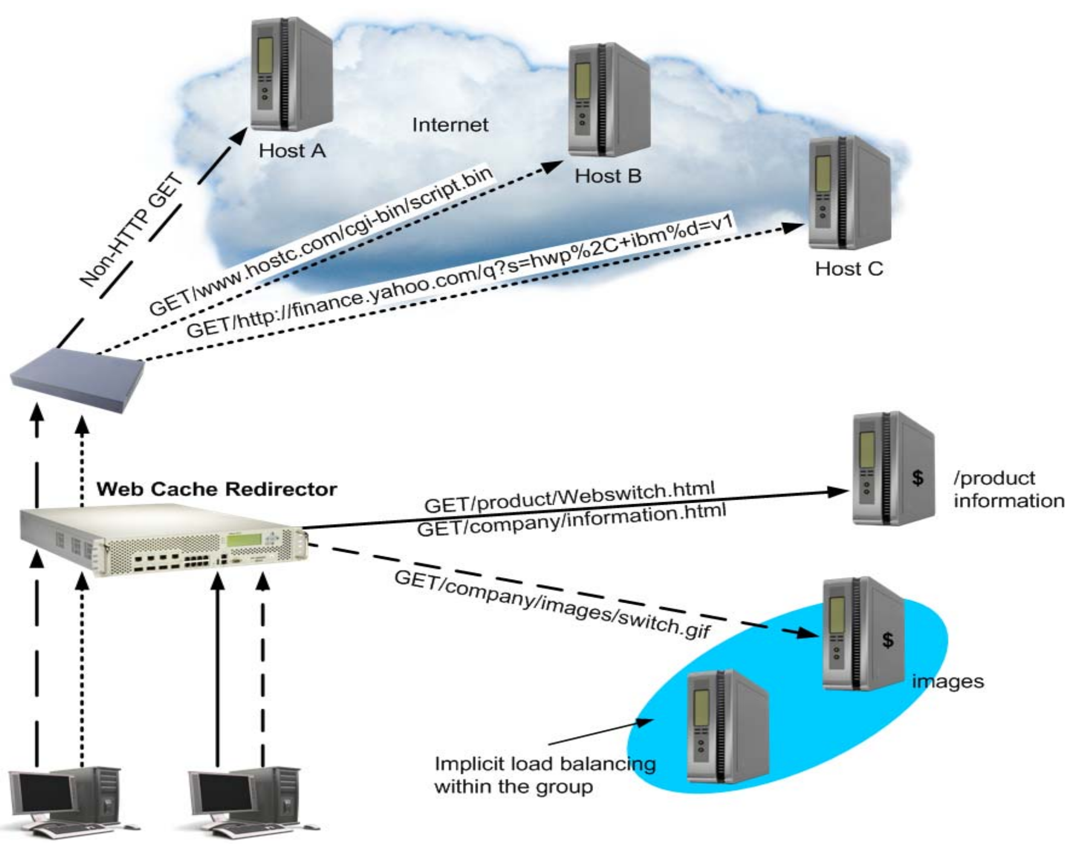
- `/product`—Any URL that starts with "/product," including any information in the "/product" directory.
- `product`—Any URL that has the string "product".

Some of the common noncacheable items that you can configure to add, delete, or modify are:

- Dynamic content files:
  - Common gateway interface files (.cgi)
  - Cold fusion files (.cfm), ASP files (.asp)
  - BIN directory
  - CGI-BIN directory
  - SHTML (scripted html)
  - Microsoft HTML extension files (.htx)
  - Executable files (.exe)
- Dynamic URL parameters: +, !, %, =, &

As shown in Figure 70 - URL-Based Cache Redirection, page 467, requests matching the URL are load balanced among the multiple servers, depending on the metric specified for the real server group (**leastconns** is the default).

Figure 70: URL-Based Cache Redirection



Network Address Translation Options

URL-based cache redirection supports three types of Network Address Translation (NAT):

- **No NAT—**Traffic is redirected to the cache with the destination MAC address of the virtual server replaced by the MAC address of the cache. The destination IP address remains unchanged, and no modifications are made to the IP address or the MAC address of the source or origin server. This works well for transparent cache servers, which process traffic destined to their MAC address but use the IP address of some other device.
- **Half NAT—**In this most commonly used NAT method, the destination IP address is replaced by the IP address of the cache, and the destination MAC address is replaced by the MAC address of the cache. Both the IP address and the MAC address of the source remain unchanged.
- **Full NAT—**The source IP address and the source MAC address are replaced by the IP address and MAC address of the cache. This method works well for proxy cache servers.

Configuring URL-Based Cache Redirection

This procedure is an example configuration for URL-based cache redirection.

To configure URL-based cache redirection

1. Before you can configure URL-based cache redirection, configure Alteon for basic SLB with the following tasks:
   — Assign an IP address to each of the real servers in the server pool.
   — Define an IP interface.
   — Define each real server.

   For information on how to configure your network for SLB, see Server Load Balancing, page 165.
2. Configure Alteon to support basic cache redirection.

   For information on cache redirection, refer to Application Redirection, page 455.
3. Configure the parameters and file extensions that bypass cache redirection.

   a. Add or remove string IDs that should *not* be cacheable.

```
>> # /cfg/slb/filt 1/adv/layer7/addstr|remstr <ID>
>> # /cfg/slb/layer7/slb/addstr|remstr <strings>
```

   b. Enable or disable ALLOW for non-GETS (such as HEAD, POST, and PUT) to the origin server:

```
>> # /cfg/slb/layer7/redir/urlal {ena|dis}
```

   - **ena—**Alteon allows all non-GET requests to the origin server.
   - **dis—**Alteon compares all requests against the expression table to determine whether the request should be redirected to a cache server or the origin server.

   c. Enable or disable cache redirection of requests that contain the string “cookie:**”** in the HTTP header:

```
>> # /cfg/slb/layer7/redir/cookie {ena|dis}
```

   - **ena—**Alteon redirects all requests that contain "cookie:" in the HTTP header to the origin server.
   - **dis—**Alteon compares the URL against the expression table to determine whether the request should be redirected to a cache server or the origin server.

d. Enable or disable cache redirection of requests that contain the string "Cache-control:no cache" in the HTTP 1.1 header or the string "Pragma:no cache" in the HTTP 1.0 header to the origin server.

```
>> # /cfg/slb/layer7/redir/nocache {ena|dis}
```

- **ena—**Alteon redirects all requests that contain the string “Cache-control: no cache” in the HTTP 1.1 header or the string “Pragma:no cache” in the HTTP 1.0 header to the origin server.
- **dis—**Alteon compares the URL against the expression table to determine whether the request should be redirected to a cache server or the origin server.

4. Define the strings to be used for cache SLB:

```
>> # /cfg/slb/layer7/slb/{addstr|remstr} <string>
```

— **addstr—**Add a string or a path.

— **remstr—**Remove string or a path.

A default string **any** indicates that the particular server can handle all URL or cache requests. Refer to the following examples:

Example 1: String Starting with the Forward Slash (/)

A string that starts with a forward slash (/), such as "/images," indicates that the server will process requests that start with the "/images" string only.

With the "/images" string, the server will handle these requests:

```
/images/product/b.gif
/images/company/a.gif
/images/testing/c.jpg
```

The server will not handle these requests:

```
/company/images/b.gif
/product/images/c.gif
/testing/images/a.gif
```

Example 2: String Without the Forward Slash (/)

A string that does not start out with a forward slash (/) indicates that the server will process any requests that contain the defined string.

With the "images" string, the server will process these requests:

```
/images/product/b.gif
/images/company/a.gif
/images/testing/c.jpg
/company/images/b.gif
/product/images/c.gif
/testing/images/a.gif
```

Example 3: String with the Forward Slash (/) Only

If a server is configured with the load balance string (/) only, it will only handle requests to the root directory.

The server will handle any files in the ROOT directory:

```
//index.htm
/default.asp
/index.shtm
```

5. Apply and save your configuration changes.
6. Identify the defined string IDs.

```
>> # /cfg/slb/layer7/slb/cur
```

For easy configuration and identification, each defined string has an ID attached, as shown in Table 42 - SLB Strings, page 470.

Table 42: SLB Strings

ID	SLB String
1	any
2	.gif
3	/sales
4	/xitami
5	/manual
6	.jpg

7. Configure the real servers to support cache redirection.

Note: If you do not add a defined string (or add the defined string **any**), the server will handle any request.

Add the defined strings to the real servers, where *ID* is the identification number of the defined string:

```
>> # /cfg/slb/real 2/layer7/addlb <ID>
```

The server can have multiple defined strings. For example: "/images", "/sales", ".gif"

With these defined strings, the server can handle requests that begin with "/images" or "/sales" and any requests that contain ".gif".

8. Define a real server group and add real servers to the group. The following configuration combines three real servers into a group.

```
>> # /cfg/slb/group 1                  (Select Real Server Group 1)
>> Real server group 1# add 1          (Add Real Server 1 to Group 1)
>> Real server group 1# add 2          (Add Real Server 2 to Group 1)
>> Real server group 1# add 3          (Add Real Server 3 to Group 1)
```

9. Configure a filter to support basic cache redirection.

The filter must be able to intercept all TCP traffic for the HTTP destination port and must redirect it to the proper port in the real server group.

`>> # /cfg/slb/filt <filter number>`	(Select the menu for *Filter #*)
`>> Filter <filter number> # sip any`	(From any source IP addresses)
`>> Filter <filter number> # dip any`	(To any destination IP addresses)
`>> Filter <filter number> # proto tcp`	(For TCP protocol traffic)
`>> Filter <filter number> # sport any`	(From any source port)
`>> Filter <filter number> # dport http`	(To an HTTP destination port)
`>> Filter <filter number> # action redir`	(Set the action for redirection)
`>> Filter <filter number> # rport http`	(Set the redirection port)
`>> Filter <filter number> # group 1`	(Select real server group 1)
`>> Filter <filter number> # ena`	(Enable the filter)

10. Enable URL-based cache redirection on the same filter.

```
>> # /cfg/slb/filt <filter number> /adv/layer7/l7lkup ena
```

11. Select the appropriate NAT option. The three NAT options are listed below. For more information about each option, see Network Address Translation Options, page 468.
    — No NAT option:

```
>> # /cfg/slb/filter <filter number> /adv/proxyadv/proxy dis
```

— Half NAT option:

```
>> # /cfg/slb/filter <filter number> /adv/proxyadv/proxy ena
```

— Full NAT option:

`>> # /cfg/slb/pip`	
`>> Proxy IP Address# add 12.12.12.12`	(Configure proxy IP address)
`>> # /cfg/slb/filt <filter number>`	
`>> Filter <filter number> # rport 3128`	(Specify redirection port)
`>> Filter <filter number> # adv/proxyadv`	(Select the advance menu)
`>> Filter <filter number> Advanced# proxy ena`	(Enable proxy IP address)

For more information on proxy IP addresses, see Client Network Address Translation (Proxy IP), page 190.

12. Create a default filter for non-cached traffic.

`>> # /cfg/slb/filt <filter number>`	(Select the default filter)
`>> Filter <filter number> # sip any`	(From any source IP addresses)
`>> Filter <filter number> # dip any`	(To any destination IP addresses)
`>> Filter <filter number> # proto any`	(For any protocol traffic)
`>> Filter <filter number> # action allow`	(Set the action to allow traffic)

`>> Filter <filter number> # ena`	(Enable the default filter)
`>> Filter <filter number> # port <port number>`	(Assign the default filter to a port)

When the proto parameter is **not tcp or udp**, then sport and dport are ignored.

13. Turn on filtering for the port.

```
>> SLB <port number> # filt ena
```

14. Add the filters to the client port.

```
>> SLB <port number> # add <filter number>
```

15. Enable Direct Access Mode (DAM).

```
>> SLB <port number> # /cfg/slb/adv
>> Layer 4 Advanced# direct ena
```

16. Enable, apply, and verify the configuration.

`>> # /cfg/slb`	(Select the *SLB* menu)
`>> # on`	(Turn SLB on)
`>> # apply`	(Make your changes active)
`>> # cur`	(View current settings)

Viewing Statistics for URL-Based Cache Redirection

To show the number of hits to the cache server or origin server, use the following command:

```
>> # /stats/slb/layer7/redir
Total URL based Web cache redirection stats:
Total cache server hits:                    73942
Total origin server hits:                   2244
Total straight to origin server hits:       0
Total none-GETs hits:                       53467
Total 'Cookie: ' hits:                      729
Total no-cache hits:                        43
Total RTSP cache server hits:               0
Total RTSP origin server hits:              0
Total HTTP redirection hits:                0
```

HTTP Header-Based Cache Redirection

This procedure is an example configuration for HTTP header-based cache redirection.

To configure Alteon for cache direction based on the "Host:" header

1. Before you can configure header-based cache redirection, ensure that Alteon is configured for basic SLB (see Server Load Balancing, page 165):
   — Assign an IP address to each of the real servers in the server pool.

- Define an IP interface.
- Define each real server.
- Assign servers to real server groups.
- Define virtual servers and services.

2. Turn on Layer 7 lookup for the filter.

```
>> # /cfg/slb/filt 1/adv/layer7/l7lkup ena
```

3. Enable header load balancing for the Host: header.

```
>> # /cfg/slb/layer7/redir/header ena host
```

4. Define the hostnames.

```
>> # /cfg/slb/layer7/slb/addstr ".com"
>> Server Load Balance Resource# add ".org"
>> Server Load Balance Resource# add ".net"
```

5. Apply and save your configuration changes.
6. Identify the string ID numbers with this command.

```
>> # /cfg/slb/layer7/slb/cur
```

Each defined string has an associated ID number:

Table 43: SLB Strings

ID	SLB String
1	any
2	.com
3	.org
4	.net

7. Configure the real servers to handle the appropriate load balance strings.
8. Add the defined string IDs to the real servers, where *ID* is the identification number of the defined string.

```
>> # /cfg/slb/real 2/layer7/addlb <ID>
```

Note: If you do not add a defined string (or add ID=1), the server will handle any request.

Browser-Based Cache Redirection

Browser-based cache redirection uses the User-agent: header.

To configure browser-based cache redirection

1. Before you can configure header-based cache redirection, ensure that Alteon is configured for basic SLB:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface.
   - Define each real server.
   - Assign servers to real server groups.
   - Define virtual servers and services.
2. Turn on Layer 7 lookup for the filter.

```
>> # /cfg/slb/filt 1/adv/layer7/l7lkup enable
```

3. Enable header load balancing for "User-Agent:" header.

```
>> # /cfg/slb/layer7/redir/header ena useragent
```

4. Define the hostnames.

```
>> # /cfg/slb/layer7/slb/addstr "Mozilla"
>> Server Load Balance Resource# add "Internet Explorer"
>> Server Load Balance Resource# add "Netscape"
```

5. Apply and save your configuration changes.
6. Identify the string ID numbers with this command.

```
>> # /cfg/slb/layer7/slb/cur
```

Each defined string has an ID number. Number of entries: four

ID	SLB String
1	any
2	Mozilla
3	Internet Explorer
4	Netscape

7. Add the defined string IDs to configure the real servers to handle the appropriate load balance strings, where *ID* is the identification number of the defined string.

```
>> # /cfg/slb/real 2/layer7/addlb <ID>
```

If you do not add a defined string (or add the ID 1), the server will handle any request.

URL Hashing for Cache Redirection

By default, hashing algorithms use the source IP address and/or destination IP address (depending on the application area) to determine content location. For example, FWLB uses both source and destination IP addresses, cache redirection uses only the destination IP address, and SLB uses only the source IP address.

Hashing is based on the URL, including the HTTP Host header (if present), up to a maximum of 255 bytes. You can optimize *cache hits* by using the hashing algorithm to redirect client requests going to the same page of an origin server to a specific cache server.

For example, Alteon could use the string "radware.com/products/Alteon/" for hashing the following request:

```
GET http://products/Alteon/ HTTP/1.0
HOST:www.radware.com
```

To configure Alteon for cache redirection based on a hash key

1. Configure basic SLB.

   Before you can configure header-based cache redirection, ensure that Alteon is configured for basic SLB (see Server Load Balancing, page 165):

   — Assign an IP address to each of the real servers in the server pool.

   — Define an IP interface.

   — Define each real server.

   — Assign servers to real server groups.

   — Define virtual servers and services.

   — Configure the load-balancing algorithm to hash or minmiss.

2. Turn on Layer 7 lookup for the filter.

```
>> # /cfg/slb/filt 1/adv/layer7/l7lkup enable
```

3. Enable hash to direct a cacheable URL request to a specific cache server.

   By default, the host header field is used to calculate the hash key and URL hashing is disabled.

   — **hash ena**—Enables hashing based on the URL and the host header if it is present. Specify the length of the URL to hash into the cache server:

```
>> # /cfg/slb/layer7/redir/hash ena
Enter new hash length [1-255]: 24
```

   — **hash disable**—Disables hashing based on the URL. Instead, the host header field to calculate the hash key.

   If the host header field does not exist in the HTTP header, then Alteon uses the source IP address as the hash key.

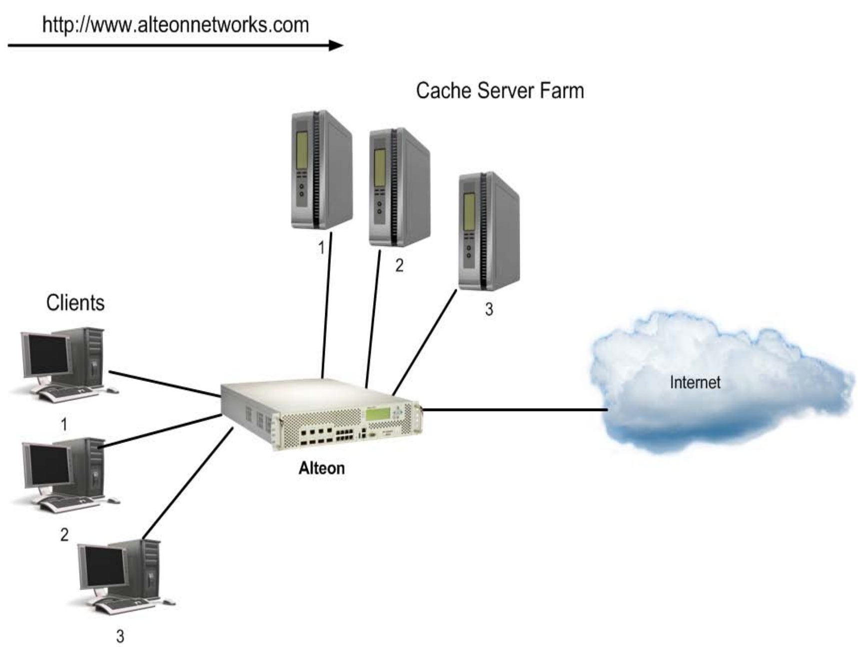
Examples

A Hashing on the URL

In this example, URL hashing is enabled. If the host field does not exist, the specified length of the URL is used to hash into the cache server as shown in Figure 71 - URL Hashing for Application Redirection, page 476. If the host field exists, the specified length of both the host field and the URL is used to hash into the cache server. The same URL request goes to the same cache server:

— Client 1 request http://www.radware.com/sales/index.htmis directed to cache server 1.
— Client 2 request http://www.radware.com/sales/index.htmis directed to cache server 1.
— Client 3 request http://www.radware.com/sales/index.htm is directed to cache server 1.

Figure 71: URL Hashing for Application Redirection



B Hashing on the Host Header Field Only

In this example, URL hashing is disabled. If you use the host header field to calculate the hash key, the same URL request goes to the same cache server:

— Client 1 request http://www.radware.com is directed to cache server 1.
— Client 2 request http://www.radware.com is directed to cache server 1.
— Client 3 request http://www.radware.com is directed to cache server 1.

C Hashing on the Source IP Address

In this example, URL hashing is disabled. Because the host header field does not exist in the HTTP header, the source IP address is used as the hash key and requests from clients 1, 2, and 3 are directed to three different cache servers:

— Client 1 request http://www.radware.com is directed to cache server 1.
— Client 2 request http://www.radware.com is directed to cache server 2.
— Client 3 request http://www.radware.com is directed to cache server 3.

RTSP Streaming Cache Redirection

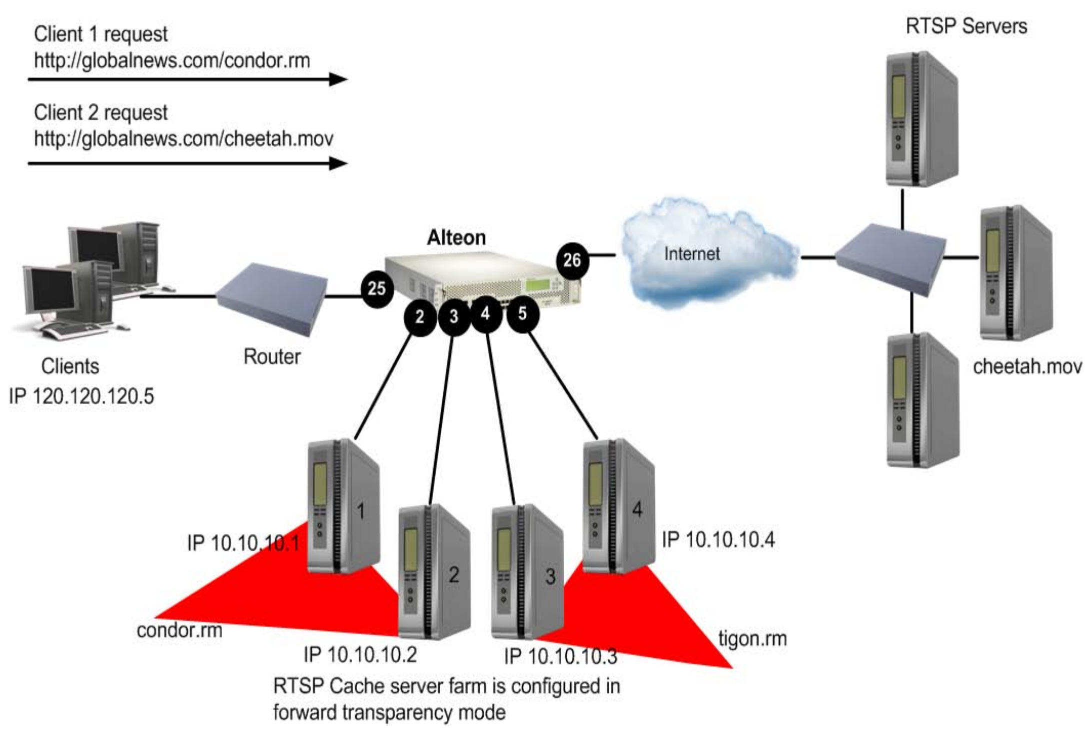
RTSP load balancing with the URL *hash* metric can be used to load balance cache servers that cache multimedia presentations. Since multimedia presentations consume a large amount of Internet bandwidth, and their correct presentation depends upon the real-time delivery of the data over the Internet, several caching servers cache the multimedia data.

As a result, the data is available quickly from the cache, when required. The Layer 7 metric of URL hashing directs all requests with the same URL to the same cache server, ensuring that no data is duplicated across the cache servers. All the stream connections and the control connections are switched to the same cache server to facilitate caching of entire presentations.

This section explains Layer 7 support for RTSP Streaming Cache Redirection. For more information on RTSP Streaming Cache Redirection, see RTSP Cache Redirection, page 461. For detailed information on two prominent commercial RTSP servers (Real Player and QuickTime), see Real Time Streaming Protocol SLB, page 291.

As shown in Figure 72 - RTSP Steaming Cache Redirection, page 477, the cache servers are configured for forward proxy mode. The cache servers process the client request even though the destination IP address is not destined for the cache servers.

Figure 72: RTSP Steaming Cache Redirection



To configure RTSP streaming cache redirection

This procedure is based on Figure 72 - RTSP Steaming Cache Redirection, page 477.

1. Before you start configuring this feature, do the following:
   — Connect each cache server to the Alteon appliance.
   — Configure the IP addresses on all devices connected to Alteon.
   — Configure the IP interfaces.

2. Configure RTSP cache servers and the IP addresses.

`>> # /cfg/slb/real 1`	
`>> Real server 1# rip 1.1.1.1`	(Configure RTSP Cache Server 1)
`>> Real server 1# ena`	(Enable RTSP Cache Server 1)
`>> Real server 1# /cfg/slb/real 2`	
`>> Real server 2# rip 1.1.1.2`	(Configure RTSP Cache Server 2)
`>> Real server 2# ena`	(Enable RTSP Cache Server 2)
`>> Real server 2# /cfg/slb/real 3`	
`>> Real server 3# rip 1.1.1.3`	(Configure RTSP Cache Server 3)
`>> Real server 3# ena`	(Enable RTSP Cache Server 3)
`>> Real server 3# /cfg/slb/real 4`	
`>> Real server 4# rip 1.1.1.4`	(Configure RTSP Cache Server 4)
`>> Real server 4# ena`	(Enable RTSP Cache Server 4)

3. Define a group to load balance the RTSP cache servers.

`>> # /cfg/slb/group 1`	
`>> Real Server Group 1# add 1`	(Add RTSP Cache Server 1 to Group 1)
`>> Real Server Group 1# add 2`	(Add RTSP Cache Server 2 to Group 1)
`>> Real Server Group 1# add 3`	(Add RTSP Cache Server 3 to Group 1)
`>> Real Server Group 1# add 4`	(Add RTSP Cache Server 4 to Group 1)

4. Configure a redirection filter.

`>> # /cfg/slb/filter 100`	(Select the menu for filter 100)
`>> Filter 100# action redir`	(Set the action for redirection)
`>> Filter 100# proto tcp`	(Enter TCP protocol)
`>> Filter 100# dport rtsp`	(Enter service port for RTSP)
`>> Filter 100# rport rtsp`	(Enter redirection port for RTSP)
`>> Filter 100# group 1`	(Select RTSP cache server group 1)
`>> Filter 100# adv/proxyadv`	(Select the *Advanced* menu for filter 100)
`>> Filter 100# Advanced# proxy disable`	(Disable proxy)

5. Enable Layer 7 lookup for the redirection filter 100.

```
>> Filter 100 Advanced# layer7/l7lkup ena
```

6. Configure a default allow filter to facilitate traffic.

`>> # /cfg/slb/filt 2048`	(Select a default allow filter 2048)
`>> Filter 2048# sip any`	(From any source IP addresses)
`>> Filter 2048# dip any`	(To any destination IP addresses)

`>> Filter 2048# ena`	(Enable a default allow filter)
`>> Filter 2048# action allow`	(Set the action to allow normal traffic)

7. Add and enable the redirection filter to the port.

`>> # /cfg/slb/port 25`	(Select the menu for port 25)
`>> SLB Port 25# add 100`	(Add RTSP filter 100 to port 25)
`>> SLB Port 25# add 2048`	(Add default filter 2048 to port 25)
`>> SLB Port 25# filt ena`	(Enable filtering on port 25)

8. Configure the parameters and file extensions that will bypass RTSP streaming cache redirection. This is for user-defined, non-cacheable content.

   For example, QuickTime files are non-cacheable—RTSP files with the extension *.mov must bypass the streaming cache redirection. Similarly, you can add other RTSP file extensions (such as *.smil, *.rm, *.ram, and so forth) to bypass the redirection.

`>> # /cfg/slb/layer7/slb`	(Select the SLB resource menu)
`>> # addstr *.mov`	(Add non-cacheable RTSP strings)

   A client request of the form “RTSP://*.mov” bypasses the cache servers and instead is routed directly to the original servers.

9. Under the filter menu, add the string IDs that need to be excluded.

`>> /cfg/slb/filt 20/adv/layer7`	(Select the *Filtering Layer 7 Advanced* menu)
`>> Layer 7 Advanced# addstr 2`	(Add the string ID for **`*.mov`**)

10. Define the RTSP file extensions to load balance among the cache servers.

```
>> # /cfg/slb/layer7/slb/addstr condor.rm
>> Server Load Balance Resource# addstr tiger.rm
```

11. Apply and save your configuration changes.
12. Identify the associated ID number for each of the defined RTSP file extension.

```
>> # /cfg/slb/layer7/slb/cur
```

Table 44: SLB Strings

ID	SLB String
1	any
2	*.mov
3	condor.rm
4	tiger.rm

13. Assign the URL string ID to the cache servers.

```
>> # /cfg/slb/real 1                                   (Select the Real Server 1)
>> Real Server 1# Layer 7/addlb 3                      (Add the URL string ID 3)
>> Real Server 1 Layer 7 Commands# cfg/slb/real 2
>> Real Server 2# Layer 7/addlb 3                      (Add the URL string ID 3)
>> Real Server 2 Layer 7 Commands# cfg/slb/real 3
>> Real Server 3# Layer 7/addlb 4                      (Add the URL string ID 4)
>> Real Server 3 Layer 7 Commands# cfg/slb/real 4
>> Real Server 4# Layer 7/addlb 4                      (Add the URL string ID 4)
```

Note: If no string is assigned to the server, the server will handle all requests.

14. Apply and save the configuration.

```
>> Real Server 4 Layer 7 Commands# apply
>> Real Server 4 Layer 7 Commands# save
```

Client requests “condor.rm**”** or **“**tiger.rm” are retrieved from the local Cache servers 1 or 2 and 3 or 4 respectively. However, a client request “cheetah.mov” bypasses the local cache servers and is forwarded to the original server.

Peer-to-Peer Cache Load Balancing

The pattern matching filter redirection feature load balances peer-to-peer caches. The pattern matching filter redirection feature supports ALLOW, DENY, and REDIR actions. For more information on this topic, see Filtering and Traffic Manipulation, page 355.

There are two instances where a packet will be redirected because of a pattern matching filter:

1. The packet matches a previously configured filter with a REDIR action.
2. A packet earlier in the session was matched against a filter configured with a REDIR action and the session has been converted to a redirect session. In this instance, subsequent packets after the initial match are not subjected to pattern matching.

   Packet redirection is accomplished by substituting the original destination MAC address with the real server MAC address. Some applications, however, require that all of the Layer 2 information remain unmodified in the redirected packet. To support instances where this is the case, you can disable destination MAC address substitution on a real server by real server basis. With this option enabled, all packets will be transparently redirected and no destination MAC address substitution will take place.

Note: Disabling destination MAC address substitution is only available for filter redirection.

To disable destination MAC address substitution, issue the following command:

```
>> Main# /cfg/slb/real <real server number> /adv/subdmac disable
```

Chapter 18 – Health Checking

Health checking allows you to verify content accessibility in large Web sites. As content grows and information is distributed across different server farms, flexible, customizable content health checks are critical to ensure end-to-end availability.

The following health-checking topics are described in this chapter:

- Understanding Health Check Monitoring, page 482—Describes the use of template health checks and reusable health checks, and how to assign them to real servers and groups.
- Supported Health Check Types, page 484—Lists all the supported health check types available:
  - Link Health Checks, page 485—Describes how to perform Layer 1 health checking on an Intrusion Detection Server (IDS).
  - TCP Health Checks, page 485—TCP health checks help verify the TCP applications that cannot be scripted.
  - UDP Health Checks, page 485—UDP health checks help verify the UDP applications that cannot be scripted.
  - ICMP Health Checks, page 486—Explains how ICMP health checks are used for UDP services.
  - HTTP/S Health Checks, page 486—Provides examples of HTTP-based health checks using hostnames.
  - TCP and UDP-based DNS Health Checks, page 488—Explains the functionality of the DNS Health Checks using UDP packets.
  - TFTP Health Check, page 488—Explains how to health check a real server using the TFTP protocol.
  - SNMP Health Check, page 488—Explains how to perform SNMP health checks to real servers running SNMP Agents.
  - FTP Server Health Checks, page 489—Describes how the File Transfer Protocol (FTP) server is used to perform health checks and explains how to configure Alteon to perform FTP health checks.
  - POP3 Server Health Checks, page 489—Explains how to use Post Office Protocol Version 3 (POP3) mail server to perform health checks between a client system and a mail server and how to configure Alteon for POP3 health checks.
  - SMTP Server Health Checks, page 490—Explains how to use Simple Mail Transfer Protocol (SMTP) mail server to perform health checks between a client system and a mail server and how to configure Alteon for SMTP health checks.
  - IMAP Server Health Checks, page 490—Describes how the mail server Internet Message Access Protocol (IMAP) protocol is used to perform health checks between a client system and a mail server.
  - NNTP Server Health Checks, page 490—Explains how to use Network News Transfer Protocol (NNTP) server to perform health checks between a client system and a mail server and how to configure Alteon for NNTP health checks
  - RADIUS Server Health Checks, page 490—Explains how the RADIUS protocol is used to authenticate dial-up users to Remote Access Servers (RASs).
  - SSL HELLO Health Checks, page 491—Explains how Alteon queries the health of the SSL servers by sending an SSL client “Hello” packet and then verifies the contents of the server's “Hello” response.
  - WAP Gateway Health Checks, page 491—Discusses how Alteon provides connection-less and connection-oriented WSP health check for WAP gateways.
  - LDAP/LDAPS Health Checks, page 492—Describes how to configure Alteon to perform Lightweight Directory Access Protocol (LDAP) health checks for Alteon to determine if the LDAP server is running.

- ARP Health Checks, page 493—Describes how to perform health checks on Intrusion Detection Servers (IDS) that do not have full TCP/IP stack support.
- RTSP Health Checks, page 494—Describes how to perform RTSP health checks.
- Script-Based Health Checks, page 495—Describes how to configure Alteon to send a series of health-check requests to real servers or real server groups and monitor the responses. Health checks are supported for TCP and UDP protocols, using either Binary or ASCII content.

- Pre-defined Health Check Summary, page 502—Lists all available out-of-the-box health check objects.
- Failure Types, page 503—Explains the service failed and server failed states.
- DSR Health Checks, page 505—Describes the servers' ability to respond to the client queries made to the Virtual server IP address when the server is in Direct Server Return (DSR) mode.
- Advanced Group Health Check, page 505—Describes how to configure an expression to fine-tune the selected health check for a real server group.
- Disabling the Fast Link Health Check, page 506—Describes how to disable fast link health checks.

Understanding Health Check Monitoring

Monitoring the availability of real servers and groups is an important component in any Application Delivery Controller. Detection of real server failure is critical in ensuring continuous service.

Alteon allows to accurately monitor the health and performance (response time) of real servers and the applications running on the servers using a wide range of health check types.

For increased flexibility, you can monitor server availability based on multiple health check types or availability of additional elements by defining complex health checks (Advanced Health Checks) as logical expression of basic health checks.

Alteon health checks are reusable objects that can be assigned to multiple monitored objects. The health check library includes:

- Pre-defined basic health checks that can be assigned to monitored objects
- User-defined basic health checks
- User-defined advanced health checks (logical expression on basic health checks)

Alteon health checks can be assigned to:

- Server Groups—A health check assigned to a server group monitors each of the servers in the group.
- Real Servers—A health check assigned to a real server monitors that server and overrides health check assigned to server groups to which it belongs.

Pre-defined Health Checks

Alteon provides out-of-the-box health checks for most popular applications. The purpose of pre-defined health checks is saving time by allowing you to quickly define group health checks without having to configure a health check object first. Pre-defined health checks cannot be edited (with the exception of WAP health checks) and are meant to be used as is.

For a full list of available pre-defined health checks, see Pre-defined Health Check Summary, page 502.

Basic Health Checks

A basic health check allows monitoring a real server by performing a single type of check. A basic health check consists of the following parameters:

- Health check identification, including:
  - ID —A unique alphanumeric identifier
  - Name—A descriptive name
- Health check type —The type of application used for the check. See Supported Health Check Types, page 484 for the available health check types.
- Health check target
  - Destination address —Defines the IP address or hostname where this health check must be sent.

    When the destination address is unspecified (default) and the health check is assigned to a monitored element, the health check destination is selected as follows:

    - When assigned to a server group, separate run-time instances are created for each real server in the group, with the destination address set to real server IP.
    - When assigned to a real server, a run-time instance is created with the destination address set to real server IP.

    When a destination address is specified, the health check is always sent to that destination, regardless of its assigned elements. This option is useful to determine real server availability based on the availability of an external element (non-real server).

    If the destination address is specified as a hostname, the IP version with which you want the hostname to be resolved must be specified.
  - Destination port —Defines the application port where the health check must be sent.

    When the destination port is unspecified (default), the health check destination port is determined by the server port used for each monitored service. When the destination address is specified, the destination port must also be specified.

Note: The destination port parameter is not relevant for Link, ICMP, and ARP health checks.

- Reverse health check result—When this parameter is enabled, if the health check behaves as expected, it is considered failed.
- Health check timers
  - Interval (1-600 seconds)—Defines the interval at which consecutive health check requests are sent to the monitored element.
  - Timeout (0-600 seconds)—If the health check response from the monitored element does not arrive within this time frame, the health check fails. This parameter value must be lower or equal to the interval parameter. When parameter is set to 0, the timeout is equal to the interval.
  - Retries to failure (1-63)—The monitored element is considered unavailable if this number of consecutive health checks fails.

— Retries to restore (1-63)—The monitored element is considered available after failure if this number of consecutive health checks is successful.

— Down-time interval (0-600 sec)—This parameter allows defining a different health check interval (usually longer than regular interval) while the server is down. When the parameter is set to 0, the server is tested at the same interval whether it is up or down.

Note: Interval, retries to failure, and retries to restore parameters can be overridden at the real server level.

- Application arguments —Application related arguments that differ based on health check type. For details on the available health check types and their arguments, see Supported Health Check Types, page 484.

Advanced Server Health Checks

Alteon lets you determine real server availability based on multiple health checks. These checks can monitor different applications and different targets. For example, to determine whether application servers are available, you must test that the application is running on the server and back-end processing servers or databases are available.

Multiple basic health checks can be bound to the monitored real server by means of an advanced logical expression (LOGEXP) health check.

Supported Health Check Types

Alteon supports the following health check types:

- Link Health Checks, page 485
- TCP Health Checks, page 485
- UDP Health Checks, page 485
- ICMP Health Checks, page 486
- HTTP/S Health Checks, page 486
- TCP and UDP-based DNS Health Checks, page 488
- TFTP Health Check, page 488
- SNMP Health Check, page 488
- FTP Server Health Checks, page 489
- POP3 Server Health Checks, page 489
- SMTP Server Health Checks, page 490
- IMAP Server Health Checks, page 490
- NNTP Server Health Checks, page 490
- RADIUS Server Health Checks, page 490
- SSL HELLO Health Checks, page 491
- WAP Gateway Health Checks, page 491
- LDAP/LDAPS Health Checks, page 492
- Windows Terminal Server Health Checks, page 493
- ARP Health Checks, page 493
- DHCP Health Checks, page 493
- RTSP Health Checks, page 494

Link Health Checks

Link health checks are performed at the Layer 1 (physical) level, and are relevant only for Intrusion Detection Servers (IDS) servers. The intrusion detection interface on IDS servers does not include the TCP/IP stack, so it is not possible to perform any health check other than Layer 1.

The server is considered to be **up** when the link (connection) is present, and **down** when the link is absent.

To perform this health check, you need to:

- Connect each IDS server to a different physical port.
- Configure real servers for each IDS server, and assign a real server ID to the physical port on which it is connected. The real server ID is used to determine to which port the server is connected to.

  Note: In most cases, real sever numbering (rindex) and port numbering match up. For example, real server id 1 is assumed to be connected to port 1. When port/link 1 is up we declare real server 1 as up.

- Assign the pre-defined Link health check to the IDS server group.

For this health check type only the pre-defined **link** object is available. It is not possible to configure a user-defined Link health check.

TCP Health Checks

TCP health checks are useful in verifying that a specific TCP application port is up.

Session devices monitor the health of servers and applications by sending Layer 4 connection requests (TCP SYN packets). When a connection request succeeds, the session device quickly closes the connection by sending a TCP FIN packet. The pre-defined **tcp** health check is available for simple TCP service monitoring.

Note: The pre-defined **tcp** health check is the default health check for a new group.

UDP Health Checks

UDP health checks are useful in verifying that a specific UDP application port is up.

Due to the nature of UDP, UDP health checks use a combination of ICMP (ping) requests and UDP requests to monitor an UDP application port. When the UDP application is operational, no reply is received. When the UDP application is not operational, the ICMP message “UDP Port Unreachable” is sent. This means that it is impossible to know whether the lack of response is because the server is available, or because the host computer is not working and is unable to send a response of any kind.

To get a clear indication if the server is available, the UDP requests are combined with ping requests. A server is available when there is no response to the UDP request, but there is a response to the ping request. The pre-defined **udp** health check is available for simple TCP service monitoring.

ICMP Health Checks

The ICMP health check monitors real server availability by sending an ICMP echo request and waiting for an echo reply with the correct sequence number.

A pre-defined **icmp** health check is available. User-defined ICMP health checks are only necessary when you want to select non-default timer values or monitor a specific network element.

Note: The pre-defined **icmp** health check is the default health check for real servers that are not attached to any virtual service, and for UDP services when the health check of the attached group is for a TCP application.

HTTP/S Health Checks

The HTTP/S health check allows you to determine HTTP/S service availability by requesting a specified web page (GET or HEAD methods), or by posting a page (POST method). The health check is successful when an HTTP/S response is received and it matches one of the specified response codes and/or strings.

The following HTTP/S specific arguments facilitate the configuration of accurate health checks:

- HTTPS—Specifies whether to perform an HTTP (disabled) or HTTPS (enabled) health check.
- Host—Specifies the host header to be used in the health check request (up to 128 characters). If this parameter is not specified an HTTP 1.0 request is sent. Otherwise an HTTP 1.1 request is sent. An Inherit value can be configured to allow the host definition per virtual service using the virtual service **hname** parameter and virtual server **dname** parameter (**hname.dname**). See Example HTTP Health Checks, page 487.
- Path—Specifies the request path (up to 256 characters). If empty, the request is sent to the Web service root (“/”). An Inherit value can be configured to allow the path configuration using the group content. See Example HTTP Health Checks, page 487.
- Method—Specifies the HTTP method used in the request. The options are GET (default), POST, and HEAD.
- Additional headers—Specifies additional headers to be included in the health check HTTP request.
- Body—Specifies the HTTP body to be included in the health check HTTP request (up to 512 characters).
- Authentication—Specifies whether the monitored server requires authentication. The options are None, Basic (user and password), and NTLM (v2).
- User name and password—Specifies the login user name and password if authentication is required.
- Proxy request—Specifies whether to perform HTTP proxy health check. This means that the full path URI is included in the GET/POST command (even in HTTP 1.1 where the host appears in Host header).
- Response codes—Specifies a list of up to 10 response codes that represent health check success (or failure if a reverse check is performed). Default: 200
- Return String and Type—Specifies a string (up to 256 characters) expected in the response that represents health check success (or failure if a reverse check is performed) and its match type (included or regex).

Pre-defined **http** and **https** health checks are available for simple HTTP and HTTPS service monitoring. The health checks have the host and path parameters set to Inherit (their definition is taken from the virtual service and group configuration) and expect 200 OK response codes.

Example HTTP Health Checks

The following examples show the health checks sent when using HTTP health check configuration inherited from virtual service and group.

Note: If content is not specified, the health check is performed using the / character.

A Host header using virtual service (**hname**) and virtual server (**dname**) parameters

```
hname= everest
dname= example.com
content= index.html
```

Health check is performed using:

```
GET /index.html HTTP/1.1
Host: everest.example.com
```

B Host header using virtual server (**dname**) parameter only

```
hname= (none)
dname= raleighduram.cityguru.com
content= /page/gen/?_template=alteon
```

Health check is performed using:

```
GET /page/gen/?_template=alteon HTTP/1.1
Host: raleighduram.cityguru.com
```

C Host header not specified

```
hname= (none)
dname= (none)
content= index.html
```

Health check is performed using:

```
GET /index.html HTTP/1.0 (since no HTTP HOST: header is required)
```

D Request path using group content

```
hname= (none)
dname= (none)
content= //everest/index.html
```

Health check is performed using:

```
GET /index.html HTTP/1.1
Host: everest
```

TCP and UDP-based DNS Health Checks

Alteon supports both TCP and UDP-based DNS health checking. This health check is performed by sending a DNS query over either protocol and monitoring the server reply.

The following DNS-specific arguments are available:

- Protocol—Specifies whether the DNS health checks should be performed using UDP (default) or TCP protocol.
- Domain—Specifies the domain name that must be resolved (up to 63 characters). An Inherit value can be configured to allow definition of the domain using the group content parameter. If no domain name is configured, the health check is performed by sending a query for a dummy host and watching for the server's reply. The reply, even though it is negative (for example, the reply is “Server not found” since the query is for a dummy host), serves the purpose of a health check.

Pre-defined **udpdns** (DNS over UDP) and **dns** (UDP over TCP) health checks are available for simple DNS service monitoring. The domain parameter of the pre-defined health checks is set to Inherit, allowing definition using the group content and the destination port set to standard DNS port (53).

TFTP Health Check

Alteon supports the Trivial File Transfer Protocol (TFTP) health check, which uses the TFTP protocol to request a file from the server. At regular intervals, Alteon transmits TFTP read requests (RRQ) to all the servers in the group. The health check is successful if the server successfully responds to the RRQ. The health check fails if Alteon receives an error packet from the real server.

The following TFTP-specific argument is available:

- Path/Filename—Specifies the file name requested (up to 256 characters). Depending on the implementation of the TFTP daemon on the real servers being health-checked, you may have to specify the full pathname of the file (/tftpboot/<filename>) on some systems. On others, a filename is sufficient. By default, if no path is specified, the switch checks the /tftpboot folder. An Inherit value can be configured to allow the file configuration using the group content.

Note: If no filename is specified (directly or via group configuration), the health check performed for that group is TCP.

A pre-defined **tftp** health check is available for simple TFTP service monitoring. The health check has the path or filename parameter set to Inherit, allowing definition using the group content and the destination port set to standard TFTP port (69).

SNMP Health Check

Alteon supports SNMP health checks by sending an SNMP GET request to the real server running an SNMP-based agent. SNMP health checks can be used on any real servers, provided they have an SNMP agent. The SNMP health check is performed by polling a single variable within the MIB.

The health check is successful if a valid response is received and the returned value is within a configured range or if it matches a configured string.

The SNMP health check response can also be used to dynamically readjust monitored real server weights.

The following SNMP-specific arguments are available:

- OID—Specifies the OID whose value the health check attempts to retrieve.
- Community—Specifies the community name that the system must use to authenticate with the host server through SNMP.

- Minimum and maximum value—Specifies the minimum and/or maximum value that can be received as response that is considered a success. This should be used when the OID value is of numeric type (integer, counter, and so on)
- Response String—Specifies a string expected in the SNMP response value that represents health check success. This should be used when the OID value is of string type.

Note: If no expected response is configured (minimal or maximal value, or string), the health check just checks that an SNMP response is received.

- Readjust Weight—Specifies whether the real server weights should be dynamically adjusted based on SNMP health check response. If the parameter is enabled and the value in the response packet is greater than 100, then 100 is used as the weight (relevant only for an integer parameter).

If an OID and community string are assigned per IDS real server (**/cfg/slb/real/ids**), then they override the health check configuration.

FTP Server Health Checks

The Internet File Transfer Protocol (FTP) provides facilities for transferring files to and from remote computer systems. Usually the user transferring a file needs authority to log in and access files on the remote system. This protocol is documented in RFC 1123.

The FTP health check monitors an FTP service by attempting to login to the server and retrieve the specified file size.

The following FTP-specific arguments are available:

- Username—Specifies the login user name to the FTP server (up to 32 characters). Default: anonymous
- Password—Specifies the login password for the configured username (up to 32 characters).
- Path/Filename—Specifies the name of the file to be downloaded (up to 256 characters). An Inherit value can be configured to allow path/filename definition using the group content parameter. If no filename is specified, the FTP health check only checks successful login to the FTP server.

A pre-defined **ftp** health check is available for simple FTP service monitoring. The health check has the username set to Anonymous and the path/filename parameter set to Inherit, allowing definition using the group content and the destination port set to standard FTP control port (21).

POP3 Server Health Checks

The Post Office Protocol—Version 3 (POP3) is intended to permit a workstation to dynamically access a maildrop on a server host. The POP3 protocol is used to allow a workstation to retrieve mail that the server is holding for it. This protocol is documented in RFC 1939.

The POP3 health check monitors service availability by attempting login to the POP3 server and requires username and password configuration (mandatory parameters). An Inherit value can be configured for the two parameters to allow the user name and password configuration using the group content (content includes user:password).

Note: If the username and password are set to Inherit but group content is empty, the health check performed for that group is TCP.

A pre-defined **pop3** health check is available for simple POP3 service monitoring. The health check has the username and password parameters set to Inherit, allowing definition using the group content and the destination port set to standard POP3 port (110).

SMTP Server Health Checks

Simple Mail Transfer Protocol (SMTP) is a protocol to transfer e-mail messages between servers reliably and efficiently. This protocol traditionally operates over TCP, port 25 and is documented in RFC 821.

The SMTP health check attempts to verify specified user name validity on the SMTP server. The username configuration is mandatory. An Inherit value can be configured to allow the user name configuration via group content.

A pre-defined **smtp** health check is available for simple SMTP service monitoring. The health check has the Username parameter set to Inherit, allowing definition using the group content and the destination port set to standard SMTP port (25).

IMAP Server Health Checks

Internet Message Access Protocol (IMAP) is a mail server protocol used between a client system and a mail server that allows a user to retrieve and manipulate mail messages.

The IMAP health check monitors service availability by attempting login to the IMAP server and requires username and password configuration (mandatory parameters). An Inherit value can be configured for the two parameters to allow the user name and password configuration using the group content (content includes user:password).

Note: If the username and password are set to Inherit but the group content is empty, the health check performed for that group is TCP.

A pre-defined **imap** health check is available for simple IMAP service monitoring. The health check has the Username and Password parameters set to Inherit, allowing definition using the group content and the destination port set to standard IMAP port (143).

NNTP Server Health Checks

Net News Transfer Protocol (NNTP) specifies a protocol for the distribution, inquiry, retrieval, and posting of news articles using a reliable stream-based transmission of news among the ARPA-Internet community. NNTP is designed so that news articles are stored in a central database allowing a subscriber to select only those items he wishes to read.

NNTP is documented in RFC 977. Articles are transmitted in the form specified by RFC 1036.

NNTP health check monitors service availability by attempting to retrieve the identification line of the specified Newsgroup Name (mandatory parameter) from the server. An Inherit value can be configured to allow the newsgroup name configuration using the group content.

Note: If the Newsgroup Name is set to Inherit but the group content is empty, the health check performed for that group is TCP.

A pre-defined **nntp** health check is available for simple NNTP service monitoring. The health check has the newsgroup name parameter set to Inherit, allowing definition using the group content and the destination port set to standard NNTP port (119).

RADIUS Server Health Checks

Alteon lets you use the Remote Authentication Dial-In User Service (RADIUS) protocol to health check the RADIUS accounting and authentication services on RADIUS servers. RADIUS is stateless and uses UDP as its transport protocol.

The following RADIUS-specific arguments are available:

- Type—Specify the type of the RADIUS service that must be monitored. Options are Accounting and Authentication.
- Username and password—Specifies the username and password parameters that are mandatory for RADIUS Authentication health check. An Inherit value can be configured for the two parameters to allow the user name and password configuration using the group content (content includes user:password).
- Shared secret—Specifies the shared secret parameter that is mandatory for a RADIUS Authentication health check. An Inherit value can be configured to allow the parameter configuration using the group secret value or advanced health check secret value if the group secret is empty.

Note: For a RADIUS Authentication health check if the username, password and secret are unspecified (directly or using the group configuration), the health check performed for that group is TCP.

The following pre-defined RADIUS health checks are available:

- radius-auth—RADIUS Authentication health check with username, password and shared secret set to Inherit.
- radius-acc—RADIUS Accounting health check with username, password empty and shared secret set to Inherit.
- radius-aa—Performs either a RADIUS Accounting or a RADIUS Authentication health check based on the server port (rport) configured on the service. If the server port is not a standard RADIUS port (1812 or 1813), a TCP health check is performed. For this health check, the username, password and shared secret are set to Inherit.

SSL HELLO Health Checks

Alteon can query the health of the SSL servers by sending an SSL client “Hello” packet and then verifying that the response is a valid Server Hello response.

During the handshake, the user and server exchange security certificates to negotiate an encryption and compression method and establish a session ID for each session.

You perform the health check using either SSL version 2 or SSL version 3.

Two pre-defined SSL Hello health checks are available:

- **sslh**—SSL Hello version 2 health check.
- **sslhv3**—SSL Hello version 3 health check.

WAP Gateway Health Checks

The Wireless Application Protocol (WAP) is a specification for wireless devices that uses TCP/IP and HTTP as part of a standards-based implementation. The translation from HTTP/HTML to WAP/WML (Wireless Markup Language) is implemented by servers known as WAP gateways on the land-based part of the network.

Wireless Session Protocol (WSP) is used within the WAP suite to manage sessions between wireless devices and WAP content servers or WAP gateways. Alteon includes a content-based health check mechanism where customized WSP packets are sent to the WAP gateways, and Alteon verifies the expected response.

Alteon supports WAP health checks for all 4 transport types: secure and non-secure connection-less transport, secure and non-secure connection-oriented transport, as detailed in Table 45:

Table 45: WAP Gateway Health Checks

WAP Health Check Type	Description	Default Port	Arguments
WSP	Connection-less WSP	9200	See below.
WTP[1]	Connection-oriented WTP + WSP	9201	See below.
WTLS[2] WSP	Encrypted connection-less WTLS + WSP	9202	No parameters required.
WTLS WTP	Encrypted connection-oriented WTLS + WTP + WSP	9203	No parameters required.

1 – Wireless Transaction Protocol
2 – Wireless Transport Layer Security

Note: In Alteon, all four WAP services are grouped together. If a health check to one of the services fail on a specific real server, then all four WAP services (9200, 9201, 9202, or 9203) are disabled on that real server.

The following WAP-specific arguments are available for WSP and WTP health check types:

- Connect message header (mandatory)—Specifies the content for the Connect message used for unencrypted WTP health check only.
- Sent content (mandatory)—Specifies the content of the packet that is sent to the gateway as a hexadecimal string, which should include all applicable WSP headers.
- Received content (mandatory)—Specifies the expected response for WTP health checks as a hexadecimal byte string. Alteon matches each byte of this string with the received content and if there is a mismatch of even a single byte on the received content, the health check fails.
- Offset—Specifies the offset from which to start search for the Received Content in the UDP data.
- RADIUS Service Dependency—Specifies whether RADIUS accounting service must also be monitored on the WAP servers. When this parameter is enabled, if the RADIUS service is unavailable, the server is unavailable.

Note: For unencrypted WSP and WTP WAP health checks, if the mandatory content arguments are empty, the health check performed for that group is TCP.

The following WAP pre-defined health checks are available:

- **wsp**, **wtp**, **wtls-wsp** and **wtls-wtp**—Unlike other pre-defined health checks available on Alteon, these health checks are editable. For WSP and WTP health checks, if the content parameters are not configured, the health check performed is TCP.
- **wtls**—Performs either WTLS+WSP or WTLS+WTP depending on virtual service port.

LDAP/LDAPS Health Checks

The Lightweight Directory Access Protocol (LDAP) health checks enable Alteon to determine whether the LDAP server is alive or not. LDAP versions 2 and 3 are described in RFC 1777 and RFC 2251.

The LDAP health check attempts to initiate an LDAP application session with the monitored server by sending a BIND request. If a BIND response is received from the server and the result code indicates that the server is alive, the health check is successful. After the BIND response is received, Alteon sends an UNBIND request so that the server can close the LDAP application session.

The following LDAP/S-specific arguments are available:

- LDAPS—Specifies whether to perform a LDAP (disabled) or LDAPS (enabled) health check.
- Username and Password—Specifies the login user name and password when authentication is required.
- Base Distinguish Name—Specifies the domain component of the root Distinguish Name.

You can configure an Inherit value for Username, Password, and Base Distinguish Name arguments to allow configuration using the group content. The group content includes all required parameters in the LDAP message format:
cn=<username>,dc=<base-part1>,dc=<base-part2>,dc=<base-part3>:<password>.

Note: If the Username, Password and Base Distinguish Name are set to Inherit, and the group content is empty, the health check performed for that group is TCP.

Pre-defined **ldap** and **ldaps** health checks are available for simple LDAP and LDAPS service monitoring. The health checks have all the parameters set to Inherit, allowing definition using the group content.

The Alteon LDAP health check is supported for LDAP version 2 and 3. The LDAP version used is defined per Alteon by the global flag **`cfg/slb/advhc/ldapver`**.

Windows Terminal Server Health Checks

Windows Terminal Services (WTS), renamed Remote Desktop Services in Windows 2008 R2, is a component of Microsoft Windows (both server and client versions) that allows a user to access applications and data on a remote computer over a network, using the Remote Desktop Protocol (RDP).

The WTS health check attempts to open a connection to the TS server. You can define a user name to be used in the TS cookie. By default, the user name *Administrator* is used. An Inherit value can be configured to allow the user name configuration via group content.

A pre-defined ***wts*** health check is available for simple WTS service monitoring. The health check has the Username parameter set to Inherit, allowing configuration using the group content. The destination port is set to the standard WTS port (3389).

ARP Health Checks

The Address Resolution Protocol (ARP) is the TCP/IP protocol that resides within the Internet layer. ARP resolves a physical address from an IP address. ARP queries computers on the local network for their physical addresses. ARP also maintains IP-to-physical address pairs in its cache memory. In any IP communication, the ARP cache is consulted to see if the IP address of the computer or the router is present in the ARP cache. The corresponding physical address is used to send a packet.

A pre-defined **arp** health check is available.

DHCP Health Checks

The DHCP health check monitors the service by sending a DHCP INFORM or REQUEST message and expecting responses.

You can specify the following:

- DHCP message type—Send an INFORM or REQUEST message
- DHCP message source port—Use a random or standard port (68 for IPv4 and 546 for IPv6)

An Inherit value can be configured for both parameters to allow configuration using the group content.

The group content supports the following options:

- **request**—DHCP REQUEST with a random source port
- **srequest**—DHCP REQUEST with source port 68 for an IPv4 target or 546 for an IPv6 target
- **strict**—DHCP INFORM with source port 68 for an IPv4 target or 546 for an IPv6 target
- **none**—DHCP INFORM with a random source port

A pre-defined **dhcp** health check is available for simple DHCP service monitoring. The health check has the parameters set to Inherit, allowing definition using the group content.

Note: Enable DAM while using this health check type.

RTSP Health Checks

The RTSP health check monitors the service by sending OPTIONS or DESCRIBE requests. The health check is successful if an RTSP response with the expected response code is received.

The following RTSP-specific arguments are available:

- Method—Specifies whether to use the OPTIONS or DESCRIBE RTSP method in the request. An Inherit value can be configured to allow the method to be based on the group content. If the content is configured, the DESCRIBE method is used with the Hostname and Path configured in the content. Otherwise the OPTIONS method is used.
- Hostname and Path—Specifies the host name and path to be used in the DESCRIBE health check request. An Inherit value can be configured to allow the host definition using the group content.
- Response codes—Specifies a list of up to 10 response codes that represent health check success. The default is 200.

A pre-defined **rtsp** health check is available for simple RTSP service monitoring. The health check has the parameters set to Inherit, allowing definition using the group content and destination port set to standard RTSP port (554).

SIP Health Checks

The Session Initiation Protocol (SIP) is an application-level control (signaling) protocol for Internet multimedia conferencing, telephony, event notification, and instant messaging. The protocol initiates call setup, routing, authentication and other feature messages to end-points within an IP domain.

Alteon can monitor SIP service using standard SIP OPTIONS health check or Nortel proprietary SIP Ping.

The following SIP-specific arguments are available:

- Method—Specifies the SIP method used by the health check (OPTIONS or PING).
- Request URI—Specifies the URI (without the sip: part) used in the health check request. If this parameter is not specified, the health check URI is RIP:rport.
- From—Specifies the content of the From and Contact headers. An Inherit value can be configured to allow configuration using the group content. If this parameter is empty, radware@radware.com is used.
- Response codes—Specifies a list of up to 10 response codes that represent health check success.

Pre-defined SIP (SIP Ping) and SIPOPTIONS (SIP OPTIONS) health checks are available for simple SIP service monitoring. For these health checks, the Request URI is set to Inherit and the expected response code is 200.

Script-Based Health Checks

Health check scripts dynamically verify application and content availability by executing a sequence of tests based on send and expect commands.

This section includes the following topics:

- Configuring Script-Based Health Checks, page 495
- Script Formats, page 495
- Scripting Commands, page 497
- Scripting Guidelines, page 498
- Script Configuration Examples, page 498

Configuring Script-Based Health Checks

You can configure Alteon to send a series of health check requests to real servers or real server groups and monitor the responses. Both ASCII and binary-based scripts, for TCP and UDP protocols, can be used to verify application and content availability.

The benefits of using script-based health checks include:

- Ability to send multiple commands
- Check for any return ASCII string or binary pattern
- Test availability of different applications
- Test availability of multiple domains or Web sites

Alteon supports the following capacity:

- 6K bytes per script
- 256 scripts per Alteon

A simple CLI controls the addition and deletion of commands to each script. New commands are added and removed from the end of the script. Commands exist to open a connection to a specific TCP or UDP port, send a request to the server, expect an ASCII string or binary pattern, and, for TCP-based health checks only, to close a connection. The string or pattern configured with an **expect** (or in the case of binary, **bexpect**) command is searched for in each response packet. If it is not seen anywhere in any response packet before the real server health check interval expires, the server does not pass the expect (or bexpect) step and fails the health check. A script can contain any number of these commands, up to the allowable number of characters that a script supports.

Notes

- Health check scripts can only be set up via the CLI, but once entered, can be assigned as the health-check method using SNMP or the BBI.
- There is no need to use double slashes when configuring a script in the BBI that uses special characters with single slashes. For example, the script entry **GET /index.html HTTP/1.1\r\nHOST:www.hostname.com\r\n\r\n** does not require the use of **\\r** or **\\n** to ensure proper functioning of the script.
- Only one protocol can be configured per script.

Script Formats

Health check script formats use different commands based on whether the content to be sent is ASCII-based or binary-based. The **close** command is used only to close a TCP connection and is not required if health checking a UDP-based protocol.

Each script should start with the command **open <protocol port number>, <protocol-name>**. The next line can be either a **send** or **expect** (for ASCII-based), or **bsend** or **bexpect** (binary-based).

ASCII-Based Health Check

The following is the general format for ASCII-based health-check:

```
open application_port, protocol-name #(for example: 80, TCP)
send request 1 (ascii string)
expect response 1
send request 2
expect response 2
send request 3
expect response 3
close #(used in TCP-based health checks only)
```

Binary-Based Health Check

The following is the general format for binary-based health check scripts. Specify the binary content in hexadecimal format:

```
open application_port, protocol-name #(for example: 80, TCP)
bsend request 1 (binary pattern in hex format)
nsend request 1-continued
bexpect response 1
nexpect response 1-continued
expect response 3
offset offset count
depth number of packets from offset to count
close #(used in TCP-based health checks only)
```

A Binary-Based TCP Health Check

The **bsend** and **bexpect** commands are used to specify binary content. The **offset** and **depth** commands are used to specify where in the packet to start looking for the binary content. For example, if your script is configured to look for an HTTP 200 (ok) response, this typically appears starting from the 7th byte in the packet, so an offset value of 7 can be specified:

```
open "80,tcp"
bsend " <binary content for request 1> "
nsend " <continuing binary content for request 1> "
bexpect " <binary content for response 1> "
nexpect " <binary content> "
offset " <byte count from the start of the IP header> "
depth "10"
wait "100"
close #(used in TCP-based health checks only)
```

Notes **UDP-Based Health Checks**

- UDP-based health check scripts can use either ASCII strings or binary patterns.
- The **close** command is not required for a health check on UDP protocol.

Notes **TCP-based Health Checks for HTTP Protocol**

- If you are performing HTTP 1.1 pipelining, you need to individually open and close each response in the script.

- For HTTP-based health checks, the first word is the method. The method is usually the **get** command. However, HTTP also supports several other commands, including **put** and **head**. The second word indicates the content desired, or request-URI, and the third word represents the version of the protocol used by the client.
- If you supplied HTTP/1.1 for the protocol version, you would also have to add in the following line: **Host: www.hostname.com**

Example

`GET /index.html HTTP/1.1`	(press the **Enter** key)
`Host: www.hostname.com`	(press the **Enter** key twice)

This is known as a host header. It is important to include because most Web sites now require it for proper processing. Host headers were optional in HTTP/1.0 but are required when you use HTTP/1.1+.

- In order to tell the application server you have finished entering header information, a blank line of input is needed after all headers. At this point, the URL will be processed and the results returned to you.

  Note: If you make an error, enter **rem** to remove the last typed script line entered. If you need to remove more than one line, enter **rem** for each line that needs to be removed.

- Alteon includes the "\" prompt, which is one **Enter** key stroke. When using the **send** command, note what happens when you type the send command with the command string. When you type **send**, press the **Enter** key and allow Alteon to format the command string (that is, \ versus \\).

Scripting Commands

Listed below are the currently available commands for building a script-based health check:

- **OPEN**—Specify which destination real server UDP port to be used. For example: OPEN 9201. You can also use Inherit to allow a script to inherit the destination port from the service server port. This enables the reuse of a script for multiple services. After entering the destination port, you will be prompted to specify a protocol. Choose **udp**.
- **CLOSE** (for TCP-based health checks only)—Close a TCP connection. It is not necessary to use this command for UDP services.
- **SEND**—Specify the send content in raw hexadecimal format.
- **BSEND** (for binary content only)—Specify binary content (in hexadecimal format) for the request packet.
- **NSEND** (for binary content only)—Specify an additional binary send value (in hexadecimal format) at the end of a UDP based health check script. The NSEND command lets the user append additional content to the packet generated by the BSEND command. Since the current CLI limit allows a maximum of 256 bytes to be entered, using one or more NSEND commands will allow binary content of more than 256 bytes in length to be generated.
- **EXPECT**—Specify the expected content in raw hexadecimal format.
- **BEXPECT** (for binary content only)—Specify the binary content (in hexadecimal format) to be expected from the server response packet.
- **NEXPECT** (for binary content only)—Similar to NSEND, specify additional binary content to be appended to the original content specified by the BEXPECT command.

- **OFFSET** (for binary content only)—Specify the offset from the beginning of the binary data area to start matching the content specified in the EXPECT command. The OFFSET command is supported for both UDP and TCP-based health checks. Specify the OFFSET command after an EXPECT command if an offset is desired. If this command is not present, an offset of zero is assumed.
- **DEPTH** (for binary content only)—Specify the number of bytes in the IP packet that should be examined. If no OFFSET value is specified, DEPTH is specified from the beginning of the packet. If an OFFSET value is specified, the DEPTH counts the number of bytes starting from the offset value.
- **WAIT**—Specify a wait interval before the expected response is returned. The wait window begins when the SEND string is sent from Alteon. If the expected response is received within the window, the WAIT step passes. Otherwise, the health check fails. The WAIT command should come after an EXPECT command in the script, or the OFFSET command if one exists after an EXPECT command. The wait window is in units of milliseconds.
- **Wildcard character (*)**—Trigger a match as long as a response is received from the server. The wildcard character is allowed with the BEXPECT command, as in BEXPECT *****. Any NEXPECT, OFFSET, or DEPTH commands that follow a wildcard character will be ignored.

Scripting Guidelines

- Use generic result codes that are standard and defined by the RFC, as applicable. This helps ensure that if the server software changes, the servers do not start failing unexpectedly.
- Avoid tasks that may take a long time to perform or the health check will fail. For example, avoid tasks that exceed the interval for load balancing.

Script Configuration Examples

This section includes the following script configuration examples:

- Example 1: A Basic ASCII TCP-Based Health Check, page 498
- Example 2: GSLB URL Health Check, page 499
- Example 3: A UDP-Based Health Check using Binary Content, page 500
- Example 4: A TCP-Based Health Check using Binary Content, page 501

Example 1: A Basic ASCII TCP-Based Health Check

Configure Alteon to check a series of Web pages (HTML or dynamic CGI scripts) before it declares a real server is available to receive requests.

```
>> /cfg/slb/group x/health script1/content none
>> /cfg/slb/advhc/script 1

open 80
send “GET /index.html HTTP/1.1\\r\\nHOST:www.hostname.com\\r\\n\\r\\n
expect "HTTP/1.1 200
close
open 80
send "GET /index.html HTTP/1.1\\r\\nHOST:www.hostname.com\\r\\n\\r\\n
expect "HTTP/1.1 200
close
open 443
...
close
```

Notes

- If you are using the CLI to create a health check script, you must use quotes (") to indicate the beginning and end of each command string.
- When you are using the CLI to enter the send string as an argument to the send command, you must type two back slahes (“\”) before an “n” or “r”. If you are instead prompted for the line, that is, the text string is entered after pressing the **Return** key, then only one “\” is needed before the “n” or “r”.

Example 2: GSLB URL Health Check

Before the introduction of the scriptable health check feature in Alteon, each remote Global Server Load Balancing (GSLB) site's virtual server IP address was required to be a real server of the local Alteon. Each Alteon sent a health check request to the other virtual servers that were configured on the local device. The health check was successful if there was at least one real server on the remote device that was up. If all real servers on the remote device were down, the remote real server (a virtual server of a remote device) responded with an HTTP redirect message to the health check.

Using the scriptable health check feature, you can set up health check statements to check all the substrings involved in all the real servers.

The following is an example GSLB URL health check configuration:

- Site 1 with Virtual Server 1 and the following real servers:
  - Real Server 1 and Real Server 2: “images”
  - Real Server 3 and Real Server 4: “html”
  - Real Server 5 and Real Server 6: “cgi” and “bin”
  - Real Server 7 (which is Virtual Server 2): “any”
- Site 2 with Virtual Server 2 and the following real servers:
  - Real Server 1 and Real Server 2: “images”
  - Real Server 3 and Real Server 4: “html”
  - Real Server 5 and Real Server 6: “cgi” and “bin”
  - Real Server 7 (which is Virtual Server 2): “any”

Script-based health checking only sends the appropriate requests to the relevant servers. Using the script as shown in Figure 73 - Example Health Checking Script, page 500, the first GET statement is only be sent to Real Server 1 and Real Server 2. Going through the health check statements serially ensures that all content is available by at least one real server on the remote site.

The remote real server IP address (the virtual server IP address of the remote site) accepts “any” URL requests. The purpose of the first GET in the script is to check if Real Server 1 or Real Server 2 is up. In other words, it checks if the remote site has at least one server for “images” content. Either Real Server 1 or Real Server 2 responds to the first GET health check.

If all the real server IP addresses are down, Real Server 7 (the virtual server IP address of the remote site) responds with an HTTP redirect (respond code 302) to the health check. As a result, the health check fails, as the expected response code is 200, ensuring that the HTTP redirect messages will not cause a loop.

Figure 73: Example Health Checking Script

```
>>/cfg/slb/group x/health script2/content none
>> /cfg/slb/advhc/script 2

open 80
send "GET /images/default.asp HTTP/1.1\\r\\nHOST: 192.192.1.2\\r\\n\\r\\n"
expect "HTTP/1.1 200"
close

open 80
send "GET /install/default.html HTTP/1.1\\r\\nHOST: 192.192.1.2\\r\\n\\r\\n"
expect "HTTP/1.1 200"
close

open 80
send "GET /script.cgi HTTP/1.1\\r\\nHOST: www.myurl.com \\r\\n\\r\\n"
expect "HTTP/1.1 200"
close
```

Example 3: A UDP-Based Health Check using Binary Content

Health check scripts can be designed to be sent over the UDP protocol with a few minor differences from a TCP-based health check script. Due to the stateless nature of the UDP protocol, the CLOSE command is not supported.

The following is an example UDP-based script that uses binary content to health check the UDP port on a real server:

```
>> /cfg/slb/group <x> /health script3/content none
>> /cfg/slb/advhc/script 3
open "53,udp"
bsend "53 53 01 00 00 01 00 00"
nsend "00 00 00 00 03 77 77 77"
nsend "04 74 65 73 74 03 63 6f"
nsend "6d 00 00 01 00 01"
bexpect "00 01 00 01"
offset "1"
depth "32"
wait "1024"
```

Note: A maximum of 255 bytes of input are allowed in the CLI. If you send or expect lengthy content, you may want to remove spaces in between the numbers to save space on the CLI. For example, type **000101** instead of **00 01 01**. Alternately, continue the content using the nsend and nexpect commands.

Example 4: A TCP-Based Health Check using Binary Content

Health check scripts can be sent over the TCP protocol using binary content.

The following is an example of a TCP-based script that uses binary content to send an HTTP GET request and expect an HTTP 200 response:

```
>> /cfg/slb/group <x> /health script 4/content none
>> /cfg/slb/advhc/script4
open "80,tcp"
bsend "474554202F746573742E68746D20"
nsend "485454502F312E300D0A0D0A"
bexpect "203230"
nexpect "3020"
offset "7"
depth "10"
wait "100"
close
```

Verifying Script-Based Health Checks

If a script fails, the expect line in the script that is failing is displayed using the **/info/slb/real <real server number>** command:

```
>># /info/slb/real 1
  1: 205.178.13.225, 00:00:00:00:00:00, vlan 1, port 0, health 4, FAILED
    real ports:
    script 2, DOWN, current
      send GET / HTTP/1.0\r\n\r\n
      expect HTTP/1.0 200
```

In this case, the server is not responding to the get with the expect string.

When the script succeeds in determining the health of a real server, the following information displays:

```
>> # /info/slb/real 1
  1: 205.178.13.223, 00:00:5e:00:01:24, vlan 1, port 2, health 4, up
     real ports:
     script 2, up, current
```

Pre-defined Health Check Summary

The following table details all available out-of-the-box health check objects:

Name	Description
link	Verifies the status of the interface using the monitored element to which it is connected. This type of health check is relevant only for monitoring IDS servers.
arp	Monitors server availability using ARP requests.
icmp	Checks connectivity to the monitored element using ICMP.
tcp	Monitors a TCP service by sending simple TCP requests to the server port (rport) of a virtual service.
udp	Monitors a UDP service by sending a combination of ICMP requests and simple UDP requests to the server port (rport) of a virtual service.
http/https	Sends an HTTP or HTTPS request to the Web page defined in the virtual service (hname and dname) and group (content) and expects a 200 response code.
dhcp	Sends a DHCP request determined by the health check content configuration in the monitored group.
dns	Sends a DNS query for domain name configured in the group health check content to standard TCP DNS port (53).
udpdns	Sends a DNS query for domain name configured in the group health check content to standard UDP DNS port (53).
ftp	Attempts an anonymous login to the FTP server and retrieval of the filename configured in the group health check content.
imap	Attempts to login to the IMAP server on the standard port (143) using the user and password configured in the group health check content.
ldap/ldapss	Attempts to login into an LDAP or LDAPS server and retrieve data using the parameters configured in the group health check content.
nntp	Attempts to access the NNTP server on the standard port (119) and retrieve the identification line of the newsgroup configured in the group health check content.
pop3	Attempts to login to the POP3 server on the standard port (110) using the user and password configured in the group health check content.
radius-auth	Sends RADIUS authentication request using the parameters values configured in the group health check content and secret.
radius-aa	Sends a RADIUS accounting request.

Name	Description
radius-any	Sends either a RADIUS authentication or a RADIUS accounting request, depending on the service port. The service port must be the standard port for either RADIUS Authentication or Accounting.
rtsp	Connects to the RTSP server on the standard 554 port and sends an RTSP request determined by the group health check content value.
sip	Sends an SIP ping (proprietary Nortel) request to the real server.
sipoptions	Sends an SIP OPTIONS request to the real server.
smtp	Attempts to access the SMTP server on the standard port 25 and verify the validity of the username configured in the group health check content.
sslh	Sends an SSL Hello version 2 to the real server.
sslh3	Sends an SSL Hello version 3 to the real server.
tftp	Attempts to connect to the TFTP server on the standard port 69 and download the file specified in the group health check content using TFTP.
wsp	Monitors unencrypted connection-less WAP service availability, optionally in conjunction with the RADIUS service. **Note:** This health check is editable.
wtls-wsp	Monitors encrypted connection-less WAP service availability, optionally in conjunction with the RADIUS service. **Note:** This health check is editable.
wtls-wtp	Monitors encrypted connection-oriented WAP service availability, optionally in conjunction with the RADIUS service. **Note:** This health check is editable.
wtls	Monitors encrypted connection-less or connection-oriented WAP service availability, depending on the server port of the virtual service. If the service port is not standard secure WSP or WTP port (9202 or 9203), a TCP health check is performed.
wts	Monitors WTS (Window Terminal Server) service availability.

Failure Types

This section describes the following failure types:

- Service Failure, page 503
- Server Failure, page 504

Service Failure

If a certain number of connection requests for a particular service fail, Alteon puts the service into the **service failed** state. While in this state, no new connection requests are sent to the server for this service. However, if graceful real server failure is enabled (using **`/cfg/slb/adv/grace ena`**), state information about existing sessions is maintained and traffic associated with existing sessions continues to be sent to the server. Connection requests to, and traffic associated with, other load-balanced services continue to be processed by the server.

Example

A real server is configured to support HTTP and FTP within two real server groups. If a session device detects an HTTP service failure on the real server, it removes that real server group from the load-balancing algorithm for HTTP, but keeps the real server in the mix for FTP. Removing only the failed service from load balancing allows users access to all healthy servers supporting a given service.

When a service on a server is in the **service failed** state, the Alteon sends Layer 4 connection requests for the failed service to the server. When Alteon has successfully established a connection to the failed service, the service is restored to the load-balancing algorithm.

Server Failure

If all load-balanced services supported on a server fail to respond to connection requests within the specified number of attempts, then the server is placed in the **server failed** state. While in this state, no new connection requests are sent to the server. However, if graceful real server failure is enabled (using **`/cfg/slb/adv/grace ena`**), state information about existing sessions is maintained and traffic associated with existing sessions continues to be sent to the server.

All load-balanced services on a server must fail before Alteon places the server in the **server failed** state.

The server is brought back into service as soon as the first service is proven to be healthy. Additional services are brought online as they are subsequently proven to be healthy.

Preventing a Flood of Server Connections

Alteon performs a slow start on the real server which comes back after the health check fails to prevent flood of connections. This occurs for groups configured with leastconns as the metric. When the real server comes up, it contains zero connections and all the new connections are directed to it. This heavy connection flow brings down the real server.

To prevent this flood, Alteon temporarily changes the group metric to round-robin for one minute and reverts back to leastconns metric, assuming the next real server which comes up is ready to accept new connections.

By default the slow start feature is disabled.

To check the slow start mode of a real server in a group

Enter the following command:

```
>> Main# /info/slb
```

The following information displays:

```
>> Server Load Balancing Information# virt 1
  1: 10.10.7.1, 00:01:81:2e:a0:8e
    virtual ports:
    http: rport http, group 1, backup none, slowstart real servers:
            1: 192.168.2.11, backup none, 0 ms, group ena, up, dynamic weight 20
            2: 192.168.2.12, backup none, 0 ms, group ena, up, dynamic weight 40
```

To enable the slow start mode of a real server in a group

Enter the following command:

```
>> Main# /cfg/slb/group <group_number>/slowstr <value_in_seconds>
```

DSR Health Checks

Direct Server Return (DSR) health checks are used to verify the existence of a server-provided service where the server replies directly back to the client without responding through the virtual server IP address. In this configuration, the server will be configured with a real server IP address and virtual server IP address. The virtual server IP address is configured to be the same address as the user's virtual server IP address. When DSR health checks are selected, the specified health check is sent originating from one of Alteon's configured IP interfaces, and is destined to the virtual server IP address with the MAC address that was acquired from the real server IP address's Address Resolution Protocol (ARP) entry.

Alteon lets you perform health checks for DSR configurations. For more information, see Direct Server Return, page 197. Alteon can verify that the server correctly responds to requests made to the virtual server IP address as required in DSR configurations. To perform this function, the real server IP address is replaced with the virtual server IP address in the health check packets that are forwarded to the real servers for health checking. With this feature enabled, the health check will fail if the real server is not properly configured with the virtual server IP address.

Note: The DSR VIP health check (**cfg/slb/group/viphlth**) is enabled by default. This has no effect on the health check unless the real server is configured with DSR.

Advanced Group Health Check

Alteon lets you configure an expression to fine-tune the selected health check for a real server group. For example, you have configured a real server group with four real servers. Two of the real servers handle the contents of the Web site and the other two real servers handle audio files. If the two content servers fail due to traffic distribution, then you want the two audio servers to fail. However, you want the audio servers up if at least one of the content servers is up.

The advanced group health check feature lets you create a boolean expression to health check the real server group based on the state of the virtual services. This feature supports two boolean operators: AND and OR. The two boolean operators are used to manipulate TRUE/FALSE values as follows:

- **OR operator (|)**—A boolean operator that returns a value of TRUE if either or both of its operands are TRUE. This is called an inclusive OR operator.
- **AND operator (&)**—A boolean operator that returns a value of TRUE if both of its operands are TRUE.

Using parenthesis with the boolean operators, you can create a boolean expression to state the health of the server group. The following two boolean expressions show two examples with real servers 1, 2, 3, and 4 in two different groups:

Examples

A (1|2)&(3|4)

Real servers 1, 2, 3, and 4 are configured in group 1 and assigned to virtual service *x* in Virtual Server 1. The boolean expression is used to calculate the status of a virtual service using group 1 based on the status of the real servers.

Virtual service *x* of Virtual Server 1 is marked UP if Real Servers 1 or 2 and Real Servers 3 or 4 are health checked successfully.

`>> # /cfg/slb/group 1`	(Select the Real Server Group 1)
`>> Real server group 1# advhlth (1|2)&(3|4)`	(Configure a boolean expression for health check)
`>> # /cfg/slb/virt 1/service x/group 1`	(Assign the Real Server Group 1)
`>> Virtual Server 1 Service# apply`	(Apply the changes)
`>> Virtual Server 1 Service# save`	(Save the changes)

B (1&2)|(2&3)|(1&3)

Real servers 1, 2, and 3 are configured in Group 2 and assigned to virtual service *x* in Virtual Server 1. The boolean expression is used to calculate the status of the virtual service using Group 2 based on the status of the real servers.

Virtual service *x* of Virtual Server 1 is marked UP only if at least two of the real servers are health checked successfully.

`>> # /cfg/slb/group 2`	(Select the Real Server Group 2)
`>> Real server group 2# advhlth (1&2)|(2&3)|(1&3)`	
	(Configure a boolean expression for health check)
`>> # /cfg/slb/virt 1/service x/group 2`	(Assign the Real Server Group 2)
`>> Virtual Server 1 Service# apply`	(Apply the changes)
`>> Virtual Server 1 Service# save`	(Save the changes)

Disabling the Fast Link Health Check

By default, Alteon sets the real server as operationally down as soon as the physical connection to it is down, without waiting for the health check to fail. This behavior may not be advantageous in certain configurations in which a link may go down and then be quickly restored, such as in VPN load balancing. By disabling this "fast health check" behavior, the real server will be marked as **down** only after the configured health check interval, thus allowing the possibility of the server re-establishing itself before the next health check.

To enable or disable fast link health checks

```
>> # /cfg/slb/real  <real-server-#> /adv/fasthc ena|dis
```

Chapter 19 – High Availability

Alteon supports high availability network topologies through an enhanced implementation of the Virtual Router Redundancy Protocol (VRRP).

This chapter describes the following topics:

Virtual Router Redundancy Protocol

This section describes the following Virtual Router Redundancy Protocol (VRRP)-related topics:

VRRP Overview

VRRP eliminates single points of failure within a network. The protocol supports redundant router configurations within a LAN, providing alternate router paths for a host.

In a high availability network topology, no device should be a single point of failure for the network or cause a single point of failure in any other part of the network. This means that a network remains in service despite the failure of any single device. To achieve this usually requires redundancy for all vital network components.

Each participating VRRP-capable routing device is configured with the same virtual router IP address and ID number. One of the virtual routers is elected as the master, based on a number of priority criteria, and assumes control of the shared virtual router IP address. If the master fails, one of the backup virtual routers takes control of the virtual router IP address and actively processes traffic addressed to it.

Because the router associated with a given alternate path supported by VRRP uses the same IP address and MAC address as the routers for other paths, the host's gateway information does not change, no matter which path is used. A VRRP-based redundancy schema reduces administrative overhead because hosts do not need to be configured with multiple default gateways.

Note: The IP address of a VRRP virtual interface router (VIR) and virtual server router (VSR) must be in the same IP subnet as the interface to which it is assigned.

Standard VRRP Components

Table 46 - Standard VRRP Components, page 508 describes the standard VRRP components.

Table 46: Standard VRRP Components

Component	Description
VRRP router	A physical router running VRRP.
Virtual router	Two or more VRRP routers can be configured to form a *virtual router* (as defined in RFC 2338). Each VRRP router may participate in one or more virtual routers. Each virtual router consists of a user-configured *virtual router identifier* (VRID) and an IP address.
Virtual router MAC address	The VRID is used to build the virtual router MAC address. • For legacy-based MAC addresses, the five highest-order octets of the virtual router MAC address are the standard MAC prefix (00-00-5E-00-01) defined in RFC 2338. The VRID is used to form the lowest-order octet. • If HAID is non-zero, the MAC address is 00:03:B2:78:XX:XX. • If HAID=0 for IPv4, the MAC address is 00:00:5E:00:01:XX. • If HAID=0 for IPv6, the MAC address is 00:00:5E:00:02:XX.
Virtual interface IP address owner	A VRRP router in a virtual router whose virtual interface router's IP address matches the real interface address. This router responds to packets addressed to the virtual interface router's IP address for ICMP pings, TCP connections, and so on. Only one of the VRRP routers in a virtual interface router may be configured as the IP address owner. There is no requirement for any VRRP router to be the IP address owner. Most VRRP installations choose not to implement an IP address owner. If the owner is not available, the backup becomes the master and takes responsibility for packet forwarding and responding to Address Resolution Protocol (ARP) requests. However, because this Alteon is not the owner, it does not have a real interface configured with the virtual interface router's IP address.
Virtual interface IP address renter	A VRRP router in a virtual router that is not the IP address owner.
Virtual router master	Within each virtual router, one VRRP router is selected to be the *virtual router master*. If the *IP address owner* is available, it always becomes the virtual router master. For an explanation of the selection process, see How VRRP Priority Decides Which Alteon is the Master, page 509. The master forwards packets sent to the virtual interface router. It also responds to Address Resolution Protocol (ARP) requests sent to the virtual interface router's IP address. The master also sends out periodic advertisements to let other VRRP routers know it is alive, and its priority.
Virtual router backup	A VRRP router within a virtual router not selected to be the master. If the virtual router master fails, one of the virtual router backups becomes the master and assumes its responsibilities.

VRRP Priority

This section describes the following topics:

How VRRP Priority Decides Which Alteon is the Master

According to the VRRP standard, a virtual interface IP address owner has a priority of 255. You configure each renter with a priority of between 1 and 254. If the IP address owner is available, it always become the virtual router master.

The master periodically sends advertisements to an IP multicast address. As long as the backups receive these advertisements, they remain in the backup state. If a backup does not receive an advertisement for three advertisement intervals, it initiates a bidding process to determine which VRRP router has the highest priority and takes over as master.

If, at any time, a backup determines that it has higher priority than the current master, it can preempt the master and become the master itself, unless configured not to do so. In preemption, the backup assumes the role of master and begins to send its own advertisements. The current master sees that the backup has higher priority and stops functioning as the master.

A backup router can stop receiving advertisements for one of two reasons: the master can be down, or all communications links between the master and the backup can be down. If the master has failed, it is clearly desirable for the backup (or one of the backups, if there is more than one) to become the master.

Note: If communication links between the master and the backup are down, but the master is healthy, Alteon may select a second master within the virtual router. To prevent this, configure redundant links between the VRRP devices within the virtual router.

Transitioning from the INIT State Based on VRRP Priority

If there is no port in the virtual router's VLAN with an active link, the interface for the VLAN fails, thus placing the virtual router into the INIT state.

The INIT state identifies that the virtual router is waiting for a startup event. If it receives a startup event, it will either transition to master if its priority is 255 (the IP address owner), or transition to the backup state if it is not the IP address owner.

Determining How to Configure Priority

A virtual router's priority is an initial value that increases or decreases depending on the parameters that are tracked. For example, if you configure the virtual router to track the link state of the physical ports, the virtual router's priority decreases by two priority points if the link to one port fails.

To ensure that a decrease in priority causes failover from the current master to the backup virtual router, set the priority of the master Alteon one point higher than the backup. For example, priority 101 for the master, and 100 for the backup. If the master and backup Alteons are set to priorities 110 and 100 respectively, a single port failure only decreases the master's priority to 108. Since 108 is still higher than the backup's priority of 100, the master does not fail due to the loss of one port link.

Determining VRRP Priority for Ports Outside the VLAN

Alteon checks hot-standby ports when calculating VRRP priority.

- If all hot-standby ports are up, Alteon adds 2 to the VRRP priority, and continues the VRRP tracking calculation.
- If at least one hot-standby port is down, Alteon leaves the VRRP priority unchanged, and does not perform a tracking calculation.

When a vADC has VRRP configured with a hot-standby port that is not part of the VLANs assigned to the vADC, the vADC ignores this port in the VRRP priority calculation.

Alteon Extensions to VRRP

This section describes the following VRRP enhancements implemented in Alteon:

- Virtual Interface Routers, page 510
- Virtual Server Routers, page 510
- Sharing Interfaces for Active-Active Failover, page 511
- Service-Based Virtual Router Groups, page 512
- Tracking VRRP Router Parameters, page 515
- Tracking Service-Based Virtual Router Groups, page 516
- VRRP Holdoff Timer, page 517
- OSPF Cost Update, page 517

Virtual Interface Routers

At Layer 3, a Virtual Interface Router (VIR) allows two VRRP routers to share an IP interface across all routers. VIRs provide a single destination IP (DIP) for upstream routers to reach various destination networks, and provide a virtual default gateway.

A VIR must be assigned an IP interface, and every IP interface must be assigned to a VLAN. When the IP interface of a VIR is down, the VIR is in the INIT state.

Virtual Server Routers

Alteon supports up to 1024 *virtual server routers* (VSRs), which extend the benefits of VRRP to virtual server IP addresses that are used to perform SLB.

Virtual server routers operate for virtual server IP (vip) addresses in much the same manner as virtual interface routers operate for IP interfaces. A master is negotiated via a bidding process, during which information about each VRRP router's priority is exchanged. Only the master can process packets that are destined for the virtual server IP address and respond to ARP requests.

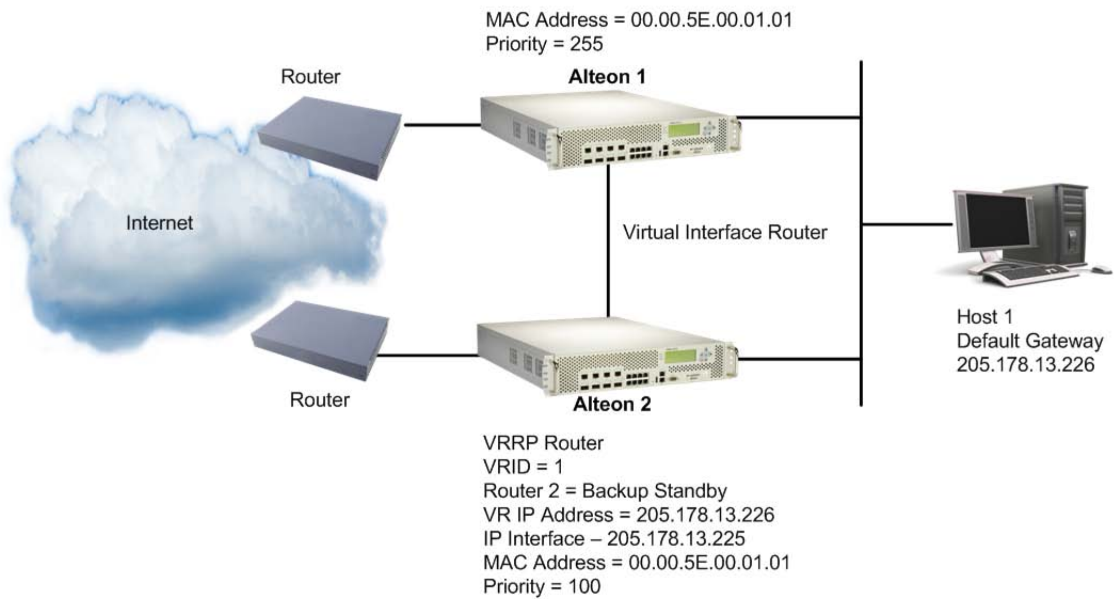
One difference between *virtual server routers* and *virtual interface routers* is that a virtual server router cannot be an *IP address owner*. All virtual server routers are *renters*.

All virtual routers, whether virtual server routers or virtual interface routers, operate independently of one another. That is, their priority assignments, advertisements, and master negotiations are separate. For example, when you configure a VRRP router's priority in a virtual server router, you are not affecting that VRRP router's priority in any virtual interface router or any other virtual server router of which it is a part. However, because of the requirement that MAC addresses be unique on a LAN, VRIDs must be unique among all virtual routers, whether virtual interface routers or virtual server routers.

Alteon VSRs with a virtual router ID (VRID) greater than 255 use a new packet format, which differs in size and location to the VRID field. When sending advertisements using a VSR with a VRID greater than 255, set the type to 15. Alteons that do not support the new packet format discard these packets because VRRP currently only supports one defined packet type (type=1).

In Figure 74 - Virtual Interface Router Configuration, page 511, Alteons are configured as VRRP routers. Together, they form a virtual interface router (VIR).

Figure 74: Virtual Interface Router Configuration



Alteon 1 has its real interface configured with the IP address of the virtual interface router, making it the IP address owner. As the IP address owner, it receives a priority of 255, and is the virtual router master.

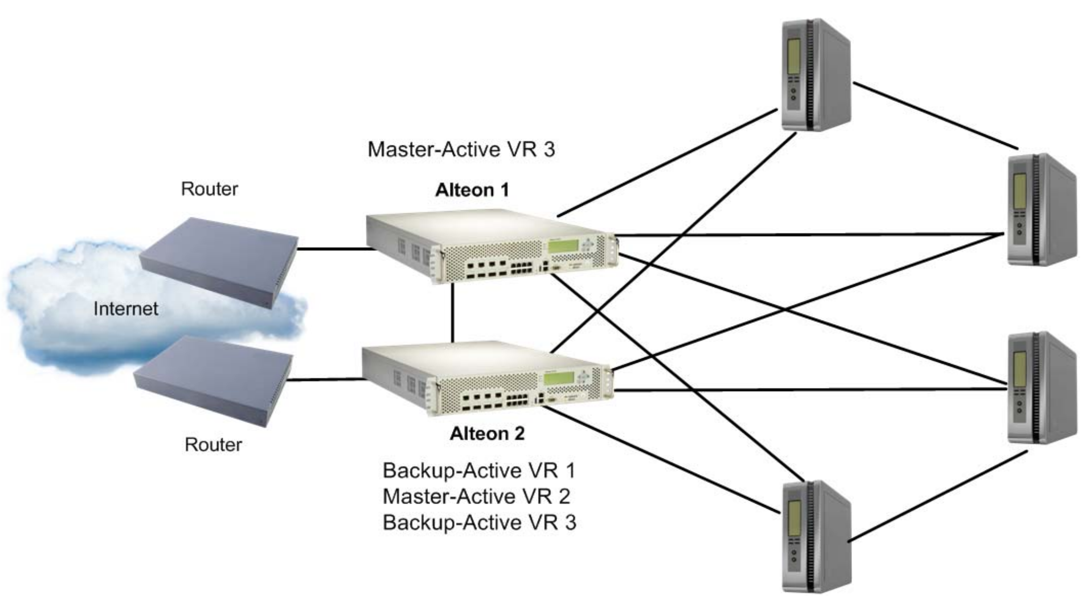
Alteon 2 is a virtual router backup. Its real interface is configured with an IP address that is on the same subnet as the virtual interface router, but is not the IP address of the virtual interface router.

The virtual interface router is assigned a VRID of 1. Both of the VRRP routers have a virtual router MAC address of 00-00-5E-00-01-01.

Sharing Interfaces for Active-Active Failover

Alteon supports sharing of interfaces at both Layer 3 and Layer 4, as shown in Figure 75 - Active-Active Failover with Shared Interfaces, page 511:

Figure 75: Active-Active Failover with Shared Interfaces



With sharing enabled, an IP interface or a VIP address can be active simultaneously on multiple Alteons, enabling active-active operation as shown in Figure 74 - Virtual Interface Router Configuration, page 511 and Table 47 - Active-Active Failover with Shared Interfaces, page 512:

Table 47: Active-Active Failover with Shared Interfaces

Alteon	Virtual Router 1	Virtual Router 2	Virtual Router 3
Alteon 1	Master-Active VRID 2 VIP: 205.178.13.226 Virtual Rtr. MAC address: 00-00-5E-00-01-02	Backup-Active VRID 4 VIP: 205.178.13.240 Virtual Rtr. MAC address: 00-00-5E-00-01-04	Master-Active VRID 6 VIP: 205.178.13.110 Virtual Rtr. MAC address: 00-00-5E-00-01-06
Alteon 2	Backup-Active VR 1 VRID 2 VIP: 205.178.13.226 Virtual Rtr. MAC address: 00-00-5E-00-01-02	Master-Active VR 2 VRID 4 VIP: 205.178.13.240 Virtual Rtr. MAC address: 00-00-5E-00-01-04	Backup-Active VR 3 VRID 6 VIP: 205.178.13.110 Virtual Rtr. MAC address: 00-00-5E-00-01-06

When sharing is used, incoming packets are processed by the Alteon on which they enter the virtual router. The ingress Alteon is determined by external factors, such as routing and Spanning Tree configuration.

Sharing cannot be used in configurations where incoming packets have more than one entry point into the virtual router. For example, where a hub is used to connect Alteons.

When sharing is enabled, the master election process still occurs. Although the process does not affect which Alteon processes packets that must be routed or that are destined for the virtual server IP address, it does determine which Alteon sends advertisements and responds to ARP requests sent to the virtual router's IP address.

Radware strongly recommends that sharing, rather than active-standby configurations, be used whenever possible. Sharing offers both better performance and fewer service interruptions in the face of fault conditions than active-standby configurations. See Active-Active Redundancy, page 527 for a configuration example.

Service-Based Virtual Router Groups

A service-based virtual router group (vrgroup) consists of one or more virtual routers on an Alteon. Virtual routers can be grouped together and behave as a single VRRP entity by updating the priority for the group. Service-based virtual router groups allow for efficient tracking and failover based on each group's tracking parameters while leaving other groups unaffected.

Virtual routers in one vrgroup (`/cfg/l3/vrrp/vrgroup 1`) will not necessarily all have the same status (master, backup or init). By contrast, virtual routers in the global vrrp group (`/cfg/l3/vrrp/group`) will always have the same status.

The priority, tracking and preemption values for each virtual router in a vrgroup are overridden by the values for the vrgroup itself.

Radware recommends that you enable preemption when working with service-based virtual router groups. If you do not want to use preemption, you should work with switch-based virtual router groups instead.

Note: For a vrgroup to work correctly, you must first set virtual router tracking for one of the virtual routers configured for that group using the `cfg/l3/vrrp/vrgroup/trackvr` command. You can only set tracking for one virtual router in a vrgroup.

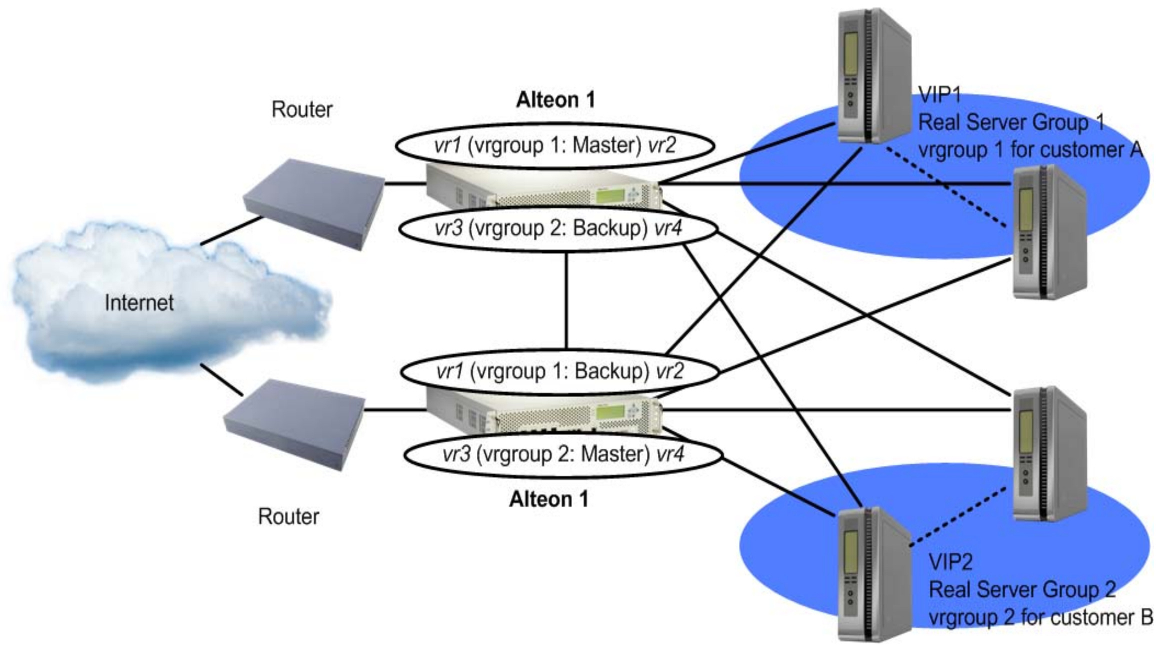
As shown in the example in Figure 76 - Service-Based Virtual Router Groups Configuration, page 513, an administrator wants to provide high availability for Customer A and Customer B's servers and services across the same two Alteons, without one affecting the other:

- Customer A's traffic load balances across real servers in Real Server Group 1.
- Customer B's traffic load balances across real servers in Real Server Group 2.

Each Alteon is configured with vrgroup 1 for Customer A, and vrgroup 2 for Customer B.

Because each vrgroup is tracked independently of the other, vrgroup 1 can fail over to its equivalent vrgroup 1 on the other Alteon while not affecting the VRRP state of vrgroup 2.

Figure 76: Service-Based Virtual Router Groups Configuration



Characteristics of Service-Based Virtual Router Groups (vrgroups)

The following are characteristics of virtual router groups:

- Physical Alteon-based VRRP groups must be disabled (**`/cfg/l3/vrrp/group dis`**)
- Up to 16 vrgroups can be configured on a single Alteon. Each vrgroup can contain up to 64 virtual routers assigned with a virtual router number from 1 through 1024. Each virtual router can be configured as a virtual interface router or a virtual service router.
- Virtual routers that become members of a vrgroup assume the priority tracking parameters configured for that vrgroup.
- When one member of a master vrgroup fails, the priority of the vrgroup decreases, and all the members of that vrgroup change from master to backup. This is done by configuring tracking on the service-based virtual router group.
- You cannot use an IPv6 vrrp group with more than 90 VRs.

Server-Based Virtual Router Group Commands

The following are procedures for server-based virtual router group commands.

To access the *vrgroup* menu

```
>> Main# /cfg/l3/vrrp/vrgroup <vrgroup # 1-16>
```

To add virtual routers to a service-based virtual router group

These set of commands are based on Figure 76 - Service-Based Virtual Router Groups Configuration, page 513.

`>> Main# /cfg/l3/vrrp/vrgroup 1`	(Select vrgroup 1)
`>> VRRP Virtual Router Vrgroup 1# add 1`	(Add virtual router 1 to vrgroup 1)
`>> VRRP Virtual Router Vrgroup 1# add 2`	(Add virtual router 2 to vrgroup 1)
`>> Main# /cfg/l3/vrrp/vrgroup 2`	(Select vrgroup 2)
`>> VRRP Virtual Router Vrgroup 2# add 3`	(Add virtual router 3 to vrgroup 2)
`>> VRRP Virtual Router Vrgroup 2# add 4`	(Add virtual router 4 to vrgroup 2)

See Tracking VRRP Router Parameters, page 515 for a configuration example.

Switch-Based VRRP Groups

A switch-based virtual router group aggregates all virtual routers on Alteon as a single entity for non-shared environments. All virtual routers will failover as a group, and cannot failover individually. As members of a group, all virtual routers on Alteon (and therefore on Alteon itself), are either in a master or backup state.

Characteristics of a Switch-Based VRRP Group

The following are characteristics of a switch-based VRRP group:

- When enabled, all virtual routers behave as one entity, and all group settings override any individual virtual router settings or service-based vrgroup settings.
- All individual virtual routers, once the switch-based VRRP group is enabled, assume the group's tracking and priority.
- When one member of a switch-based VRRP group fails, the priority of the group decreases, and the state of the entire Alteon changes from master to backup.
- If an Alteon is in the backup state, Layer 4 processing is still enabled. If a virtual server is not a virtual router, the backup can still process traffic addressed to that virtual server IP address. Filtering is also still functional. Only traffic addressed to virtual server routers is not processed.
- Each VRRP advertisement can include up to 1024 addresses. All virtual routers are advertised within the same packet, conserving processing and buffering resources.

Note: A switch-based virtual router group cannot be used for active-active configurations or any other configuration that requires shared interfaces.

Switch-Based VRRP Group Commands

The following are procedures for server-based VRRP group commands.

To configure a switch-based VRRP group

```
>> Main# /cfg/l3/vrrp/group ena
```

For more information on using switch-based VRRP groups with hot standby, see Hot Standby Configuration, page 537.

Tracking VRRP Router Parameters

Alteon supports a tracking function that dynamically modifies the priority of a VRRP router based on its current state. The objective of tracking is to have, whenever possible, the master bidding processes for various virtual routers in a LAN converge on the same Alteon. Tracking ensures that the selected Alteon is the one that offers optimal network performance. For tracking to have any effect on virtual router operation, preemption must be enabled.

Note: Tracking only affects hot standby and active-standby configurations. It does not have any effect on active-active sharing configurations.

Alteon can track the parameters outlined in the following table:

Table 48: VRRP Tracking Parameters

Parameters and Commands	Description
Number of virtual routers in master mode on Alteon. • To enable tracking on VRs: `/cfg/l3/vrrp/vr<#>/track/vrs/ena` • To change the virtual router increment: `/cfg/l3/vrrp/track/vrs <[0-254]>`	Useful for ensuring that traffic for any particular client/server pair is handled by the same Alteon, increasing routing and load balancing efficiency. This parameter influences the VRRP router's priority in both virtual interface routers and virtual server routers. **Note:** The **vrs** parameter is not available for tracking for a service-based virtual router group (vrgroup).
Number of IP interfaces active on Alteon. • To enable tracking on IP interfaces: `/cfg/l3/vrrp/vr/track/ifs <#> /ena` • To change the interfaces tracking increment: `/cfg/l3/vrrp/track/ifs <[0-254]>`	Helps elect the virtual routers with the most available routes as the master. An IP interface is considered active when there is at least one active port on the same VLAN. This parameter influences the VRRP router's priority in both virtual interface routers and virtual server routers.
Number of active ports on the same VLAN. • To enable tracking on ports on the same VLAN: `/cfg/l3/vrrp/vr/track/port<#>/ena` • To change the ports tracking increment: `/cfg/l3/vrrp/track/ports <[0-254]>`	Helps elect the virtual routers with the most available ports as the master. This parameter influences the VRRP router's priority in both virtual interface routers and virtual server routers.

Table 48: VRRP Tracking Parameters (cont.)

Parameters and Commands	Description

Number of physical ports that have active Layer 4 processing.
• To enable tracking on Layer 4 ports:
`/cfg/l3/vrrp/vr/track/l4pts/ena`
• To change the Layer 4 ports tracking increment:
`/cfg/l3/vrrp/track/l4pts <[0-254]>`
	Helps elect the main Layer 4 Alteon as the master. This parameter influences the VRRP router's priority in both virtual interface routers and virtual server routers.

Number of healthy real servers behind the virtual server IP address that is the same as the IP address of the virtual server router on Alteon. `/cfg/l3/vrrp/track/reals`	Helps elect the Alteon with the largest server pool as the master, increasing Layer 4 efficiency. This parameter influences the VRRP router's priority in virtual server routers only.
In networks where the Hot Standby Router Protocol (HSRP) is used for establishing router failover, the number of Layer 4 client-only ports that receive HSRP advertisements. `/cfg/l3/vrrp/track/hsrp`	Helps elect the Alteon closest to the master HSRP router as the master, optimizing routing efficiency. This parameter influences the VRRP router's priority in both virtual interface routers and virtual server routers.
Tracking HSRP on VLAN. `/cfg/l3/vrrp/track/hsrv`	A hot standby router on VLAN (HSRV) is used in VLAN-tagged environments. Enable this option to increment only that VRRP instance that is on the same VLAN as the tagged HSRP master flagged packet. This command is disabled by default.

Each tracked parameter is associated with a user-configurable weight. As the count associated with each tracked item increases (or decreases), so does the VRRP router's priority, subject to the weighting associated with each tracked item. If the priority level of a backup is greater than that of the current master, then the backup can assume the role of the master.

For an example on how to configure Alteon for tracking VRRP priority, see Tracking Virtual Routers, page 542.

Tracking Service-Based Virtual Router Groups

Alteon supports a tracking function that dynamically modifies the priority of a service-based virtual router group (vrgroup), which contains one or more virtual routers. Once a VRRP router is added to a vrgroup, the group's tracking configuration overrides an individual VRRP router's tracking.

Alteon allows for the independent failover of individual virtual router groups on the same Alteon. When Web hosting is shared between two or more customers on a single VRRP device, several virtual routers can be grouped to serve the high availability needs of a specific customer.

Each vrgroup is treated as a single entity regardless of how many virtual routers belong to the vrgroup. When Alteon tracks a vrgroup, it measures the resources contained in this group, and updates all members of the vrgroup with the same priority. When any of the tracked parameters changes the priority for one of the virtual routers belonging to the vrgroup, then the entire vrgroup will fail over.

Tracking can be configured for each vrgroup, with the same resources tracked on individual virtual routers (see Table 48 - VRRP Tracking Parameters, page 515). The only resource that cannot be tracked on a vrgroup basis is the number of virtual routers.

If failover occurs on a customer link, only the group of virtual routers associated with that customer's vrgroup will fail over to the backup. Other vrgroups configured for other customers do not fail over. For example, if a vrgroup is configured to track ports, a port failure will decrease the priority of the vrgroup. The lowered priority causes this vrgroup to fail over to its equivalent vrgroup on the other Alteon.

For an example on how to configure Alteon for tracking VRRP priority, see Service-Based Virtual Router Groups, page 543.

VRRP Holdoff Timer

When an Alteon becomes the VRRP master at power up or after a failover operation, it may begin to forward data traffic before the connected gateways or real servers are operational. Alteon may create empty session entries for the coming data packets and the traffic cannot be forwarded to any gateway or real server.

Alteon supports a VRRP holdoff timer, which pauses VRRP instances from starting or changing to master state during the initialization.The VRRP holdoff timer can be set from 0 to 255 seconds. The VRRP master waits the specified number of seconds before forwarding traffic to the default gateway and real servers.

To set the VRRP holdoff timer

```
>> Main# /cfg/l3/vrrp/holdoff <0-255 seconds>
```

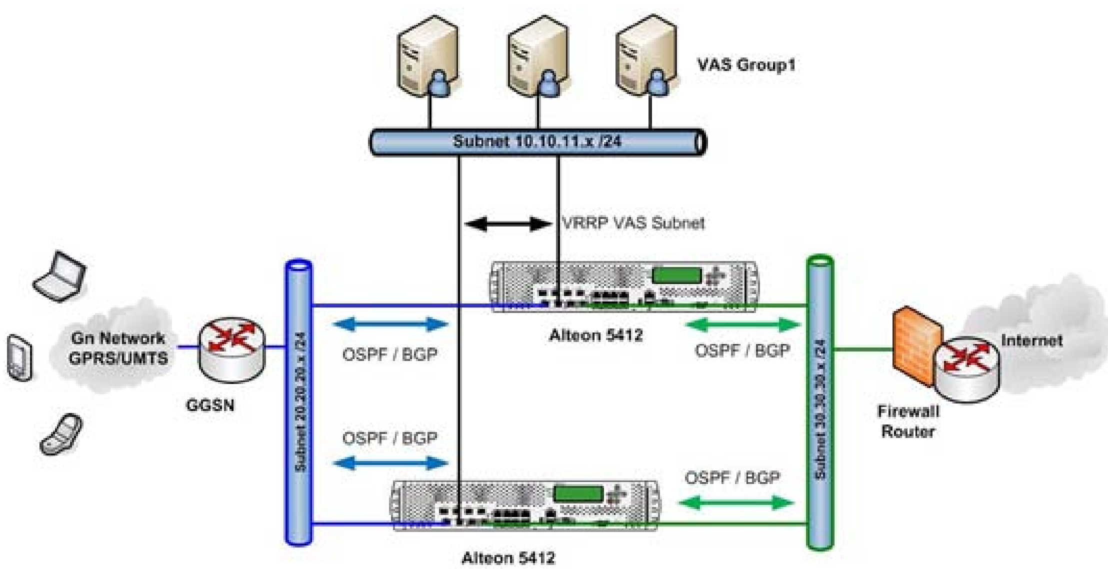
OSPF Cost Update

Starting with version 28.1.50.0, Alteon supports OSPF cost updates based on VRRP status. This includes new VRRP tracking options and OSPF cost awareness.

Using cost updating, the entire OSPF path remains consistent across multiple links, ensuring that services are not interrupted.

Figure 77 - OSPF VRRP Topology Using Cost Updating, page 517 illustrates an example of OSPF VRRP topology using cost updating.

Figure 77: OSPF VRRP Topology Using Cost Updating



In this example, VRRP is configured as active-active. Both Alteons are OSPF-enabled and receive traffic. The following is a further explanation of Figure 77 - OSPF VRRP Topology Using Cost Updating, page 517:

1. The cost of the first Alteon is less than the cost of the second Alteon.
2. Mobile clients send traffic from network 20.20.20.*x* through the first Alteon to GGSN on network 30.30.30.*x*.
3. Alteon intercepts and redirects the traffic based on the HTTP policy of the 10.10.11.*x* network.
4. The 10.10.10.*x* network does not appear in the OSPF routing and is accessed only by Alteon.
5. If the link between the first Alteon and the 10.10.11.*x* network fails, OSPF is not affected because the interface of the 10.10.10.X network is not bound to OSPF.
6. The traffic passes from the mobile clients to the first Alteon and the service is interrupted.
7. If the link fails when the traffic returns from the servers in the 10.10.10.*x* network, traffic returns through the second Alteon. This causes an asymmetric routing traffic flow.

VRRP cost update support does not require any changes to the OSPF settings. The VRRP functionality is part of the existing tracking options. This enables OSPF to remain a pure routing protocol regardless of the services running on top of it.

OSPF maintains a cost value per interface flexibility designed for routers creating deterministic paths. In the example in Figure 77 - OSPF VRRP Topology Using Cost Updating, page 517, the traffic flow is handled as a service with path dependencies. That is, the service paths are related and affect one another.

You can set the OSPF cost increment for the VR (single interface), VR group (multiple interface), and group (multiple interface). For more information on configuring the OSPF cost, refer to the *Alteon Application Switch Operating System Command Reference*.

IPv6 VRRP Support

Alteon supports using IPv6 with VRRP. For background information on IPv6, see IPv6, page 835.

This section describes the following topics:

IPv6 VRRP Support Overview

IPv6 hosts on a VLAN usually learn about other routers by receiving IPv6 routing advertisements. The routing advertisements are multicast periodically at a rate such that the hosts usually learn about the other routers within a few minutes. They are not sent frequently enough for the hosts to rely on them to detect router failures.

IPv6 hosts can also use the neighbor discovery mechanism to detect router failure by sending unicast neighbor solicitation messages to the other routers. By using the default setting, it takes a host about 38 seconds to learn that a router is unreachable before it switches to another router.

IPv6 VRRP support provides a much faster mechanism for the switch over to a backup router than can be obtained using standard neighbor discovery procedures. Using IPv6 VRRP support, a backup router can take responsibility for the virtual router master within seconds. This is done without any interaction with the hosts, and a minimum amount of traffic in the subnet.

Two types of addresses are used in IPv6 that facilitate VRRP support:

- **Unicast address**—The global unicast address is an address that is accessible and identifiable globally.

The link-local unicast address is an address used to communicate with neighbors on the same link. The source address of an IPv6 VRRP packet is set to the IPv6 link-local address of the transmission interface.

- **Multicast address**—The IPv6 multicast address is an identifier for a group interface. IPv6 VRRP support has an IPv6 link-local scope multicast address assigned by IANA. This multicast address follows the format `FF02:0:0:0:0:0:XXXX:XXXX`. The destination address of the IPv6 packet is set to this link-local scope multicast address. A router must not forward a datagram with this destination address regardless of its hop limit setting.

IPv6 VRRP Packets

IPv6 VRRP packets differ in some aspects from VRRP implemented in an IPv4 network. The key differences are:

- The **Version** field specifies the VRRP protocol version. In IPv4 packets this value is 2, and in IPv6 packets this value is 3.
- The **Authentication Type** field is not present in IPv6 packets. This field is used in IPv4 to identify the authentication method in use.
- The **Advertisement Interval** field is a 12-bit field that indicates the advertisement interval in centiseconds (1/100 second). This is an 8-bit field in IPv4 that specifies this interval in seconds.

Note: Radware recommends setting the default to 100 (1 second) or greater to avoid a high load on the management CPU.

- The **Hop Limit** field is used to track how many nodes have forwarded the packet. The field value is decremented by one for each node that forwards the packet. VRRP routers are instructed to discard IPv6 VRRP packets that do not have a Hop Limit value of 255.
- The **Next Header** field is used to identify the type of protocol immediately following the IPv6 header. The IPv6 Next Header assigned by IANA for VRRP is 112.
- The neighbor discovery protocol replaces IPv4 ARP, ICMP router discovery, and ICMP redirection. Neighbor discovery enables nodes (hosts and routers) to determine the link-layer address of a neighbor on the same network and to detect any changes in these addresses. It also enables a router to advertise its presence and address prefix to inform hosts of a better next hop address to forward packets.

IPv6 VRRP Configuration

This section includes the two procedures required to enable IPv6 VRRP support.

Notes

- You cannot use IPv6 VRRP groups with more than 90 virtual routers.
- The VRRP3 VRID for IPv6 VRRP configuration has a range of 1 to 255.

To enable IPv6 support on the virtual router

1. Change the IP version supported by the virtual router.

   Use the command `/cfg/l3/vrrp/vr <virtual router number> /ipver v6` to configure the virtual router for IPv6 support.

2. Assign an IPv6 address to the virtual router.

   Use the command address <IPv6_address> to assign an IPv6 address to the virtual router.

To enable IPv6 support on the virtual router group

After IPv6 support has been enabled on the virtual router, enable it on the virtual router group using the /cfg/l3/vrrp/group/ipver v6 command.

IPv6 VRRP Information

The following are sample informational and statistical displays for IPv6 VRRP support.

To view IPv6 VRRP information

```
>> Main# /info/l3/vrrp
VRRP information:
    9: vrid    9, 2005:0:0:0:0:0:10:9
                                  if  9, renter, prio 101, master
   10: vrid   10, 10.10.10.50,    if  1, renter, prio 101, master
   20: vrid   20, 2005:0:0:0:0:0:20:20
                                  if 20, renter, prio 105, master, server
```

To view IPv6 VRRP statistics

```
>> Main# /stat/l3/vrrp6
--------------------------------------------------------------------------
VRRP6 statistics information:
vrrp6InAdvers:                                 7
vrrp6BadAdvers:                                0
vrrp6OutAdvers:                            86801
vrrp6BadVersion:                               0
vrrp6BadVrid:                                  0
vrrp6BadAddress:                               0
vrrp6BadData:                                  0
vrrp6BadInterval:                              0
```

Failover Methods and Configurations

Alteon has the flexibility to implement redundant configurations.

High availability configurations are based on VRRP. VRRP implementation includes proprietary extensions to accommodate Layer 4 though Layer 7 load balancing features.

This section describes a few of the more useful and easily deployed configurations, including:

- Active-Standby Redundancy, page 521
- Active-Active Redundancy, page 527
- Hot Standby Redundancy, page 535
- Tracking Virtual Routers, page 542
- Service-Based Virtual Router Groups, page 543

Note: The current configurations described in this section are valid without session synchronization.

Active-Standby Redundancy

This section describes the following topics:

- Active-Standby Environments, page 521
- Configuring Active-Standby Redundancy, page 522

Active-Standby Environments

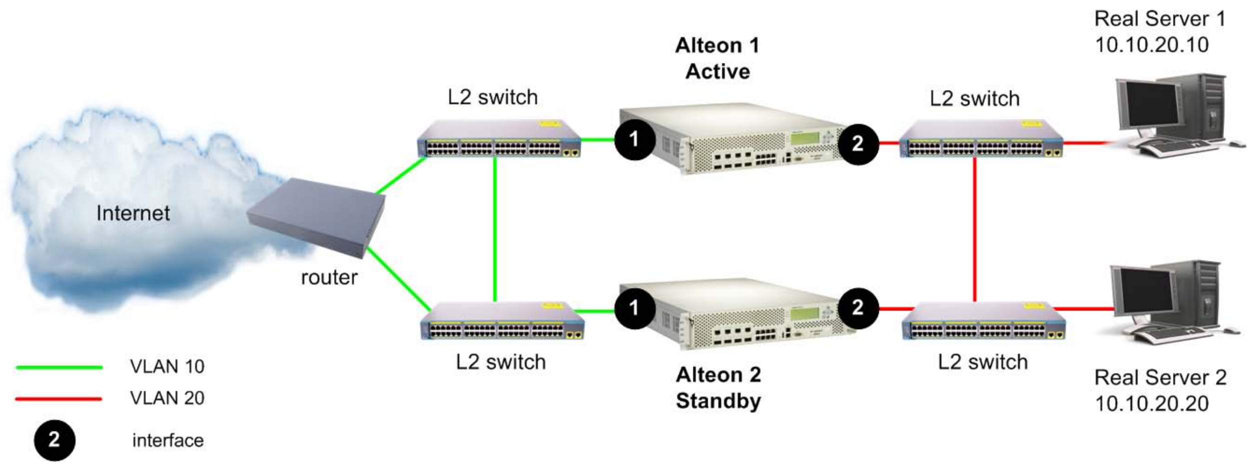
In an active-standby configuration, the active switch supports all traffic or services. The backup switch acts as a standby for services on the active master switch. If the master switch fails, the backup switch takes over processing for all services. The backup switch may forward Layer 2 and Layer 3 traffic, as appropriate.

Radware recommends that you do not allow sharing between the Alteon devices. Without sharing, only the active Alteon performs load balancing. This is a very robust configuration that does not require dedicated interswitch links (ISL), or hotstandby settings on ports.

Note: In an active-standby configuration, the same service cannot be active simultaneously on both devices.

[Figure 78 - Active-Standby Configuration, page 522](#) shows an active-standby configuration. In this example, there are two VLANs, each with their own interface. The same two services are configured on each Alteon.

Figure 78: Active-Standby Configuration



Configuring Active-Standby Redundancy

Perform the following steps on both the active and the standby Alteon:

1. Disable the Spanning Tree protocol. For more information, see To disable the Spanning Tree protocol, page 523.

   Using the Spanning Tree protocol or VLANs prevents Layer 2 loops. Radware recommends that you use VLANs.

   Note: The configuration does not require dedicated interswitch links (ISL), or hotstandby settings on ports.

2. Enable IP forwarding. For more information, see To enable IP forwarding, page 523.
3. Configure two interfaces, one for each VLAN. For more information, see To configure Layer 3 physical interface settings, page 523.
4. Configure virtual routers—one for each interface, and one for each service. For more information, see To configure Layer 3 virtual router settings for VLANs, page 524.
5. Define each service on a virtual server, and associate each service with a virtual router. For more information, see To configure Layer 3 virtual router settings for services, page 526.
6. Add all virtual routers to a VRRP group. For more information, see To configure VRRP grouping for virtual routers, page 527.

   Grouping virtual routers enables you to easily give them a common status (active or standby). A virtual router can belong to one VRRP group only.

To disable the Spanning Tree protocol

1. Perform the following steps:

`>> Main# /cfg/l2/stg 1`	(Select the STP Group number)
`>> Main# /cfg/l2/stg 1/off`	(Disable STP)
`>> Main# /cfg/l2/stg 1/apply`	(Make your changes active)

2. Repeat for other Spanning Tree groups. Alteon supports up to 16 Spanning Tree groups.

To enable IP forwarding

IP forwarding is enabled by default. Make sure IP forwarding is enabled if the virtual server IP addresses and real server IP addresses are on different subnets, or if the device is connected to different subnets and those subnets need to communicate through the device. If you are not sure whether to enable IP forwarding, enable it as follows:

```
>> Main# /cfg/l3/frwd/on
```

To configure Layer 3 physical interface settings

1. On the active Alteon, configure two interfaces and associate a different VLAN with each interface.

   Each interface has a unique IP address.

`>> Main # /cfg/l3/if 1`	(Name the device interface)
`>> Main # /cfg/l3/if 1/ena`	(Enable the interface)
`>> Main # /cfg/l3/if 1/addr 10.10.10.253`	(Set the IP address for the interface)
`>> Main # /cfg/l3/if 1/mask 255.255.255.0`	(Set the subnet mask for the interface)
`>> Main # /cfg/l3/if 1/vlan 10`	(Set the VLAN number for the interface)
`>> Main # /cfg/l3/if 2`	(Name the device interface)
`>> Main # /cfg/l3/if 2/ena`	(Enable the interface)
`>> Main # /cfg/l3/if 2/addr 10.10.20.253`	(Set the IP address for the interface)
`>> Main # /cfg/l3/if 2/mask 255.255.255.0`	(Set the subnet mask for the interface)

`>> Main # /cfg/l3/if 2/vlan 20`	(Set the VLAN number for the interface)

2. On the standby Alteon, configure two more interfaces and associate a different VLAN with each interface.

   Each interface has a unique IP address.

`>> Main # /cfg/l3/if 1`	(Name the device interface)
`>> Main # /cfg/l3/if 1/ena`	(Enable the interface)
`>> Main # /cfg/l3/if 1/addr 10.10.10.252`	(Set the IP address for the interface)
`>> Main # /cfg/l3/if 1/mask 255.255.255.0`	(Set the subnet mask for the interface)
`>> Main # /cfg/l3/if 1/vlan 10`	(Set the VLAN number for the interface)
`>> Main # /cfg/l3/if 2`	(Name the device interface)
`>> Main # /cfg/l3/if 2/ena`	(Enable the interface)
`>> Main # /cfg/l3/if 2/addr 10.10.20.252`	(Set the IP address for the interface)
`>> Main # /cfg/l3/if 2/mask 255.255.255.0`	(Set the subnet mask for the interface)
`>> Main # /cfg/l3/if 2/vlan 20`	(Set the VLAN number for the interface)

To configure Layer 3 virtual router settings for VLANs

1. On the active Alteon, configure a different virtual router for each VLAN.

`>> Main # /cfg/l3/vrrp/on`	(Enable VRRP)
`>> Main # /cfg/l3/vrrp/vr 1`	(Specify the virtual router number for VLAN 10 at interface 1)
`>> Main # /cfg/l3/vrrp/vr 1/ena`	(Enable the virtual router)
`>> Main # /cfg/l3/vrrp/vr 1/vrid 25`	(Set the virtual router ID)
`>> Main # /cfg/l3/vrrp/vr 1/if 1`	(Select a device IP interface)
`>> Main # /cfg/l3/vrrp/vr 1/prio 101`	(Set the priority bias for the virtual router)
`>> Main # /cfg/l3/vrrp/vr 1/addr 10.10.10.254`	(Set the virtual router IP address)
`>> Main # /cfg/l3/vrrp/vr 2`	(Specify the virtual router number for VLAN 20 at interface 2)

`>> Main # /cfg/l3/vrrp/vr 2/ena`	(Enable the virtual router)
`>> Main # /cfg/l3/vrrp/vr 2/vrid 35`	(Set the virtual router ID)
`>> Main # /cfg/l3/vrrp/vr 2/if 2`	(Select a device IP interface)
`>> Main # /cfg/l3/vrrp/vr 2/prio 101`	(Set the priority bias for the virtual router)
`>> Main # /cfg/l3/vrrp/vr 2/addr 10.10.20.254`	(Set the virtual router IP address)

2. On the standby Alteon, copy these active Alteon settings, but lower the priority of each virtual router.

`>> Main # /cfg/l3/vrrp/on`	(Enable VRRP)
`>> Main # /cfg/l3/vrrp/vr 1`	(Specify the virtual router number for VLAN 10 at interface 1)
`>> Main # /cfg/l3/vrrp/vr 1/ena`	(Enable the virtual router)
`>> Main # /cfg/l3/vrrp/vr 1/vrid 25`	(Set the virtual router ID)
`>> Main # /cfg/l3/vrrp/vr 1/if 1`	(Select a device IP interface)
`>> Main # /cfg/l3/vrrp/vr 1/prio 100`	(Set the priority bias for the virtual router)
`>> Main # /cfg/l3/vrrp/vr 1/addr 10.10.10.254`	(Set the virtual router IP address)
`>> Main # /cfg/l3/vrrp/vr 2`	(Specify the virtual router number for VLAN 20 at interface 2)
`>> Main # /cfg/l3/vrrp/vr 2/ena`	(Enable the virtual router)
`>> Main # /cfg/l3/vrrp/vr 2/vrid 35`	(Set the virtual router ID)
`>> Main # /cfg/l3/vrrp/vr 2/if 2`	(Select a device IP interface)
`>> Main # /cfg/l3/vrrp/vr 2/prio 100`	(Set the priority bias for the virtual router)
`>> Main # /cfg/l3/vrrp/vr 2/addr 10.10.20.254`	(Set the virtual router IP address)

To configure Layer 3 virtual router settings for services

1. On the active Alteon, configure each service.

   For example, for an HTTP service:

`>> Main # /cfg/slb/virt 1`	(Select the virtual server)
`>> Main # /cfg/slb/virt 1/ena`	(Enable the virtual server)
`>> Main # /cfg/slb/virt 1/vip 10.10.10.200`	(Set the virtual server virtual IP)
`>> Main # /cfg/slb/virt 1/service 80 http`	(Specify the service)

2. On the active Alteon, configure a different virtual router for each service.

`>> Main # /cfg/l3/vrrp/on`	(Enable VRRP)
`>> Main # /cfg/l3/vrrp/vr 4`	(Specify the virtual router number for the service)
`>> Main # /cfg/l3/vrrp/vr 4/ena`	(Enable the virtual router)
`>> Main # /cfg/l3/vrrp/vr 4/vrid 55`	(Set the virtual router ID)
`>> Main # /cfg/l3/vrrp/vr 4/if 1`	(Select a device IP interface)
`>> Main # /cfg/l3/vrrp/vr 4/prio 101`	(Set the priority bias for the virtual router)
`>> Main # /cfg/l3/vrrp/vr 4/addr 10.10.10.200`	(Set the virtual server IP address)

3. On the standby Alteon, copy these active Alteon settings, but lower the priority of each virtual router.

`>> Main # /cfg/l3/vrrp/on`	(Enable VRRP)
`>> Main # /cfg/l3/vrrp/vr 4`	(Specify the virtual router number for the service)
`>> Main # /cfg/l3/vrrp/vr 4/ena`	(Enable the virtual router)
`>> Main # /cfg/l3/vrrp/vr 4/vrid 55`	(Set the virtual router ID)
`>> Main # /cfg/l3/vrrp/vr 4/if 1`	(Select a device IP interface)
`>> Main # /cfg/l3/vrrp/vr 4/prio 100`	(Set the priority bias for the virtual router)
`>> Main # /cfg/l3/vrrp/vr 4/addr 10.10.10.200`	(Set the virtual server IP address)

To configure VRRP grouping for virtual routers

1. On the active Alteon, define a single VRRP group for all virtual routers.

`>> Main # /cfg/l3/vrrp/group en`	(Enable VRRP grouping)
`>> Main # /cfg/l3/vrrp/group/vrid 60`	(Specify the virtual router ID for the VRRP group)
`>> Main # /cfg/l3/vrrp/group/if 1`	(Set the IP interface to which VRRP group advertisements will be sent)
`>> Main # /cfg/l3/vrrp/group/prio 101`	(Set the priority base for all virtual routers in the VRRP group)
`>> Main # /cfg/l3/vrrp/group/share dis`	(Disable sharing for the VRRP group and all the virtual routers in the group)
`>> Main # /cfg/l3/vrrp/group/track/if e`	(Set tracking to IP interfaces)

2. On the standby Alteon, define the same VRRP group with a lower base priority.

`>> Main # /cfg/l3/vrrp/group en`	(Enable VRRP grouping)
`>> Main # /cfg/l3/vrrp/group/vrid 60`	(Specify the virtual router ID for the VRRP group)
`>> Main # /cfg/l3/vrrp/group/if 1`	(Set the IP interface to which VRRP group advertisements will be sent)
`>> Main # /cfg/l3/vrrp/group/prio 100`	(Set the priority base for all virtual routers in the VRRP group)
`>> Main # /cfg/l3/vrrp/group/share dis`	(Disable sharing for the VRRP group and all the virtual routers in the group)
`>> Main # /cfg/l3/vrrp/group/track/if e`	(Set tracking to IP interfaces)

Active-Active Redundancy

This section describes the following topics:

- Active-Active Environments, page 527
- Configuring Active-Active Redundancy, page 529

Active-Active Environments

This configuration is based on proprietary Alteon extensions to VRRP.

Alteon has extended VRRP features to include virtual servers, allowing full active-active redundancy between its Layer 4 devices. In an active-active configuration, both switches can process traffic for the same service at the same time. Both switches share interfaces at Layer 3 and Layer 4, meaning that both switches can be active simultaneously for a given IP routing interface or load-balancing virtual server (VIP).

This configuration is often used to allow two different data centers located at different locations to have different Internet access paths.

Since both sites work independently, these virtual routers are not grouped, and priority is not dynamically changed by tracking.

In an active-active configuration, the active switch supports all traffic or services. The backup switch acts as a standby for services on the active master switch. If the master switch fails, the backup switch takes over processing for all services. The backup switch may forward Layer 2 and Layer 3 traffic, as appropriate.

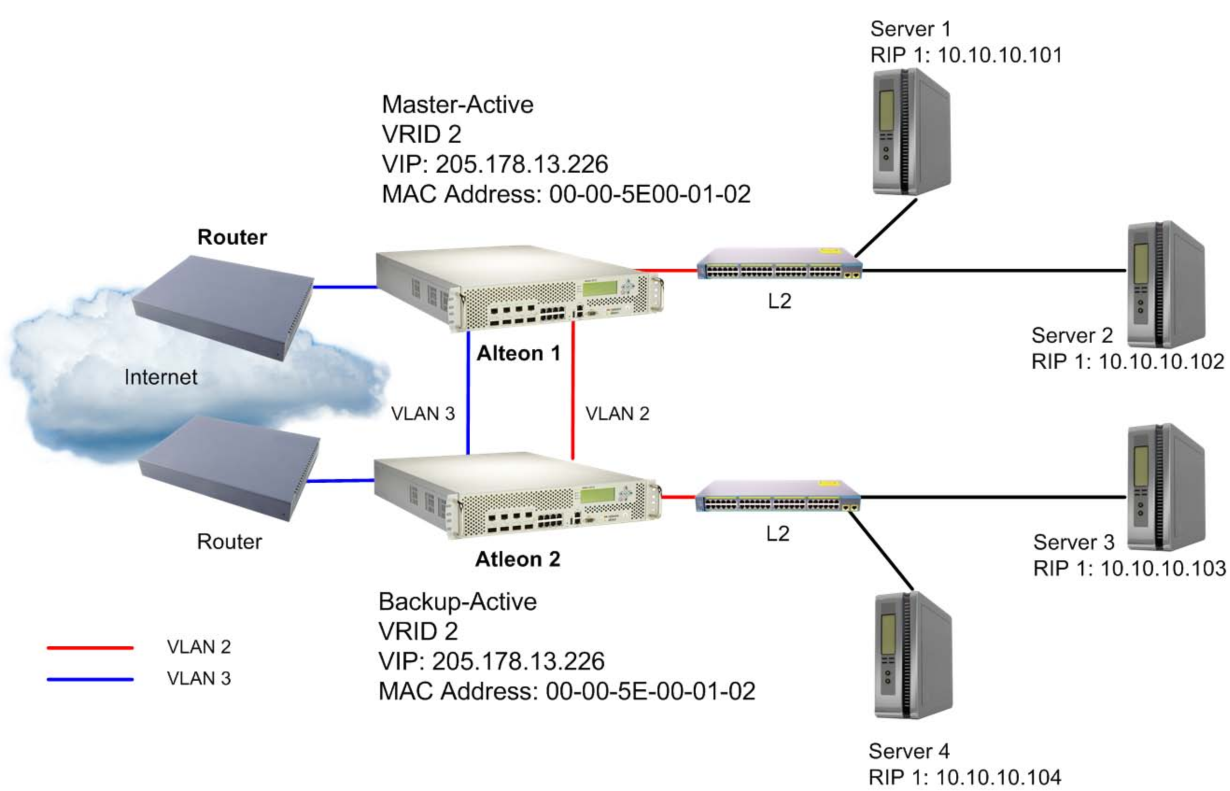
Alteon supports active-active redundancy in a shared or a non-shared environment.

- In a *shared* (or service-based) environment, two Alteon devices are used. Both devices support active traffic but are configured so that they do not simultaneously support the same service. Each Alteon is active for its own set of services, such as IP routing interfaces or load balancing virtual server IP addresses, and acts as a standby for other services on the other Alteon. If either Alteon fails, the remaining Alteon takes over processing for all services. The backup may forward Layer 2 and Layer 3 traffic, as appropriate.
- In a *non-shared* (or switch-based) environment, two Alteon devices are used as VRRP routers, implementing a virtual server router (VSR). The active switch supports all traffic or services. The backup switch acts as a standby for services on the active master switch. If the master switch fails, the backup switch takes over processing for all services. The backup switch may forward Layer 2 and Layer 3 traffic, as appropriate. When both devices are healthy, only the master responds to packets sent to the virtual server IP address. This is a very robust solution that allows up to 100% of throughput.

  This environment is suitable for configurations that cannot support sharing of interfaces at Layer 3 and Layer 4. This includes configurations where incoming packets are seen by more than one device, such as instances where a hub is used to connect the devices.

Figure 78 - Active-Standby Configuration, page 522 shows an active-active configuration in a non-shared environment. In this example, there are two VLANs, each with its own interface. There are four virtual servers. Each virtual server runs a unique service.

Figure 79: Active-Active Configuration in Non-shared Environment



In this configuration, when both devices are healthy, the load-balanced packets are sent to the virtual server IP address (205.178.13.226 in Figure 78 - Active-Standby Configuration, page 522), resulting in higher capacity and performance than when the devices are used in an active-standby configuration.

The Alteon device on which a frame enters the virtual server router is the one that processes that frame. The ingress device is determined by external factors, such as routing and STP settings.

Note: Each VRRP-capable device is autonomous. There is no requirement that the devices in a virtual router be identically configured. Different Alteon models with different numbers of ports and different enabled services may be used in a virtual router.

Configuring Active-Active Redundancy

Perform the following steps on both the active and the standby Alteon:

1. Configure the appropriate Layer 2 and Layer 3 parameters on both devices.

   This configuration includes any required VLANs, IP interfaces, default gateways, and so on. If IP interfaces are configured, none of them should use the VIP address described in step 3. For more information, see To configure background configuration, page 530.
2. Define all filters required for your network configuration. Filters may be configured on one device and synchronized with the settings on the other device (see step 5).
3. Configure all required SLB parameters on one of the devices. For the purposes of this example, assume that Alteon 1 is configured in this step. Configure a VIP set to 205.178.13.226, and one real server group with two real servers:

   RIP 10.10.10.103 should be configured as a backup server to RIP 10.10.10.101.

   RIP 10.10.10.104 should be configured as a backup server to RIP 10.10.10.102.

   Note: In this configuration, each server's backup is attached to the other device. This ensures that operation continues if all of the servers attached to a device fail.

   For more information, see To configure SLB, page 531.
4. Configure the VRRP parameters on the device, as follows:
   — Set the VRID set to 2.
   — Set the VIP address to 205.178.13.226
   — Set the priority to 101.
   — Enable sharing.

   For more information, see To configure virtual router redundancy, page 533.
5. Configure virtual router redundancy:
   — Disable synchronization of VRRP priority to Alteon 2. Use the `/cfg/slb/sync/prios dis` command. This leaves Alteon 2 with its default priority of 100.
   — Synchronize the SLB and VRRP configurations by pushing the configuration from Alteon 1 to Alteon 2. Use the `/oper/slb/sync` command.

   For more information, see To configure virtual router redundancy, page 533.
6. Reverse the roles of the real servers and their backups in Alteon 2's configuration:

   RIP 10.10.10.101 should be configured as a backup server to RIP 10.10.10.103.

   RIP 10.10.10.102 should be configured as a backup server to RIP 10.10.10.104.

In this configuration, if a link between a device and a server fails, the server fails health checks and its backup (attached to the other device) goes online. If a link between a device and its Internet router fails, the protocol used to distribute traffic between the routers (for example, OSPF) reroutes traffic to the other router. Since all traffic enters the virtual server router on one device, that device processes all incoming connections.

If an entire master device fails, the backup detects this failure because it no longer receives advertisements. The backup assumes the master's responsibility of responding to ARP requests and issuing advertisements.

For more information, see To configure the second Alteon device, page 534.

Note: The maxconn metric is not shared between devices. Therefore, if a server is used for normal operation by one device and is activated simultaneously as a backup by the other device, the total number of possible connections to that server is the sum of the maximum connection limits defined for it on both devices.

To configure background configuration

In this procedure, you perform the following:

- Define IP interfaces
- Define VLANs
- Disable the Spanning Tree protocol
- Verify that IP forwarding is enabled

1. Define the IP interfaces. Alteon needs an IP interface for each subnet to which it will be connected so it can communicate with devices attached to it. Each interface needs to be placed in the appropriate VLAN. In this example, Interfaces 1, 2, 3, and 4 are in VLAN 2 and Interface 5 is in VLAN 3.

   Note: In Alteon, you may configure more than one subnet per VLAN.

   To configure the IP interfaces for this example, enter the following commands from the CLI:

`>> Main# /cfg/l3/if 1`	(Select IP interface 1)
`>> IP Interface 1 # addr 10.10.10.10`	(Assign an IP address to the interface)
`>> IP Interface 1 # vlan 2`	(Assign a VLAN to the interface)
`>> IP Interface 1 # ena`	(Enable IP interface 1)

   Repeat the commands for each of the following interfaces:

   — IF 2—20.10.10.10
   — IF 3—30.10.10.10
   — IF 4—40.10.10.10
   — IF 5—200.1.1.10

2. Define the VLANs. In this configuration, set up two VLANs:
   - One for the outside world—the ports connected to the upstream devices, toward the routers (VLAN 3 in Figure 78 - Active-Standby Configuration, page 522).
   - One for the inside—the ports connected to the downstream devices, toward the servers (VLAN 2 in Figure 78 - Active-Standby Configuration, page 522).

`>> Main# /cfg/l2/vlan vlan 3`	(Select VLAN 3)
`>> vlan 3 # add <port number>`	(Add a port to the VLAN membership)
`>> vlan 3 # ena`	(Enable VLAN 3)

   Repeat this command for the second VLAN:
   - VLAN 3 (interface 5)—Physical ports connected to upstream devices.
   - VLAN 2 (interfaces 1, 2, 3, and 4)—Physical ports connected to downstream devices.
3. Disable Spanning Tree.

`>> Main# /cfg/l2/stg 1`	(Select the STP Group number)
`>> Main# /cfg/l2/stg 1/off`	(Disable STP)
`>> Main# /cfg/l2/stg 1/apply`	(Make your changes active)

4. Enable IP forwarding. IP forwarding is enabled by default. Make sure IP forwarding is enabled if the virtual server IP addresses and real server IP addresses are on different subnets, or if the device is connected to different subnets and those subnets need to communicate through the device. If you are not sure whether to enable IP forwarding, enable it. In this example, the virtual server IP addresses and real server IP addresses are on different subnets, so it should be enabled:

```
>> Main# /cfg/l3/frwd/on
```

To configure SLB

In this procedure, you perform the following:
- Define real servers
- Define real server groups
- Define virtual servers
- Define client and server port states

1. Define the real servers.

   The real server IP addresses are defined and put into four groups, depending on the service they are running. Notice that RIPs 7 and 8 are on routable subnets in order to support passive FTP. For each real server, you must assign a real server number, specify its actual IP address, and enable the real server:

`>> Main# /cfg/slb/real 1`	(Server A is Real Server 1)
`>> Real server 1 # rip 10.10.10.5`	(Assign Server A IP address)
`>> Real server 1 # ena`	(Enable Real Server 1)

Repeat this sequence of commands for the following real servers:

— RIP 2—10.10.10.6/24
— RIP 3—20.10.10.5/24
— RIP 4—20.10.10.6/24
— RIP 5—30.10.10.5/24
— RIP 6—30.10.10.6/24
— RIP 7—200.1.1.5/24
— RIP 8—200.1.1.6/24

2. Define the real server groups, adding the appropriate real servers.

   This combines the three real servers into one service group:

`>> Real server 8 # /cfg/slb/group 1`	(Select Real Server Group 1)
`>> Real server group 1# add 1`	(Add Real Server 1 to Group 1)
`>> Real server group 1# add 2`	(Add Real Server 2 to Group 1)

Repeat this sequence of commands for the following real server groups:

— Group 2—Add RIP 3 and 4
— Group 3—Add RIP 5 and 6
— Group 4—Add RIP 7 and 8

3. Define the virtual servers.

   After defining the virtual server IP addresses and associating them with a real server group number, you must define which IP ports, services, or sockets you want to load balance on each VIP. You can specify the service by either the port number, service name, or socket number:

`>> Real server group 4 # /cfg/slb/virt 1`	(Select virtual server 1)
`>> Virtual server 1 # vip 200.200.200.100`	(Assign a virtual server IP address)
`>> Virtual Server 1 # service 80`	(Assign HTTP service port 80)
`>> Virtual server 1 http Service # group 1`	(Associate virtual port to real group)
`>> Virtual server 1 # ena`	(Enable the virtual server)

Repeat this sequence of commands for the following virtual servers:

— VIP 2—200.200.200.101 load balances HTTPS (Port 443) to Group 2
— VIP 3—200.200.200.102 load balances POP/SMTP (Ports 110/25) to Group 3
— VIP 4—200.200.200.104 load balances FTP (Ports 20/21) to Group 4

4. Define the client and server port states.

   The defined client port state results in the port watching for any frames destined for the VIP and to load balance them if they are destined for a load-balanced service, as well as remapping (NAT) the real server IP address back to the virtual server IP address.

Notes

- The ports connected to the upstream devices (the ones connected to the routers) need to be in the client port state.
- The ports connected to the downstream devices (the ones providing fan out for the servers) need to be in the server port state.

`>> Virtual server 4# /cfg/slb/port 1`	(Select device Port 1)
`>> SLB port A1 # client ena`	(Enable client processing on Port 1)
`>> SLB port A1 # /cfg/slb/port 2`	(Select device Port 2)
`>> SLB port A2 # server ena`	(Enable server processing on Port 2)

To configure virtual router redundancy

In this procedure, you perform the following:

- Define virtual routers
- Configure virtual router priority
- Configure virtual router priority tracking

1. Configure virtual routers 2, 4, 6, and 8.

   These virtual routers have the same IP addresses as the virtual server IP address. Alteon identifies these as virtual service routers (VSRs). In this example, Layer 3 bindings are left in their default configuration, which is disabled:

`>> Virtual server 4 # /cfg/l3/vrrp/vr 2`	(Select Virtual Router 2)
`>> Virtual router 2 # vrid 2`	(Set virtual router ID)
`>> Virtual router 2 # addr 200.200.200.100`	(Assign virtual router IP address)
`>> Virtual router 2 # if 5`	(Assign virtual router interface)
`>> Virtual router 2 # ena`	(Enable Virtual Router 2)

   Repeat this sequence of commands for the following virtual routers:

   — VR 4 - VRID 4 - IF 5 (associate with IP interface 5)—Address 200.200.200.101

   — VR 6 - VRID 6 - IF 5 (associate with IP interface 5)—Address 200.200.200.103

   — VR 8 - VRID 8 - IF 5 (associate with IP interface 5)—Address 200.200.200.104

2. Configure virtual routers 1, 3, 5, and 7.

   These virtual routers act as the default gateways for the servers on each respective subnet. Because these virtual routers are survivable next hop or default gateways, they are called virtual interface routers (VIRs). Configure each virtual router:

   — VR 1 - VRID 1 - IF 1 (associate with IP interface 1)—Address 10.10.10.1

   — VR 3 - VRID 3 - IF 2 (associate with IP interface 2)—Address 20.10.10.1

   — VR 5 - VRID 5 - IF 3 (associate with IP interface 3)—Address 30.10.10.1

   — VR 7 - VRID 7 - IF 4 (associate with IP interface 4)—Address 40.10.10.1

3. Set the renter priority for each virtual router.

   Since you want Alteon 1 to be the master router, you need to bump the default virtual router priorities (which are 100 to 101 on virtual routers 1 through 4) to force Alteon 1 to be the master for these virtual routers:

`>> Virtual server 4 # /cfg/l3/vrrp/vr 1`	(Select Virtual Router 1)
`>> Virtual router 1 # prio 101`	(Set virtual router priority)

Apply this sequence of commands to the following virtual routers, assigning each a priority of 101:

— VR 2—Priority 101
— VR 3—Priority 101
— VR 4—Priority 101

4. Configure priority tracking parameters for each virtual router. For this example, the best parameter to track is Layer 4 ports (l4pts):

```
>> Virtual server 4# /cfg/l3/vrrp/vr 1/track l4pts
```

This command sets the priority tracking parameter for Virtual Router 1, electing the virtual router with the most available ports as the master router. Repeat this command for the following virtual routers:

— VR 2 - Track l4ptsVR 6 - Track l4pts
— VR 3 - Track l4ptsVR 7 - Track l4pts
— VR 4 - Track l4ptsVR 8 - Track l4pts

Configuration for Alteon device 1 is complete.

To configure the second Alteon device

Use this procedure to configure Alteon device 2:

1. Dump the configuration script (text dump) of Alteon 1 using either of the following tools:
   — BBI
      a. You need a serial cable that is a DB-9 male to DB-9 female, straight-through (not a null modem) cable.
      b. Connect the cable from a COM port on your computer to the console port on Alteon 1.
      c. Open HyperTerminal (or the terminal program of your choice) and connect to the device using the following parameters: Baud: 115200, Data Bits: 8, Parity: None, Stop Bits:1, Flow Control: None.
   — HyperTerminal
      a. Only the Baud Rate and Flow Control options need to be changed from the default settings.
      b. Once you connect to the device, start logging your session in HyperTerminal (transfer/capture text).
      c. Save the file as “Customer Name” Alteon 1, then type the following command in the CLI: `/cfg/dump` to dump a script.
      d. Stop logging your session (transfer/capture text/stop).
2. Modify the script by opening the text file that was created and change the following:
   a. Delete anything above **Script Start**.
   b. Delete the two lines directly below **Script Start**. These two lines identify the device from which the dump was taken and the date and time. If these two lines are left in, it will confuse Alteon 2 when you dump in the file.
   c. Change the last octet in all the IP interfaces from .10 to .11. Find this line in the file:
      /cfg/l3/if 1/addr 10.10.10.10
   d. Delete the **0** and replace it with a 1. Be sure to do this for all the IP interfaces or duplicate IP addresses will be present in the network.

e. Change the virtual router priorities. Virtual routers 1 through 4 need to have their priority set to 100 from 101, and virtual routers 5 through 7 need to have their priorities set to 101 from 100. You can find this in the line `/cfg/l3/vrrp/vr 1/vrid 1/if 1/prio 101`.

f. Scroll to the bottom of the text file and delete anything past **Script End**.

g. Save the changes to the text file as **<Customer Name>** Alteon 2.

3. Move your serial cable to the console port on the second device. Any configuration on it needs to be deleted by resetting it to factory settings, using the following command:

```
>> Main# /boot/conf factory/reset
```

You can tell if the device is at factory default when you log in because it will prompt you if you want to use the step-by-step configuration process. When it does, respond **No**.

Note: After completing the setup you cannot proceed further without configuring the ports. To configure ports enter **y,** or enter **n** to ignore.

4. In HyperTerminal, go to transfer/send text file and send the Alteon 2 text file. The configuration dumps into the device. Type **apply**, then **save**. When you can type characters in the terminal session again, reboot the device (`/boot/reset`).

Hot Standby Redundancy

This configuration is based on proprietary Alteon extensions to VRRP.

In a hot standby configuration, the Spanning Tree Protocol (STP) is not needed to eliminate bridge loops. This speeds up failover when an Alteon fails. The standby Alteon blocks all ports configured as standby ports, while the master Alteon enables these same ports. Consequently, on a given Alteon, all virtual routers are either master or backup—they cannot change state individually.

In a hot standby configuration, two or more Alteons provide redundancy for each other. One Alteon is elected master and actively processes Layer 4 traffic. The other Alteons (the backups) assume the master role if the master fails. The backups may forward Layer 2 and Layer 3 traffic, as appropriate.

Note: Alteon considers a trunk port failed and changes its priority only when all the ports in the trunk are down.

Note: When a hot standby port is not part of a VLAN assigned to a vADC, Alteon does track the port for VRRP priority.

This section describes the following topics:

Switch-Centric Virtual Router Group

Hot standby requires all virtual routers on an Alteon to fail over together as a group. For more information about the switch-based virtual router groups, see Switch-Based VRRP Groups, page 514.

When enabled, the switch-centric virtual router group aggregates all virtual routers as a single entity. All virtual routers fail over as a group, and cannot fail over individually. As members of a group, all virtual routers on Alteon (and therefore Alteon itself), are either the master or a backup.

To enable a switch-based virtual router group

```
>> Main# /cfg/l3/vrrp/group ena
```

Layer 4 Port States

When Alteon changes from master to backup, it does not process any traffic destined to the virtual server routers configured on that Alteon. However, Layer 4 processing is still enabled. An Alteon that has become the backup can still process traffic addressed to its virtual server IP address. Filtering is also still functional.

Each VRRP advertisement can include up to 1024 addresses, and is therefore is not limited to a single virtual router IP address. A VRRP advertisement packet that contains all virtual routers are advertised in the same packet, thus conserving processing and buffering resources.

Hot Standby and Interswitch Port States

The hot standby configuration includes two Layer 4 port states: hotstan (hot standby), and intersw (interswitch).

Notes

- A port cannot be configured to support both hot standby and interswitch links.
- The interswitch setting for hot standby is *not* the same as Cisco's ISL protocol.

To set the state for links that attach to the standby Alteon

```
>> Main# /cfg/slb/port x/hotstan
```

When the hotstan option is enabled and all hot standby ports have a link, the virtual router group's priority is incremented by the **track other virtual routers** value. This lets the Alteons fail over when a hot standby port loses a link. Other enabled tracking features have an effect only when all hot standby ports on Alteon have a link. The default virtual routers tracking value is 2 seconds. This is an automatic process that cannot be turned off.

Note: All ports with hot standby enabled must be connected to another Alteon.

To set the state for links that are used by VRRP to deliver updates

```
>> Main# /cfg/slb/port <port number> /intersw ena
Enter VLAN Number: 202
Current inter-switch processing: disabled
Enter new inter-switch processing [d/e]: e
```

The hot standby Alteon listens to the master's VRRP updates. After an interval has expired without receiving a update, the backup takes over. The forwarding states of hot standby ports are controlled much like the forwarding states of the hot standby (hotstan) approach. Enabling hot standby on a port allows the hot standby algorithm to control the forwarding state of the port. If an Alteon is the master, the forwarding states of the hot standby ports are enabled. If an Alteon is a backup, the hot standby ports are blocked from forwarding or receiving traffic.

Note: The VRRP hot standby approach does *not* support single-link failover. If one hot standby port loses a link, the entire Alteon must become the master to eliminate loss of connectivity.

The forwarding states of non-hot standby ports are not controlled via the hot standby algorithm, allowing the additional ports to provide added port density. The client ports on both Alteons should be able to process or forward traffic to the master.

The interswitch port state is only a place holder. It forces you to configure an interswitch link when hot standby is globally enabled and prohibits the interswitch link from also being a hot standby link for VRRP advertisements. These advertisements must be able to reach the backup Alteon.

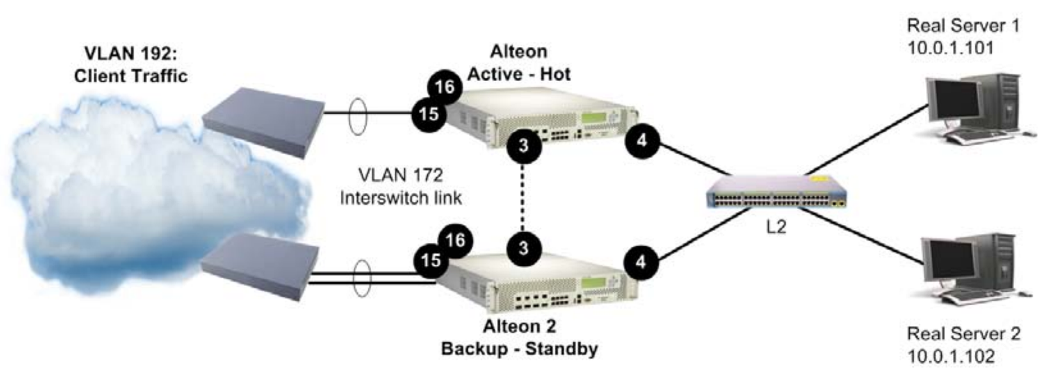
Hot Standby Configuration

A hot standby configuration enables all processes to fail over to a backup if any type of failure occurs. The primary application for hot standby redundancy is to avoid bridging loops when using the Spanning Tree Protocol (STP), IEEE 802.1d. VRRP-based hot standby supports the default spanning tree only. It does not support multiple spanning trees.

Figure 80 - Hot Standby Configuration, page 537 illustrates a classic network topology designed to use redundancy. This topology contains bridging loops that require using STP. In the typical network, STP failover time is 45-50 seconds, much longer than the typical failover rate using VRRP only.

Note: To use hot standby redundancy, Alteon peers must have an equal number of ports.

Figure 80: Hot Standby Configuration



The key to hot standby is that the interswitch link does not participate in STP, so there are no loops in the topology (see Hot Standby Configuration, page 537). You have to disable STP globally to use the VRRR hot standby scenario.

Note: In a host-standby configuration, Radware recommends that you locate the server and the interswitch link on the same VLAN. This ensures that services are in the UP state for both the master as well as the backup.

The following procedures refer to Figure 80 - Hot Standby Configuration, page 537:

- To configure Layer 2 and Layer 3 parameters on Alteon 1, page 538
- To configure virtual router redundancy, page 539
- To prepare a configuration script for Alteon 2, page 540
- To synchronize Layer 4 parameters from Alteon 1 to Alteon 2, page 542

To configure Layer 2 and Layer 3 parameters on Alteon 1

This procedure assumes you have already configured SLB parameters.

1. On Alteon 1, configure the external ports into their respective VLANs as shown in Figure 80 - Hot Standby Configuration, page 537.

`>> Main# cfg/port 3/tag ena`	(Enable VLAN tagging for Port 3)
`>> Main# /cfg/l2/vlan 172`	(VLAN 172 is for th einterswitch link and for server traffic)
`>> VLAN 172# ena`	(Enable the VLAN)
`>> VLAN 172# name ISL_and_servers`	(Name VLAN 172 for the interswitch link and for server traffic)
`>> VLAN 172# add 3`	(Add Port 3 to VLAN 172)
`>> VLAN 172# add 4`	(Add Port 4 to VLAN 172)
`>> Main# /cfg/l2/vlan 192`	(VLAN 192 is for client traffic)
`>> VLAN 192# ena`	(Enable the VLAN)
`>> VLAN 192# name clients`	(Name VLAN 192 for client traffic)
`>> VLAN 192# add 15`	(Add Port 15 to VLAN 192)
`>> VLAN 192# add 16`	(Add Port 16 to VLAN 192)

2. Trunk the ports you configured for the client VLAN.

`>> Main # /cfg/l2/trunk 1`	(Select Trunk Group 1)
`>> Trunk group 1# ena`	(Enable the Trunk Group)
`>> Trunk group 1# add 15 16`	(Add the external ports to the trunk)

3. Turn off spanning tree.

`>> Main # /cfg/l2/stg 1/off`	(Disable STG group)
`>> Spanning Tree Group 1# apply`	(Make your changes active)
`>> Spanning Tree Group 1# save`	(Save for restore after reboot)

4. Configure the IP addresses for each VLAN.

`>> Main # /cfg/l3/`
`>> IP Interface 1# /cfg/l3/if 2`
`>> IP Interface 2# ena`
`>> IP Interface 2# addr 192.168.1.251`
`>> IP Interface 2# vlan 192`
	(IP Interface 2: client traffic)

`>> IP Interface 2# /cfg/l3/if 3`
`>> IP Interface 3# ena`
`>> IP Interface 3# addr 172.16.2.251`
`>> IP Interface 3# vlan 172`
	(IP Interface 3: interswitch link and servers)

To configure virtual router redundancy

1. Configure virtual routers for the server, client, and interswitch link traffic.

`>> Main #/cfg/l3/vrrp/vr 2`
`>> VRRP Virtual Router 2# ena`
`>> VRRP Virtual Router 2# vrid 2`
`>> VRRP Virtual Router 2# if 2`
`>> VRRP Virtual Router 2# addr 192.168.1.250`
	(Virtual router for client)

`>> Main # /cfg/l3/vrrp/vr 3`
`>> VRRP Virtual Router 3# ena`
`>> VRRP Virtual Router 3# vrid 3`
`>> VRRP Virtual Router 3# if 3`
`>> VRRP Virtual Router 3# addr 172.16.2.250`
	(Virtual router for interswitch link and servers)

2. From the *VRRP* menu, enable VRRP group mode and hot standby on all connected ports except the interswitch link.

`>> Main # /cfg/l3/vrrp/on`	(Enable VRRP)
`>> Virtual Router Redundancy Protocol# hotstan ena`	(Enable hot standby)
`>> Virtual Router Redundancy Protocol# group ena`	(Enable VR group)
`>> Virtual Router Redundancy Protocol# if 3`	(Define the tracking interface)
`>> VRRP Virtual Router Group# apply`	(Make your changes active)
`>> VRRP Virtual Router Group# save`	(Save for restore after reboot)

Note: When you enable hot standby for a vrgroup, the currently set priority for the vrgroup is increased by 2.

3. Set VRRP tracking for the ports.

   If a link on any of the connected ports goes down, the VRRP priority of Alteon decreases and the backup takes over as the master.

`>> Main # /cfg/l3/vrrp`	
`>> VRRP Virtual Router Group# vrid 254`	
`>> VRRP Virtual Router Group# prio 101`	(Set priority at 101 for the master)
`>> VRRP Virtual Router Group# track/ports ena`	

4. Setup the peer Alteon to receive synchronization.

   Make sure to disable synchronization of VRRP priorities. The peer Alteon assumes its own priority based on the VRRP election process and should not acquire the VRRP priority from the master's configuration. If you want to configure real servers for VRRP hot standby, also enable synchronization of real server configuration.

`>> Main # /cfg/slb/sync/prios d`	(Do not synchronize VRRP priorities to the peer Alteon)
`>> Config Synchronization# peer 1/ena`	(Enable synchronization to the peer Alteon)
`>> Peer Switch 1# addr 172.16.2.252`	(Set IP address of Alteon 1)

5. From the *SLB* menu, enable a hot standby link on the Layer 4 ports, then enable the interswitch link on the crosslink.

```
>> Main # /cfg/slb/port 2
>> SLB port 2# hotstan ena
>> SLB port 2# /cfg/slb/port 3
>> SLB port 3# intersw
   Enter VLAN Number: 172
   Current inter-switch processing: disabled
   Enter new inter-switch processing [d/e]: e
>> SLB port 3# /cfg/slb/port 4
>> SLB port 3# hotstan ena
```

6. Apply and save changes to the configuration.

```
>> SLB port 3# apply
>> SLB port 3# save
```

To prepare a configuration script for Alteon 2

This procedure dumps the configuration script (text dump) from Alteon 1. This configuration will be modified and loaded onto Alteon 2.

1. Dump Alteon configuration using the following command:

```
>> Main # /cfg/dump
```

   A script is dumped out.

2. Copy and paste the entire contents of the script to a text file. The first and last lines of the file must be written as shown in the following example:

```
script start "Alteon Application Switch" 4  /**** DO NOT EDIT THIS LINE!


/* Configuration dump taken  1:52:02 Fri Feb  6, 2004
/* Version 0.0.0,  Base MAC address 00:0e:40:32:7c:00
:
:
script end  /**** DO NOT EDIT THIS LINE!

>> Configuration#
```

3. Edit the text file that you just created as follows:
   - Change all the IP interface addresses for Alteon 2. Otherwise, this results in duplicate IP addresses in the network. In this example, change the last octet in all the IP interfaces from .251 to .252.

```
/c/l3/if 1
addr 10.0.1.252
```

   - Change the synchronization peer Alteon to use the IP address of Alteon 1. In this example, change the last octet from .252 (denoting Alteon 2), to .251 (Alteon 1).

```
/c/slb/sync/peer 1
ena
addr 172.16.2.251
```

   - Change the virtual router priority from 100 to 101. This indicates that Alteon 2 is the backup for now.

```
/c/l3/vrrp/group
:
prio 101
```

4. Save the changes to the text file as ***Customer-Name*****_backup_config** and load it onto a TFTP server.
5. Begin a Telnet session for the second Alteon. Delete any existing configuration on it by resetting it to factory settings, using the following command:

```
>> Main # /boot/conf factory/reset
```

   A confirmation message displays. Do one of the following:
   - Enter **y** to save changes and restart.
   - Enter **n** to ignore changes and cancel restart.

   Note: If you are prompted to use the step-by-step configuration process when you log in, Alteon is set to the factory defaults. When prompted, enter **No**.

6. From the CLI, download the configuration script into Alteon from the TFTP server using the following command:

```
>> Main # /cfg/gtcfg <tftp-server-addr> <cfg-filename>
```

To synchronize Layer 4 parameters from Alteon 1 to Alteon 2

1. On Alteon 1, synchronize the VRRP, SLB, real server, and filter settings to the other Alteon (same ports).

   Note: Alteons that are peers of one another should have an equal number of ports.

```
>> Main# /oper/slb/sync
NOTE: Use the "/c/slb/sync" menu to configure omitting sections of the
configuration.
Synchronizing VRRP, VR priorities, FILT, PORT, BWM, SLB and Acceleration configs
to 172.16.2.252
Confirm synchronizing the configuration to 172.16.2.252 [y/n]:y
```

2. On Alteon 2, apply and save the configuration changes.

Tracking Virtual Routers

The tracking configuration largely depends on user preferences and network environment. The following example is based on the configuration shown in Figure 78 - Active-Standby Configuration, page 522 and summarized below:

- Alteon 1 is the master router upon initialization.
- If Alteon 1 is the master and it has one fewer active servers than Alteon 2, then Alteon 1 remains the master.

  This behavior is preferred because running one server down is less disruptive than bringing a new master online and severing all active connections in the process.
- If Alteon 1 is the master and it has two or more active servers fewer than Alteon 2, then Alteon 2 becomes the master.
- If Alteon 2 is the master, it remains the master even if servers are restored on Alteon 1 such that it has one fewer or an equal number of servers.
- If Alteon 2 is the master and it has one active server fewer than Alteon 1, then Alteon 1 becomes the master.

To configure tracking virtual routers

1. Set the priority for Alteon 1 to the default value of 100.
2. Set the priority for Alteon 2 to 96.
3. On both Alteons, enable tracking based on the number of virtual routers in master mode on Alteon and set the value to 5.

4. On both Alteons, enable tracking based on the number of healthy real servers behind the VIP address. The VIP address is the same as the IP address of the virtual server router on Alteon. Set the value to 6.

   Initially, Alteon 1 has a priority of 100 (base value), plus 5 (initially it is the master), plus 24 (4 active real servers multiplied by 6, per real server), resulting in 129.

   Alteon 2 has a priority of 96 (base value), plus 24 (4 active real servers multiplied by 6, per real server), resulting in 120.

   If a server attached to Alteon 1 fails, then the priority for Alteon 1 is reduced by 6, resulting in 123. Since 123 is greater than 120 (the priority for Alteon 2), Alteon 1 remains the master.

   If a second server attached to Alteon 1 fails, then the priority for Alteon 1 is reduced by 6 more, resulting in 117. Since 117 is less than 120 (the priority for Alteon 2), then Alteon 2 becomes the master. At this point, the priority for Alteon 1 will falls by 5 more, and the priority for Alteon 2 rises by 5, because the Alteons are tracking how many masters they are running. As a result, the priority for Alteon 1 results in 112, and the priority for Alteon 2 results in 125.

   When both servers are restored to Alteon 1, its priority rises by 12 (2 healthy real servers multiplied by 6, per healthy server), resulting in 124. Because 124 is less than 125, Alteon 2 remains the master.

   If, at this point, a server fails on Alteon 2, its priority falls by 6, resulting in 119. Because 119 is less than 124, Alteon 1 becomes the master. Its priority results in 129, since it is now the master, while the priority for Alteon 2 drops by 5 more, resulting in 114.

Tip: There is no shortcut to setting tracking parameters. Your goals must first be set and the outcomes of various configurations and scenarios analyzed to find settings that meet your goals.

Service-Based Virtual Router Groups

Service-based virtual router groups can be used for failover in either an active-active or active-standby configuration.

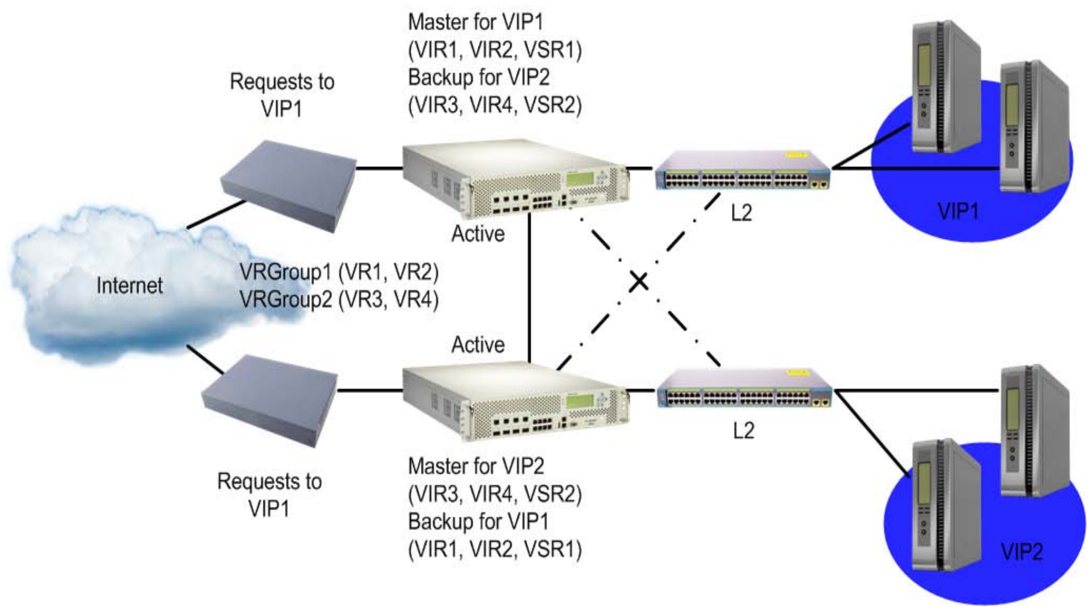
Figure 81 - Service-Based Virtual Router Groups in and Active-Standby Configuration, page 544, illustrates two customers sharing the same VRRP devices configured in active-standby configuration for VIP 1 and 2. Virtual routers 1, 2, 3, and 4 are defined on both Alteons as follows:

- Virtual Routers 1 and 3 are virtual interface routers—they use the IP interface addresses.
- Virtual Routers 2 and 4 are virtual service routers—they use the virtual server IP addresses.

Virtual Router 1 on the master forwards the packets sent to the IP addresses associated with the virtual router, and answers ARP requests for these IP addresses. The virtual router backup assumes forwarding responsibility for a virtual router should the current master fail.

Virtual routers 1 and 2 are members of Vrgroup 1, and virtual routers 3 and 4 are members of Vrgroup 2.

Figure 81: Service-Based Virtual Router Groups in and Active-Standby Configuration



Example Service-Based Virtual Router Groups Configuration

In this example, if the interface or link to the real server fails for the vrgroup 1 on Alteon 1, then all the virtual routers in vrgroup 1 change to the backup state. At the same time, all virtual routers in vrgroup 1 on Alteon 2 change to the master state. Meanwhile, the virtual routers in vrgroup 2 continue to operate via Alteon 1.

The separate real server groups provide segregation of services for each customer, so neither customer's traffic interferes with the others. To implement this active-standby example with tracking of service-based virtual router groups, do the following:

1. Define the IP interfaces.

   Alteon needs an IP interface for each subnet to which it is connected so it can communicate with devices attached to it. To configure the IP interfaces for this example, enter the following commands from the CLI:

`>> Main# /cfg/l3/if 1`	(Select IP interface 1)
`>> IP Interface 1 # addr 200.200.200.1`	(Assign IP address for the interface)
`>> IP Interface 1 # ena`	(Enable IP interface 1)

   Repeat the commands for the following interfaces:

   — IF 2: 205.178.13.2

   — IF 3: 200.200.200.3

   — IF 4: 205.178.13.4

2. Define all filters required for your network configuration. Filters may be configured on one Alteon and synchronized with settings on the other Alteon.

3. Configure all required SLB parameters on Alteon 1.

   Required Layer 4 parameters include two virtual server IP addresses, two groups, and four real servers.

`>> Main# /cfg/slb/real 1/`	(Configure real servers)

`>> Real server 1# rip 10.10.10.101`
`>> Real server 1# /cfg/slb/real 2/rip 10.10.10.102`
`>> Real server 2# /cfg/slb/real 3/rip 10.10.10.103`
`>> Real server 3# /cfg/slb/real 4/rip 10.10.10.104`	

`>> Real server 3# /cfg/slb/group 1`	(Select Real Server Group 1)
`>> Real server group 1# add 1`	(Add Real Server 1 to Group 1)
`>> Real server group 1# add 2`	(Add Real Server 2 to Group 1)
`>> Main # /cfg/slb/virt 1/vip 205.178.13.226`	(Configure Virtual Server IP 1)
`>> Virtual server 1# ena`	(Enable the virtual server)
`>> Virtual server 1# service http`	(Select the *HTTP Service Port* menu)
`>> Virtual server 1 http Service# group 1`	(Associate the virtual port to real group)
`>> Main # /cfg/slb/group 2`	
`>> Real server group 1# add 3`	(Add Real Server 1 to Group 1)
`>> Real server group 1# add 4`	(Add Real Server 2 to Group 1)
`>> Main # /cfg/slb/virt 1/vip 205.178.13.300`	
`>> Virtual server 1# ena`	(Enable the virtual server)
`>> Virtual server 1# service http`	(Select the HTTP service menu)
`>> Virtual server 1 http Service# group 2`	(Associate the virtual port to real group)

4. Configure virtual interface routers 1 and 3, and make sure to disable sharing.

   These virtual routers are assigned the same IP address as the IP interfaces configured in step 1, resulting in Alteon recognizing these as virtual interface routers (VIRs). In this example, Layer 3 bindings are left in their default configuration (disabled). For an active-standby configuration, sharing is disabled.

`>> Main # /cfg/l3/vrrp/vr 1`	(Select Virtual Router 1)
`>> VRRP Virtual Router 1# vrid 1`	(Set the virtual router ID)
`>> VRRP Virtual Router 1# addr 200.200.200.100`	(Assign the VR IP address)
`>> VRRP Virtual Router 1# if 1`	(Assign the virtual router interface)
`>> VRRP Virtual Router 1# share dis`	(Disable sharing of interfaces)
`>> VRRP Virtual Router 1# ena`	(Enable Virtual Router 1)
`>> Main # /cfg/l3/vrrp/vr 3`	(Select Virtual Router 3)
`>> VRRP Virtual Router 3# vrid 3`	(Set the virtual router ID)
`>> VRRP Virtual Router 3# addr 200.200.200.103`	(Assign VR IP address)
`>> VRRP Virtual Router 3# if 3`	(Assign the virtual router interface)
`>> VRRP Virtual Router 3# share dis`	(Disable sharing of interfaces)
`>> VRRP Virtual Router 3# ena`	(Enable Virtual Router 3)

5. Configure virtual server routers 2 and 4.

   These virtual routers have the same IP addresses as the virtual server IP address. This is how Alteon recognizes that these are virtual service routers (VSRs).

   For an active-standby configuration, sharing is disabled.

`>> Main # /cfg/l3/vrrp/vr 2`	(Select Virtual Router 2)
`>> VRRP Virtual Router 2# vrid 2`	(Set the virtual router ID)
`>> VRRP Virtual Router 2# addr 205.178.13.226`	(Assign VR IP address)
`>> VRRP Virtual Router 2# if 2`	(Assign virtual router interface)
`>> VRRP Virtual Router 2# share dis`	(Disable sharing of interfaces)
`>> VRRP Virtual Router 2# ena`	(Enable Virtual Router 2)
`>> Main # /cfg/l3/vrrp/vr 4`	(Select Virtual Router 4)
`>> VRRP Virtual Router 4# vrid 4`	(Set virtual router ID)
`>> VRRP Virtual Router 4# addr 205.178.13.300`	(Assign VR IP address)
`>> VRRP Virtual Router 4# if 4`	(Assign virtual router interface)
`>> VRRP Virtual Router 4# share dis`	(Disable sharing of interfaces)
`>> VRRP Virtual Router 4# ena`	(Enable virtual router 4)

6. Add virtual routers 1 and 2 to the Vrgroup 1.

`>> Main# /cfg/l3/vrrp/vrgroup 1`	
`>> VRRP Virtual Router Vrgroup 1# add 1`	(Add virtual router 1—the VIR)
`>> VRRP Virtual Router Vrgroup 1# add 2`	(Add virtual router 2—the VSR)
`>> VRRP Virtual Router Vrgroup 1# ena`	
`>> VRRP Virtual Router Vrgroup 1# track`	(Select the *Priority Tracking* menu)
`>> VRRP Vrgroup 1 Priority Tracking# ports ena`	(Track on physical ports)

7. Add virtual routers 3 and 4 to switch-based Vrgroup 2.

`>> Main# /cfg/l3/vrrp/vrgroup 2`	
`>> VRRP Virtual Router Vrgroup 2# add 3`	(Add Virtual Router 1)
`>> VRRP Virtual Router Vrgroup 2# add 4`	(Add Virtual Router 2)
`>> VRRP Virtual Router Vrgroup 2# ena`	
`>> VRRP Virtual Router Vrgroup 2# track`	(Select the *Priority Tracking* menu)
`>> VRRP Vrgroup 2 Priority Tracking# l4ports ena`	(Track on layer 4 ports)

8. Disable synchronizing of priority on Alteon 1.

   The priorities should not be synchronized between the two Alteons. The priority for each vrgroup will change based on the tracking parameters configured in step 6 and step 7.

```
>> Main # /cfg/slb/sync prios disable
```

9. Synchronize the SLB and VRRP configurations from Alteon 1 with Alteon 2.

   Use the **/oper/slb/sync** command (see Configuring VRRP Peers for Synchronization, page 569).

IPv6 VRRP Configuration Examples

This section contains three IPv6 VRRP configuration examples covering hot standby, active-standby, and active-active configurations. For background information on these VRRP configuration types, see Failover Methods and Configurations, page 521.

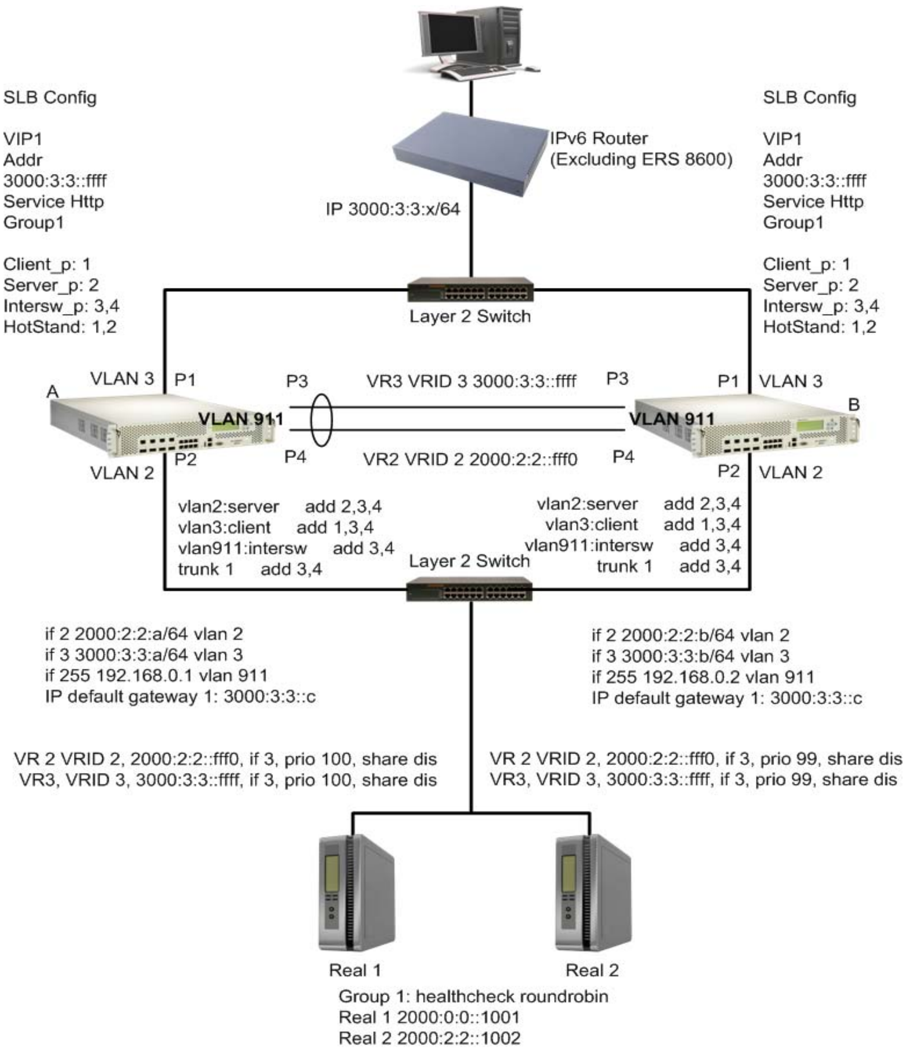
This section describes the following topics:

Hot Standby Configuration

This configuration example illustrates a hot standby configuration between two Alteons. The following are considerations for a IPv6 hot standby configuration:

- For Layer 2 (port and VLAN) configurations
  - Each VLAN must be configured per interface.
  - Client-side and server-side VLANs must also be members in an interswitch-link (ISL) port, or have the ISL interface as the VRRP group interface.
  - In a one-arm setup, the VR group can be configured using the ISL link. For more information, see One Arm SLB Configuration, page 199.
- For spanning tree group configurations
  - The Spanning Tree Protocol must be turned off.
- Layer 3 interface and VRRP configurations
  - In this example, tracking is performed by Layer 2 ports so that any failures on the master result in a successful failover to the backup.
- Server Load Balancing
  - Ports connected to the Alteon peer directly, or via a Layer 2 device, must have hot standby (**/cfg/slb/port hot**) enabled. ISL and other ports should not have hot standby enabled.

Figure 82: Example IPv6 Hot Standby Configuration



To configure an IPv6 hot standby configuration

1. Alteon A configuration:

   — Layer 2 (port and VLAN) and Layer 3 (interface) configuration:

```
/cfg/port 1
       pvid 3
/cfg/port 2
       pvid 2
/cfg/port 3
       tagged ena
       pvid 911
/cfg/port 4
       tagged ena
       pvid 911
/cfg/l2/vlan 2
       ena
       name "server"
       learn ena
       def  2,3,4
/cfg/l2/vlan 3
       ena
       name "client"
       learn ena
       def 1,3,4
/cfg/l2/vlan 911
       ena
       name "intersw"
       learn ena
       def 3,4
cfg/l2/trunk 1
       ena
       add 3
       add 4
```

— Spanning tree group configuration:

```
/cfg/l2/stg 1/off
/cfg/l2/stg 1/add 1 2 3 911
```

— Interface configuration:

```
/cfg/l3/if 2
        ena
        ipver v6
        addr 2000:2:2:0:0:0:0:a
        mask 96
        vlan 2
/cfg/l3/if 3
        ena
        ipver v6
        addr 3000:3:3:0:0:0:0:a
        mask 96
        vlan 3
/cfg/l3/if 254
        ena
        ipver v4
        addr 192.168.0.1
        mask 255.255.255.0
        broad 192.168.0.255
        vlan 911
```

— Default gateway configuration:

```
/cfg/l3/gw 1
       ena
       ipver v6
       addr 3000:3:3:0:0:0:0:c
```

— VRRP configuration:

```
/cfg/l3/vrrp/on
/cfg/l3/vrrp/vr 2
        ena
        ipver v6
        vrid 2
        if 2
        addr 2000:2:2:0:0:0:0:fff0
        share dis
/cfg/l3/vrrp/vr 3
        ena
        ipver v6
        vrid 3
        if 3
        addr 3000:3:3:0:0:0:0:ffff
        share dis
/cfg/l3/vrrp/group
        ena
        ipver v6
        vrid 254
        if 2
        share dis
        track ports
```

— General SLB configuration:

```
/cfg/slb
        on
/cfg/slb/adv
        direct ena
```

— IPv6 real server configuration:

```
/cfg/slb/real 1 ena
        ipver v6
        rip 2000:2:2:0:0:0:0:1001
/cfg/slb/real 2
        ena
        ipver v6
        rip 2000:2:2:0:0:0:0:1002
```

— IPv6 Real Server Group 1 configuration:

```
/cfg/slb/group 1
       ipver v6
       add 1
       add 2
```

— IPv6 VIP 1 HTTP service configuration:

```
/cfg/slb/virt 1
       ena
       ipver v6
       vip 3000:3:3:0:0:0:0:ffff
       vname "v6http"
/cfg/slb/virt 1/service http
       group 1
```

— Layer 4 port configuration:

```
/cfg/slb/port 1
       client ena
       hotstan en
/cfg/slb/port 2
       server ena
       hotstan en/
cfg/slb/port 3
       intersw ena
       vlan 400
/cfg/slb/port 4
       intersw ena
       vlan 400
```

— Synchronization configuration:

```
/cfg/slb/sync
         prios d
/cfg/slb/sync/peer 1
         ena
         addr 192.168.0.2
```

2. Alteon B configuration:

— Layer 2 (port and VLAN) and Layer 3 (interface) configuration:

```
/cfg/port 1
        pvid 3
/cfg/port 2
        pvid 2/
cfg/port 3
        tagged ena
        pvid 911
/cfg/port 4
        tagged ena
        pvid 911
/cfg/l2/vlan 2
        ena
        name "server"
        learn ena
        def  2,3,4
/cfg/l2/vlan 3
        ena
        name "client"
        learn ena
        def 1,3,4
/cfg/l2/vlan 911
        ena
        name "intersw"
        learn ena
        def 3,4
/cfg/l2/trunk 1
        ena
        add 3
        add 4
```

— Spanning tree group configuration:

```
/cfg/l2/stg 1/off
/cfg/l2/stg 1/add 1 2 3 911
```

— Interface configuration:

```
/cfg/l3/if 2
        ena
        ipver v6
        addr 2000:2:2:0:0:0:0:b
        mask 96
        vlan 2
/cfg/l3/if 3
        ena
        ipver v6
        addr 3000:3:3:0:0:0:0:b
        mask 96
        vlan 3
/cfg/l3/if 255
        ena
        ipver v4
        addr 192.168.0.2
        mask 255.255.255.0
        broad 192.168.0.255
        vlan 911
```

— Default gateway configuration:

```
/cfg/l3/gw 1
        ena
        ipver v6
        addr 3000:3:3:0:0:0:0:c
```

— VRRP configuration:

```
/cfg/l3/vrrp/on
/cfg/l3/vrrp/vr 2
        ena
        ipver v6
        vrid 2
        if 2
        addr 2000:2:2:0:0:0:0:fff0
        share dis/
cfg/l3/vrrp/vr 3
        ena
        ipver v6
        vrid 3
        if 3
        addr 3000:3:3:0:0:0:0:ffff
        share dis
/cfg/l3/vrrp/group
        ena
        ipver v6
        vrid 254
        if 2
        share dis
        track ports
```

— General SLB configuration:

```
/cfg/slb
        on
/cfg/slb/adv
        direct ena
```

— IPv6 real server configuration:

```
/cfg/slb/real 1
       ena
       ipver v6
       rip 2000:2:2:0:0:0:0:1001
/cfg/slb/real 2
       ena
       ipver v6
       rip 2000:2:2:0:0:0:0:1002
```

— IPv6 Real Server Group 1 configuration:

```
/cfg/slb/group 1
       ipver v6
       add 1
       add 2
```

— IPv6 VIP 1 HTTP service configuration:

```
/cfg/slb/virt 1
       ena
       ipver v6
       vip 3000:3:3:0:0:0:0:ffff
       vname "v6http"
/cfg/slb/virt 1/service http
       group 1
```

— Layer 4 ports configuration:

```
/cfg/slb/port 1
       client ena
       hotstan en/
cfg/slb/port 2
       server ena
       hotstan en
/cfg/slb/port 3
       intersw ena
       vlan 400
/cfg/slb/port 4
       intersw ena
       vlan 400
```

— Synchronization configuration:

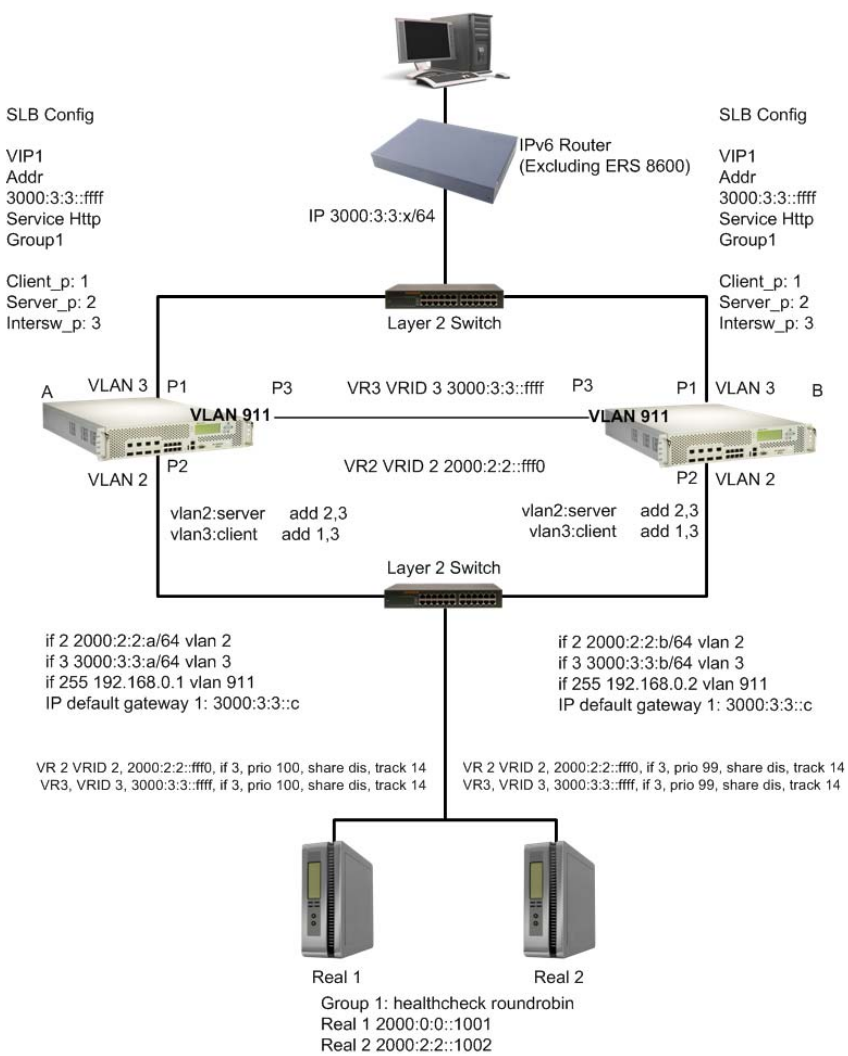
```
/cfg/slb/sync
       prios d
/cfg/slb/sync/peer 1
       ena
       addr 192.168.0.1
```

Active-Standby Configuration

Figure 83 - Active-Standby Configuration Example, page 555 illustrates an active-standby configuration between two Alteon units. The following are considerations for a IPv6 active-standby configuration:

- Layer 2 (port and VLAN) configuration:
  - — Each VLAN must be configured per interface.
- Layer 3 interface and VRRP configuration:
  - — In this example, tracking is performed by Layer 4 ports so that the two virtual routers fail over when one of the master virtual routers declares itself as the backup.

Figure 83: Active-Standby Configuration Example



To configure an IPv6 active-standby configuration

1. Alteon A configuration:
   — Layer 2 (port and VLAN) and Layer 3 (Interface) configuration:

```
/cfg/port 1
        pvid 3
/cfg/port 2
        pvid 2
/cfg/port 3
        pvid 911
/cfg/l2/vlan 2
        ena
        name "server"
        learn ena
        def  2
/cfg/l2/vlan 3
        ena
        name "client"
        learn ena
        def 1
/cfg/l2/vlan 911
        ena
        name "intersw"
        learn ena
        def 3
```

   — Interface configuration:

```
/cfg/l3/if 2
        ena
        ipver v6
        addr 2000:2:2:0:0:0:0:a
        mask 96
        vlan 2
/cfg/l3/if 3
        ena
        ipver v6
        addr 3000:3:3:0:0:0:0:a
        mask 96
        vlan 3
/cfg/l3/if 254
        ena
        ipver v4
        addr 192.168.0.1
        mask 255.255.255.0
        broad 192.168.0.255
        vlan 911
```

— Default gateway configuration:

```
/cfg/l3/gw 1
        ena
        ipver v6
        addr 3000:3:3:0:0:0:0:c
```

— VRRP configuration:

```
/cfg/l3/vrrp/on
/cfg/l3/vrrp/vr 2
        ena
        ipver v6
        vrid 2
        if 2
        addr 2000:2:2:0:0:0:0:fff0
        share dis
        track
        l4pts ena
/cfg/l3/vrrp/vr 3
        ena
        ipver v6
        vrid 3
        if 3
        addr 3000:3:3:0:0:0:0:ffff
        share dis
        track
        l4pts ena
```

— General SLB configuration:

```
/cfg/slb
        on
/cfg/slb/adv
        direct ena
```

— IPv6 real server configuration:

```
/cfg/slb/real 1
        ena
        ipver v6
        rip 2000:2:2:0:0:0:0:1001
/cfg/slb/real 2
        ena
        ipver v6
        rip 2000:2:2:0:0:0:0:1002
```

— IPv6 Real Server Group 1 configuration:

```
/cfg/slb/group 1
        ipver v6
        add 1
        add 2
```

— IPv6 VIP 1 HTTP Service configuration:

```
/cfg/slb/virt 1
        ena
        ipver v6
        vip 3000:3:3:0:0:0:0:ffff
        vname "v6http"
/cfg/slb/virt 1/service http
        group 1
```

— Layer 4 ports configuration:

```
/cfg/slb/port 1
        client ena
/cfg/slb/port 2
        server ena
```

— Synchronization configuration:

```
/cfg/slb/sync
        prios d
/cfg/slb/sync/peer 1
        ena
        addr 192.168.0.2
```

2. Alteon-B configuration:

— Layer 2 (port and VLAN) and Layer 3 (interface) configuration:

```
/cfg/port 1
       pvid 3
/cfg/port 2
       pvid 2
/cfg/port 3
       pvid 911
/cfg/l2/vlan 2
       ena
       name "server"
       learn ena
       def  2
/cfg/l2/vlan 3
       ena
       name "client"
       learn ena
       def 1
/cfg/l2/vlan 911
       ena
       name "intersw"
       learn ena
       def 3
```

— Interface configuration:

```
/cfg/l3/if 2
       ena
       ipver v6
       addr 2000:2:2:0:0:0:0:b
       mask 96
       vlan 2
/cfg/l3/if 3
       ena
       ipver v6
       addr 3000:3:3:0:0:0:0:b
       mask 96
       vlan 3
/cfg/l3/if 255
       ena
       ipver v4
       addr 192.168.0.2
       mask 255.255.255.0
       broad 192.168.0.255
       vlan 911
```

— Default gateway configuration:

```
/cfg/l3/gw 1
        ena
        ipver v6
        addr 3000:3:3:0:0:0:0:c
```

— VRRP configuration:

```
/cfg/l3/vrrp/on
/cfg/l3/vrrp/vr 2
        ena
        ipver v6
        vrid 2
        if 2
        addr 2000:2:2:0:0:0:0:fff0
        share dis
        track
        l4pts ena
/cfg/l3/vrrp/vr 3
        ena
        ipver v6
        vrid 3
        if 3
        addr 3000:3:3:0:0:0:0:ffff
        share dis
        track
        l4pts ena
```

— General SLB configuration:

```
/cfg/slb
        on
/cfg/slb/adv
direct ena
```

— IPv6 real server configuration:

```
/cfg/slb/real 1
        ena
        ipver v6
        rip 2000:2:2:0:0:0:0:1001
/cfg/slb/real 2
        ena
        ipver v6
        rip 2000:2:2:0:0:0:0:1002
```

— IPv6 Real Server Group 1 configuration:

```
/cfg/slb/group 1
         ipver v6
         add 1
         add 2
```

— IPv6 VIP 1 HTTP service configuration:

```
/cfg/slb/virt 1
         ena
         ipver v6
         vip 3000:3:3:0:0:0:0:ffff
         vname "v6http"
/cfg/slb/virt 1/service http
         group 1
```

— Layer 4 ports configuration:

```
/cfg/slb/port 1
         client ena
/cfg/slb/port 2
         server ena
```

— Synchronization configuration:

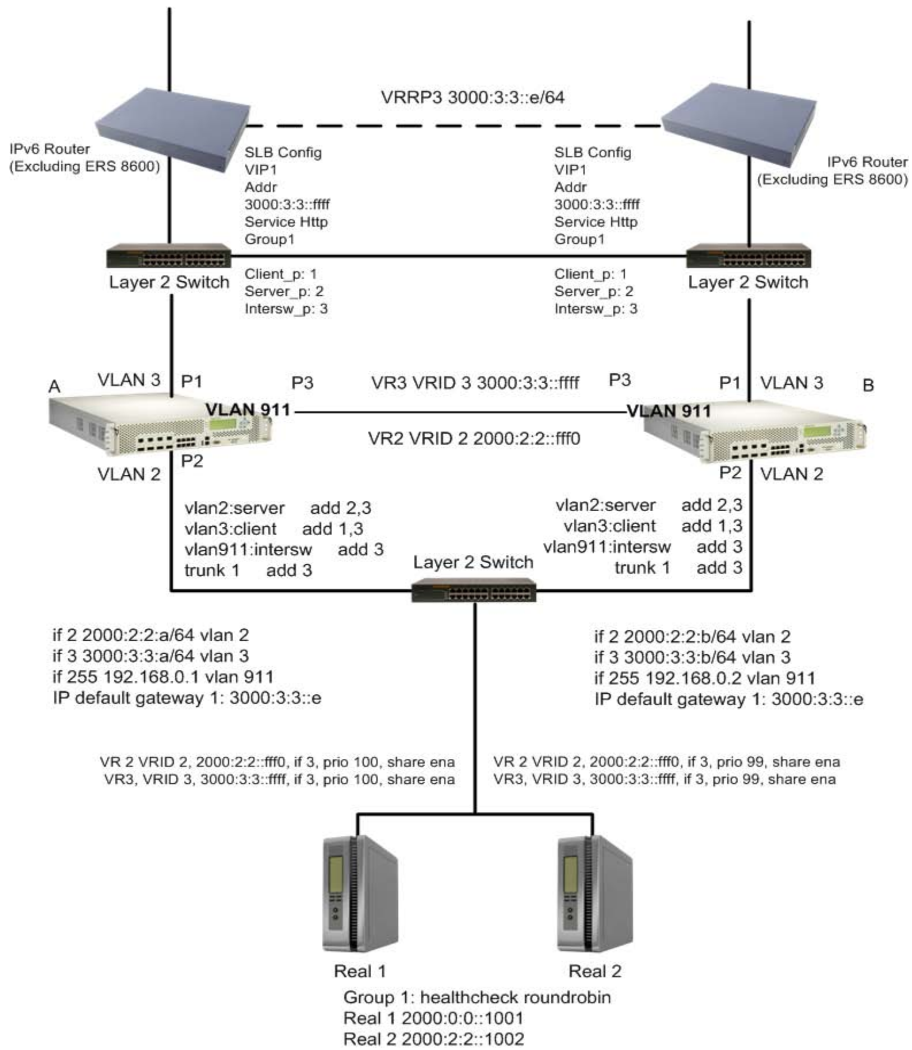
```
/cfg/slb/sync
         prios d
/cfg/slb/sync/peer 1
         ena
         addr 192.168.0.1
```

Active-Active Configuration

Figure 84 - Active-Active Configuration Example, page 561 illustrates an active-active configuration between two Alteons. The following are considerations for a IPv6 active-active configuration:

1. Layer 2 (port and VLAN) configuration:
   — Each VLAN must be configured per interface.
2. Layer 3 interface and VRRP configuration:
   — In this example, tracking is performed by Layer 4 ports so that the two virtual routers fail over when one of the master virtual routers declare itself as the backup.

Figure 84: Active-Active Configuration Example



To configure an IPv6 active-active configuration

1. Alteon A configuration:
   — Layer 2 (port and VLAN) and Layer 3 (interface) configuration.

```
/cfg/port 1
        pvid 3
/cfg/port 2
        pvid 2
/cfg/port 3
        pvid 911
/cfg/l2/vlan 2
        ena
        name "server"
        learn ena
        def  2
/cfg/l2/vlan 3
        ena
        name "client"
        learn ena
        def 1
/cfg/l2/vlan 911
        ena
        name "intersw"
        learn ena
        def 3
```

   — Interface configuration:

```
/cfg/l3/if 2
        ena
        ipver v6
        addr 2000:2:2:0:0:0:0:a
        mask 96
        vlan 2
/cfg/l3/if 3
        ena
        ipver v6
        addr 3000:3:3:0:0:0:0:a
        mask 96
        vlan 3
/cfg/l3/if 254
        ena
        ipver v4
        addr 192.168.0.1
        mask 255.255.255.0
        broad 192.168.0.255
        vlan 911
```

— Default gateway configuration:

```
/cfg/l3/gw 1
       ena
       ipver v6
       addr 3000:3:3:0:0:0:0:c
```

— VRRP configuration:

```
/cfg/l3/vrrp/on
/cfg/l3/vrrp/vr 2
        ena
        ipver v6
        vrid 2
        if 2
        addr 2000:2:2:0:0:0:0:fff0
        share en
        track
        l4pts ena
/cfg/l3/vrrp/vr 3
        ena
        ipver v6
        vrid 3
        if 3
        addr 3000:3:3:0:0:0:0:ffff
        share en
        track
        l4pts ena
```

— General SLB configuration:

```
/cfg/slb
        on
```

— IPv6 real server configuration:

```
/cfg/slb/real 1
        ena
        ipver v6
        rip 2000:2:2:0:0:0:0:1001
/cfg/slb/real 2
        ena
        ipver v6
        rip 2000:2:2:0:0:0:0:1002
```

— IPv6 Real Server Group 1 configuration:

```
/cfg/slb/group 1
        ipver v6
        add 1
        add 2
```

— IPv6 VIP 1 HTTP service configuration:

```
/cfg/slb/virt 1
        ena
        ipver v6
        vip 3000:3:3:0:0:0:0:ffff
        vname "v6http"
/cfg/slb/virt 1/service http
        group 1
```

— Layer 4 ports configuration:

```
/cfg/slb/port 1
        client ena
        hotstan en
/cfg/slb/port 2
        server ena
        hotstan en
/cfg/slb/port 3
        intersw ena
       vlan 400
/cfg/slb/port 4
        intersw ena
       vlan 400
```

— Synchronization configuration:

```
/cfg/slb/sync
        prios d
/cfg/slb/sync/peer 1
        ena
        addr 192.168.0.2
```

2. Alteon B configuration:

— Layer 2 (port and VLAN) and Layer 3 (interface) configuration:

```
/cfg/port 1
        pvid 3
/cfg/port 2
        pvid 2
/cfg/port 3
        pvid 911
/cfg/l2/vlan 2
        ena
        name "server"
        learn ena
        def  2
/cfg/l2/vlan 3
        ena
        name "client"
        learn ena
        def 1
/cfg/l2/vlan 911
        ena
        name "intersw"
        learn ena
        def 3
```

— Interface configuration:

```
/cfg/l3/if 2
        ena
        ipver v6
        addr 2000:2:2:0:0:0:0:b
        mask 96
        vlan 2
/cfg/l3/if 3
        ena
        ipver v6
        addr 3000:3:3:0:0:0:0:b
        mask 96
        vlan 3
/cfg/l3/if 255
        ena
        ipver v4
        addr 192.168.0.2
        mask 255.255.255.0
        broad 192.168.0.255
        vlan 911
```

— Default gateway configuration:

```
/cfg/l3/gw 1
        ena
        ipver v6
        addr 3000:3:3:0:0:0:0:c
```

— VRRP configuration:

```
/cfg/l3/vrrp/on
/cfg/l3/vrrp/vr 2
        ena
        ipver v6
        vrid 2
        if 2
        addr 2000:2:2:0:0:0:0:fff0
        share en
        track
        l4pts en
/cfg/l3/vrrp/vr 3
        ena
        ipver v6
        vrid 3
        if 3
        addr 3000:3:3:0:0:0:0:ffff
        share en
        track
        sl4pts en
```

— General SLB configuration:

```
/cfg/slb
        on
```

— IPv6 real server configuration:

```
/cfg/slb/real 1
        ena
        ipver v6
        rip 2000:2:2:0:0:0:0:1001
/cfg/slb/real 2
        ena
        ipver v6
        rip 2000:2:2:0:0:0:0:1002
```

— IPv6 Real Server Group 1 configuration:

```
/cfg/slb/group 1
        ipver v6
        add 1
        add 2
```

— IPv6 VIP 1 HTTP service configuration:

```
/cfg/slb/virt 1
        ena
        ipver v6
        vip 3000:3:3:0:0:0:0:ffff
        vname "v6http"
/cfg/slb/virt 1/service http
        group 1
```

— Layer 4 ports configuration:

```
/cfg/slb/port 1
        client ena
        hotstan en
/cfg/slb/port 2
        server ena
        hotstan en
/cfg/slb/port 3
        intersw ena
        vlan 400
/cfg/slb/port 4
        intersw ena
        vlan 400
```

— Synchronization configuration:

```
/cfg/slb/sync
        prios d
/cfg/slb/sync/peer 1
        ena
        addr 192.168.0.1
```

Virtual Router Deployment Considerations

Review the issues described in this section to prevent network problems when deploying virtual routers:

- Mixing Active-Standby and Active-Active Virtual Routers, page 567
- Eliminating Loops with STP and VLANs, page 568
- Assigning VRRP Virtual Router ID, page 569
- Configuring VRRP Peers for Synchronization, page 569
- Synchronizing Active/Active Failover, page 570

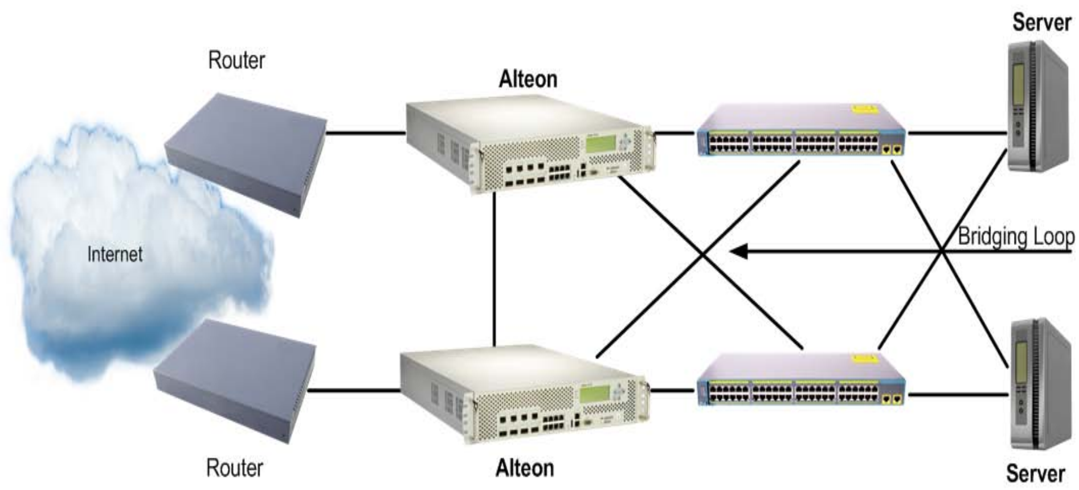
Mixing Active-Standby and Active-Active Virtual Routers

If your network environment can support sharing, enable it for all virtual routers in the LAN. If not, use active-standby for all virtual routers. Do not mix active-active and active-standby virtual routers in a LAN. Mixed configurations may result in unexpected operational characteristics, and is not recommended.

Eliminating Loops with STP and VLANs

Active-active configurations can introduce loops into complex LAN topologies, as illustrated in Figure 85 - Loops in an Active-Active Configuration, page 568:

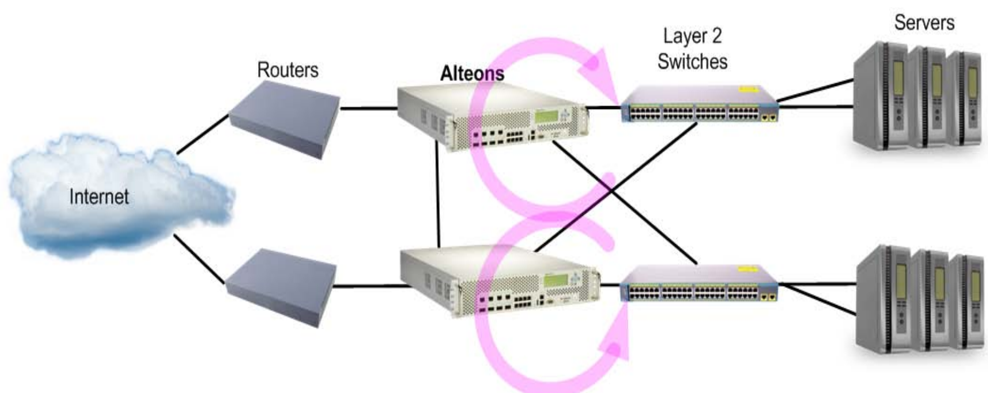
Figure 85: Loops in an Active-Active Configuration



Using Spanning Tree Protocol to Eliminate Loops

VRRP generally requires Spanning Tree Protocol (STP) to be enabled in order to resolve bridge loops that usually occur in cross-redundant topologies. In Figure 86 - STP Resolving Cross-Redundancy Loops, page 568, a number of loops are wired into the topology. STP resolves loops by blocking ports where looping is detected.

Figure 86: STP Resolving Cross-Redundancy Loops

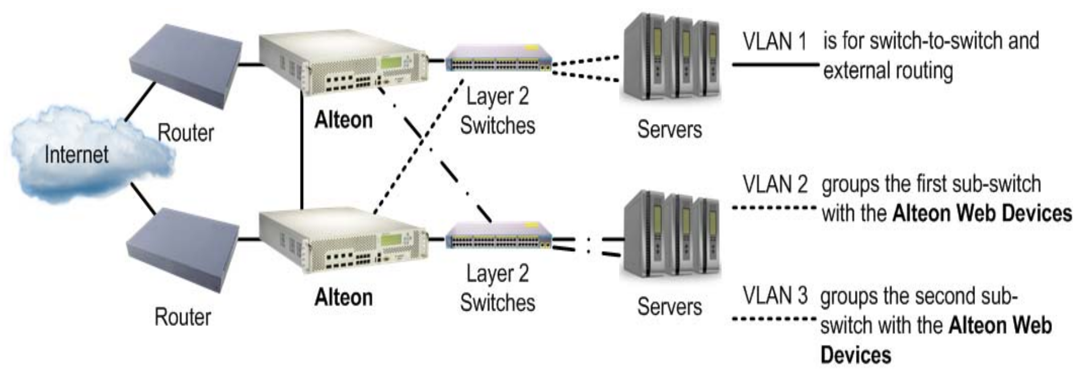


One drawback to using STP with VRRP is the failover response time. STP could take as long as 45 seconds to re-establish alternate routes after an Alteon or link failure.

Using VLANs to Eliminate Loops

When using VRRP, you can decrease failover response time by using VLANs instead of STP to separate traffic into non-looping broadcast domains, as shown in Figure 87 - Using VLANs to Create Non-Looping Topologies, page 569:

Figure 87: Using VLANs to Create Non-Looping Topologies



This topology allows STP to be disabled. On the Alteons, IP routing allows traffic to cross VLAN boundaries. The servers use the Alteons as default gateways. For port failure, traffic is rerouted to the alternate path within one health check interval (configurable between 1 and 60 seconds, with a default of 2 seconds).

Assigning VRRP Virtual Router ID

During the software upgrade process, VRRP virtual router IDs is assigned if failover is enabled. When configuring virtual routers at any point after upgrade, virtual router ID numbers (**using the /cfg/l3/vrrp/vr #/vrid** command) must be assigned. The virtual router ID may be configured as any number between 1 and 255.

Configuring VRRP Peers for Synchronization

The final step in configuring a high availability solution includes the addition of synchronization options to simplify the manual configuration. Synchronization configuration options refine what is synchronized, to what, and to disable synchronizing certain configurations. These options include proxy IP addresses, Layer 4 port configuration, filter configuration, and virtual router priorities.

The *peer* menu (**cfg/slb/sync/peer**) enables you to configure the IP addresses of the Alteons that should be synchronized. This provides added synchronization validation but does not require users to enter the IP address of the redundant Alteon for each synchronization.

Each VRRP-capable device is autonomous. Alteons in a virtual router need not be identically configured. As a result, configurations cannot be synchronized.

You can synchronize a configuration from one VRRP-capable device to another using the **/oper/slb/sync** command. All server load balancing, port configurations, filter configurations, and VRRP parameters can be synchronized using this command.

Note: Before you synchronize the configuration between two Alteons, a peer must be configured on each Alteon. Alteons that are synchronized must use the same administrator password.

To configure two Alteons as peers to each other

1. From Alteon 1, configure Alteon 2 as a peer and specify its IP address:

`>> Main # /cfg/slb/sync`	(Select the *Synchronization* menu)
`>> Config Synchronization # peer 1`	(Select a peer)
`>> Peer Switch 1 # addr <IP address>`	(Assign the Alteon 2 IP address)
`>> Peer Switch 1 # enable`	(Enable peer Alteon)

2. From Alteon 2, configure Alteon 1 as a peer and specify its IP address:

`>> Main # /cfg/slb/sync`	(Select the *Synchronization* menu)
`>> Config Synchronization # peer 2`	(Select a peer)
`>> Peer Switch 2 # addr <IP address>`	(Assign Alteon 1 IP address)
`>> Peer Switch 2 # enable`	(Enable peer Alteon)

Note: Port specific parameters, such as which filters are applied and enabled on which ports, are part of what is pushed by the **/oper/slb/sync** command. Therefore, if you use the **/oper/slb/sync** command is used, Radware recommends that the hardware configurations and network connections of all Alteons in the virtual router be identical. This means that each Alteon should be the same model, have the same line cards in the same slots (if modular), and have the same ports connected to the same external network devices. Otherwise, unexpected results may occur when the **/oper/slb/sync** command attempts to configure a non-existent port or applies an inappropriate configuration to a port.

You can define whether or not to synchronize certificate repository components by enabling **cfg/slb/sync/certs**. By default, this option is disabled. When certificate repository synchronization is enabled, you are required to set a passphrase to be used during the configuration synchronization for the encryption of private keys (**cfg/slb/sync/passphrs**). The same passphrase should be set manually by the administrator in all VRRP members for private key decryption.

To encrypt or decrypt certificate private keys during configuration synchronization, the same passphrase must be set at all peer devices.

Synchronizing Active/Active Failover

With VRRP and active-active failover, the primary and secondary Alteons do not require identical configurations and port topology. Each Alteon can be configured individually with a different port topology, SLB, and filters. If you would rather force two active-active Alteons to use identical settings, you can synchronize their configuration using the following command:

```
>> Main # /oper/slb/sync
```

The **sync** command copies the following settings to Alteon at the specified IP interface address:

- VRRP settings (including priority)
- SLB settings (including port settings)
- Filter settings (including filter port settings)
- Proxy IP settings

If you use the **sync** command, you should check the configuration on the target Alteon to ensure that the settings are correct.

Note: When using both VRRP and GSLB, you must change the **/cfg/sys/access/wport** (Browser-Based Interface port) value of the target Alteon (the Alteon to which you are synchronizing) to a port other than port 80 before VRRP synchronization begins.

Stateful Failover of Persistent Sessions

Alteon includes stateful failover of persistent session states, including:

- Client IP
- SSL session state
- HTTP cookie state
- Layer 4 persistent
- FTP session state
- WAP session state

For more information about the supported persistence types, see Persistence, page 583.

Stateful failover lets you mirror Layer 7 and Layer 4 persistent transactional states on the Alteon peers.

Note: Stateful failover is not supported in active-active mode. Also, stateful failover does not synchronize all sessions, except persistent sessions (SSL session ID persistence and cookie-based persistence). If a service fails in the middle of a connection, the current session is discarded, but the new connection binds the session request correctly to the same real server.

To provide stateful failover, the state of the connection and session table must be shared between Alteon in high availability configurations. If the Virtual Matrix Architecture (VMA) is enabled, all URL and cookie-parsing information is stored in the session table on the last port number on Alteon. Sharing this information between Alteons is necessary to ensure the persistent session goes back to the same server.

Stateful failover only ensures that the client's request returns to the same server based on the persistent session entries being shared by the master and the slave Alteon. The TCP session information, however, is not shared.

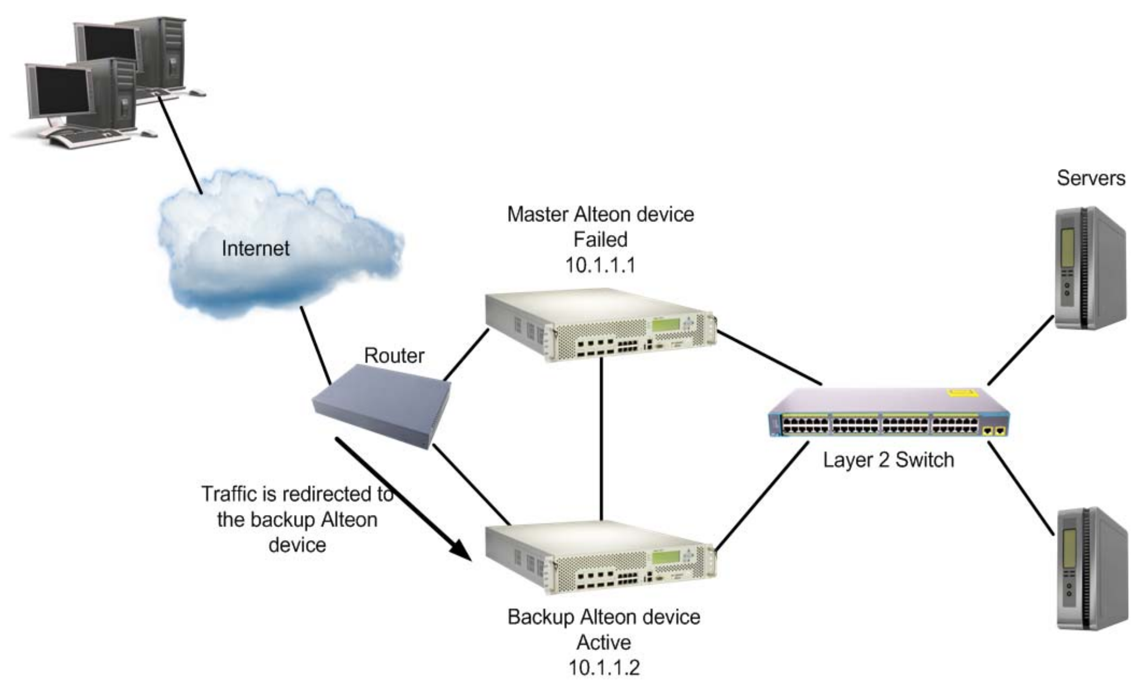
What Happens When Alteon Fails

Assume that the user performing an e-commerce transaction has selected a number of items and placed them in the shopping cart. The user has already established a persistent session on the top server, as shown in Figure 88 - Stateful Failover Example when the Master Alteon Fails, page 572. The user then clicks **Submit** to purchase the items. At this time, the active Alteon fails. With stateful failover, the following sequence of events occurs:

1. The backup becomes active.
2. The incoming request is redirected to the backup.
3. When the user clicks **Submit** again, the request is forwarded to the correct server.

Even though the master has failed, the stateful failover feature prevents the client from having to re-establish a secure session. The server that stores the secure session now returns a response to the client via the backup.

Figure 88: Stateful Failover Example when the Master Alteon Fails



To configure stateful failover

This procedure is based on Figure 88 - Stateful Failover Example when the Master Alteon Fails, page 572.

1. On the master:
   a. Enable stateful failover.

```
>> Main # /cfg/slb/sync/state ena
```

   b. Set the update interval. The default is 30.

```
>> Main # /cfg/slb/sync/update 10
```

c. Configure the backup as a peer and specify its IP address.

`>> Main # /cfg/slb/sync`	
`>> Config Synchronization # peer 1`	(Select a peer)
`>> Peer Switch 1 # addr 10.1.1.2`	(Assign backup Alteon IP address)
`>> Peer Switch 1 # enable`	(Enable peer Alteon)

2. On the backup Alteon:
   a. Enable stateful failover.

```
>> Main # /cfg/slb/sync/state ena
```

   b. Set the update interval. The default is 30.

```
>> Main # /cfg/slb/sync/update 10
```

   The update does not have to be the same for both Alteons. Stateful failover supports up to two peers. Repeat the steps mentioned above to enable stateful failover on all the peers.

   c. Configure the master as a peer and specify its IP address.

`>> Main # /cfg/slb/sync`	
`>> Config Synchronization # peer 2`	(Select a peer)
`>> Peer Switch 2 # addr 10.1.1.1`	(Assign master Alteon IP address)
`>> Peer Switch 2 # enable`	(Enable peer Alteon)

Viewing Statistics on Persistent Port Sessions

You can view statistics on persistent port sessions using the **/stats/slb/ssl** command. To determine which Alteon is the master and which is the backup, use the **/info/l3/vrrp** command. The column on the far right displays Alteon status.

- If Alteon is a master:

```
>> Main # /info/l3/vrrp
VRRP information:
   1: vrid   1, 172.21.16.187,if  4, renter, prio 109, master, server
   3: vrid   3, 192.168.1.30, if  2, renter, prio 109, master
   5: vrid   5, 172.21.16.10, if  4, renter, prio 109, master
```

- If Alteon is a backup:

```
>> Main # /info/l3/vrrp
VRRP information:
   1: vrid   1, 172.21.16.187,if  1, renter, prio 104, backup, server
   3: vrid   3, 192.168.1.30, if  3, renter, prio 104, backup
   5: vrid   5, 172.21.16.10, if  1, renter, prio 104, backup
```

Service-Based Session Failover

This section describes the following session failover-related topics:

Session Failover for Hot Standby Configurations

Alteon supports the failover of a session based on a service. The Network Access, Authentication, and Accounting protocol (NAAP) is used as the communication mechanism between the master and backup. Since NAAP is a Layer 2 protocol, the Alteons need to be connected directly over the interswitch link.

You must also enable the Spanning Tree protocol in hot standby configurations.

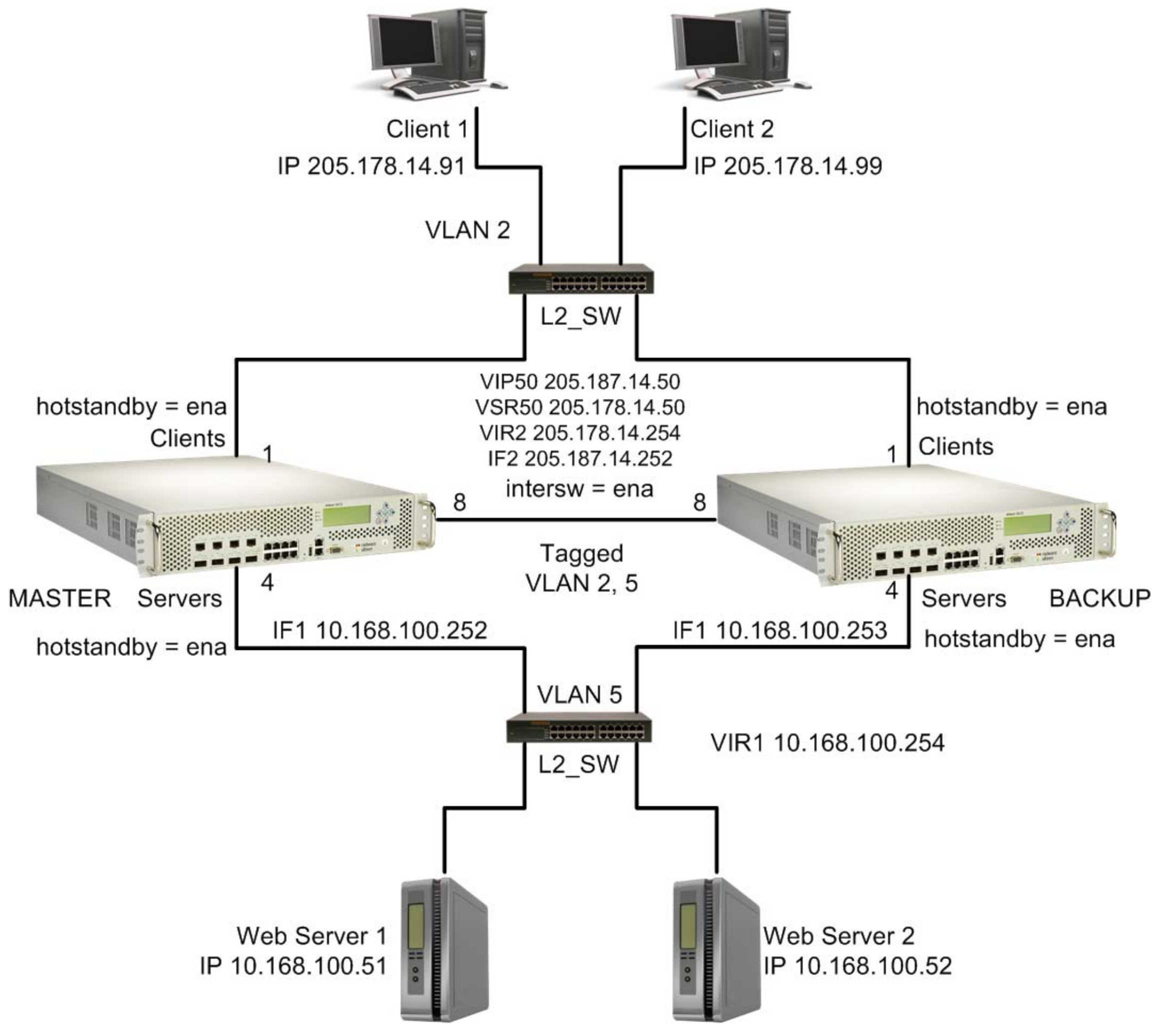
In ADC-VX, enabling and disabling the interswitch link for session mirroring is available per vADC. As vADCs are completely independent instances, they are neither dependant on the physical attributes of Alteon nor on the physical connectivity available. The vADC administrator decides whether to enable the interswitch link feature. To use the feature, the vADC administrator selects one of the assigned VLAN IDs and enables the functionality as follows:

```
>> SLB Port 1# intersw
Current inter-switch processing: disabled
Enter new inter-switch processing [d/e]: e
Enter new ISL VLAN: 200
```

Similar to a standalone Alteon, the vADCs must share a broadcast domain in order to send the session updates the neighboring vADC.

Figure 89 - Service-Based Session Failover for Hot Standby Configurations, page 575 illustrates a service-based session failover network topology:

Figure 89: Service-Based Session Failover for Hot Standby Configurations



When a new session is created on the master, the session entry is sent to the backup using NAAP. The backup creates the session and sets the age to a maximum age. This prevents the session from aging out on the backup and prevents frequent updates between the master and backup. When the session is updated or deleted on the master, the session on the backup is also updated or deleted.

When the master becomes a backup due to reboot or link failure, it sends a session sync message to the new master. The new master mirrors all the sessions that need to be mirrored to the backup. To avoid performance impact, all sessions are not sent at the same time. A timer routine is used to mirror the sessions. In each round, a maximum of 1024 sessions are sent to the backup.

Operations During Session Mirroring on Reboot

The following are the operations that take place during session mirroring on reboot:

- A session synchronize message is sent to the peer while booting.
- The timer routine sends the sync request based on following conditions:
  - — VRRP state is **backup**.
  - — The interswitch port is in forwarding or enable mode.

After the request is sent, the timer routine is disabled with reset flag **sfo_sync_req_flg**.

- On the receipt of the sync message, the master invokes a response timer routine with one second time interval.

  After the master sends all the sessions to the backup, the total number of synced sessions are logged with a syslog.

Service-based Session Failover (Session Mirroring) Limitations and Recommendations

Limitations

- The feature is supported only in group-based active-standby and hot standby VRRP configurations.
- The following filters and protocols are supported:
  - SIP
  - FTP (when **ftpp** is disabled under `cfg/slb/virt/service`)
  - Layer 4 SLB with delayed binding
  - NAT filters
- The following filters and protocols are not supported:
  - Active-active VRRP
  - RTSP
  - Layer 7 SLB
  - `Allow/deny/redir` filters

Recommendations

- Ensure a direct interswitch link between the master and backup, as NAAP packets cannot be routed.
- The same priorities on all Alteons.
- Preemption enabled.
- Layer 4 switch port tracking enabled (`/cfg/l3/vrrp/vr/track/l4pts`).
- A **holdoff** interval of at least 3 seconds (`/cfg/l3/vrrp/holdoff`).

Service-Based Session Failover Commands

The following are commands that support this feature.

To enable and disable mirroring for a service

```
>> Main # /cfg/slb/virt <Virtual Server> /service <Service Number> / mirror
{enable|disable}
```

To enable and disable mirroring for a filter

```
>> Main # /cfg/slb/filt <Filter Number> /adv/mirror {enable|disable}
```

To display mirroring statistics

```
>> Main # /stats/slb/mirror
```

Note: Due to the difference in the amount of physical memory and session capacity between different Alteon models, not all sessions can be synchronized. Session synchronization works correctly if the same model Alteons are used in VRRP HA topologies.

Automate Session Mirroring

Session mirroring can be automated to synchronize sessions from master to backup at the configured time and frequency. Time is specified as hour/minute/date/month/year, and frequency is specified as daily, weekly, or monthly. Based on this, configuration mirroring is done at every configured period.

For example, if the frequency is configured as weekly (**w**), mirroring is performed on the date it is configured once a week. If frequency is configured as monthly (**m**), mirroring is done on the date it is configured once a month.

To set time and frequency for session mirroring

1. Enter the following command:

```
/cfg/l3/vrrp/autosmir
```

2. Specify the time and frequency for session mirroring.

   The following is the sample output:

```
 >> Main# /cfg/l3/vrrp/autosmir
Enter hour in 24-hour format [21]: 20
Enter minutes [10]:     5
Enter the date [08]:    02
Enter the month [10]:   09
Enter the Year [08]:    08
Enter the interval [d/w/m]:     m
```

Session Failover for Active-Standby Configurations

Although group-based VRRP is supported for active-standby configurations, it is not required in order to enable session failover for active-standby configurations. However, group-based VRRP *is* required in order to enable session failover for hot standby configurations.

A VLAN dedicated to interswitch traffic (VLAN 6) is required to avoid network loops. The interswitch link does not require IP interfaces in the VLAN.

You do not need to enable the Spanning Tree protocol in active-standby configurations.

To enable session failover for active-standby configurations

1. Connect an interswitch link.
2. Enable interswitch on connected ports.

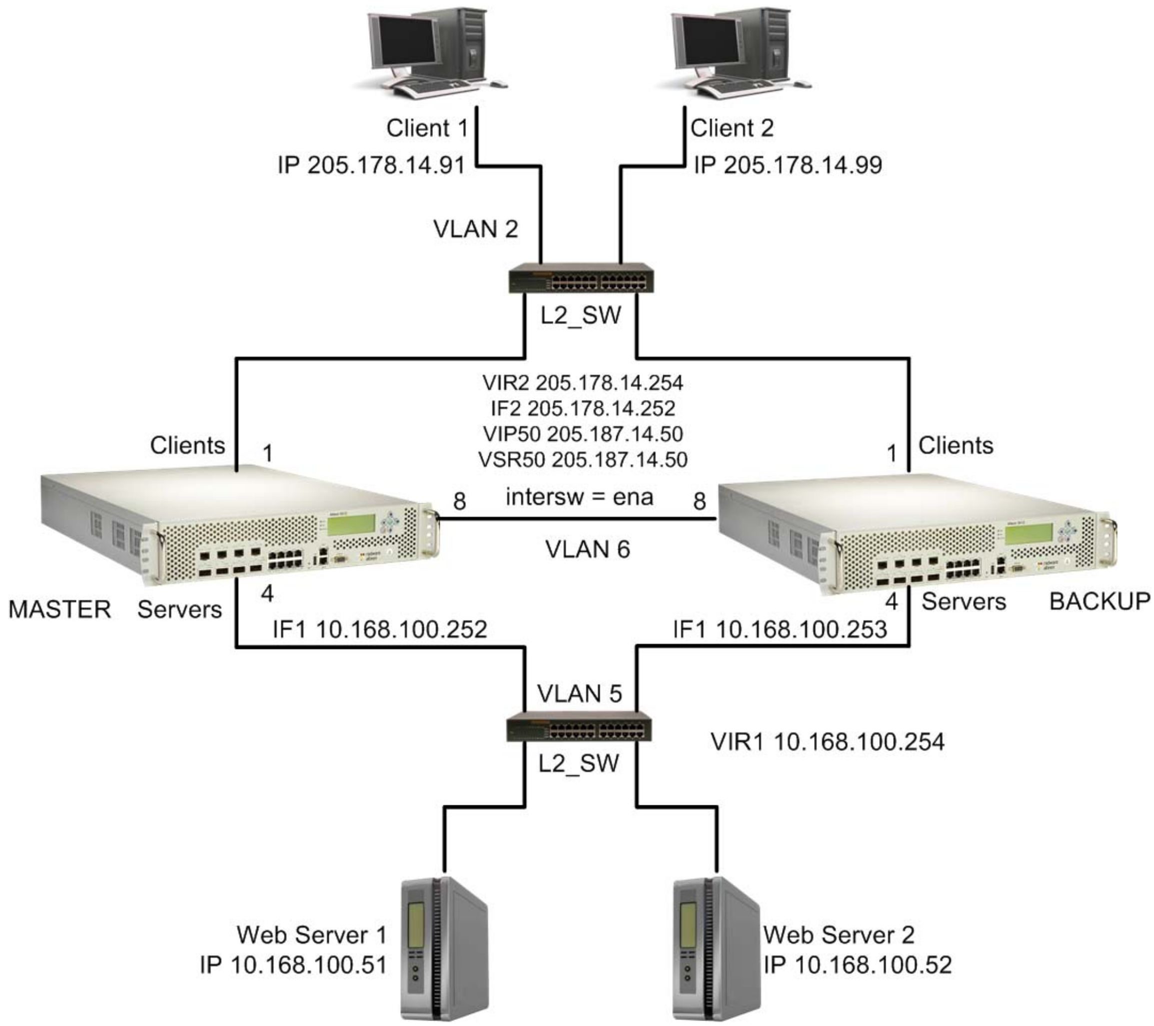
```
>> /cfg/slb/port x/intersw
Enter VLAN Number: 202
Current inter-switch processing: disabled
Enter new inter-switch processing [d/e]: e
```

3. Enable service based session failover.

```
>>/cfg/slb/virt x/service 80/mirror ena
```

Figure 90 - Session Failover for Active-Standby Configurations, page 579 illustrates a service-based session failover network topology for active-standby configurations.

Figure 90: Session Failover for Active-Standby Configurations

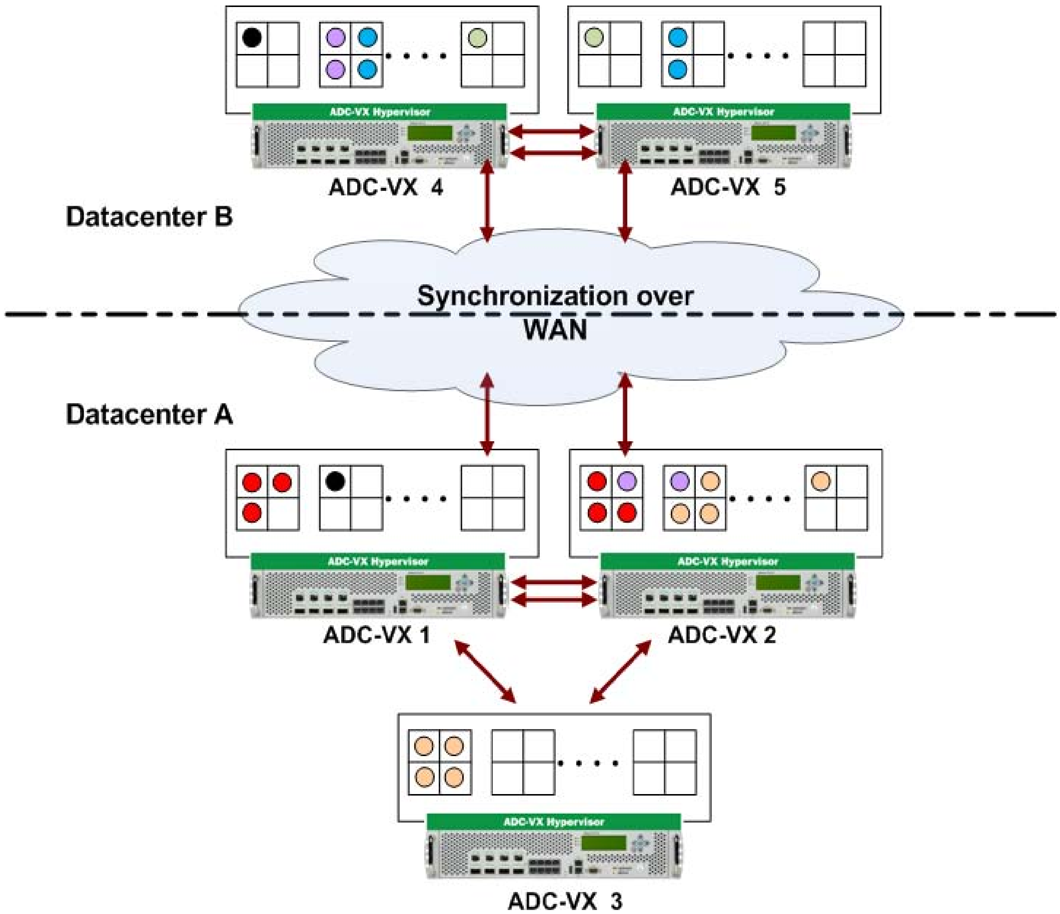


Peer Synchronization

Starting with version 28.0, you can define up to five (5) peers for each ADC-VX. This enables you to plan your system according to considerations such as risk, resource availability and internal organizational priorities. For more information on vADCs, see ADC-VX Management, page 403.

Figure 91 - Example Peer Synchronization Topology, page 580is an example topology for a set of Alteons that use peer synchronization:

Figure 91: Example Peer Synchronization Topology



Configuring Peer Synchronization

To configure peer synchronization, you must:

1. Configure peer switches (Alteons) for your Alteon (see To configure peers, page 581)
2. Associate the peer switches to vADCs (see To associate peer switches to a single vADC, page 581)

To configure peers

1. From the *Peer Switch* menu, define the address settings of the Global Administrator environment for the peer you want to configure.

   You can associate vADCs with the **range** option. You can enter a combination of single vADCs and ranges of vADCs. For example: 1, 3-5, 8

   Note: For a description of these menu options, see the *Alteon Application Switch Operating System Command Reference*.

```
>> # /cfg/sys/sync/peer
Enter peer switch number (1-5):1
------------------------------------------------------------
[Peer Switch 1 Menu]
     addr     - Set peer switch IP address
     ena      - Enable peer switch
     dis      - Disable peer switch
     range    - Set synchronization target for a range of vADCs
     del      - Delete peer switch
     cur      - Display current peer switch configuration
```

2. **Apply** and **save**. After setting peer switch addresses, vADC configuration is synchronized to the assigned peers.

To associate peer switches to a single vADC

When you create a vADC, you are prompted to associate peer switches to that vADC (see Creating a Basic vADC with the Creation Dialog, page 420). After creating the vADC, you can also separately associate and configure peers switches to it.

1. Access the *Peer Switch Addresses* prompt.

```
>> # /cfg/vadc/sys/sync/
Enter vADC Number [1-28]:1
[Peer Switch Addresses]
         Peer switch 1:   10.1.1.1, enabled
         Peer switch 2:   20.1.1.1, enabled
         Peer switch 3:   30.1.1.1, enabled
         Peer switch 4:   40.1.1.1, enabled
         Peer switch 5:   0.0.0.0 , disabled
Enter peer switch number (1-5):1
------------------------------------------------------------
```

2. Enter the peer switch number you want to associate to the selected vADC.
3. **Apply** and **save**. After setting peer switch addresses, vADC configuration is synchronized to the assigned peers.

Chapter 20 – Persistence

The persistence feature ensures that all connections from a specific client session reach the same real server, even when Server Load Balancing (SLB) is used.

The following topics are addressed in this chapter:

- Overview of Persistence, page 583—Gives an overview of persistence and the different types of persistence methods implemented in Alteon.
- Cookie-Based Persistence, page 585—Cookie persistence provides a mechanism for inserting a unique key for each client of a virtual server. This feature is only used in non-Secure Sockets Layer (SSL) connections. Discusses how persistence is maintained between a client and a real server using different types of cookies.
- HTTP and HTTPS Persistence Based on Client IP, page 584—Explains how both HTTP and HTTPS traffic map to the same server based on client IP.
- Server-Side Multi-Response Cookie Search, page 595—Explains how to configure Alteon to look through multiple HTTP responses from the server to achieve cookie-based persistence.
- SSL Session ID-Based Persistence, page 596—Explains how an application server and a client communicate over an encrypted HTTP session.
- Windows Terminal Server Load Balancing and Persistence, page 598—Explains how to configure load balancing and persistence for Windows Terminal Services.

Overview of Persistence

In a typical SLB environment, traffic comes from various client networks across the Internet to the virtual server IP address on Alteon. Alteon then load balances this traffic among the available real servers.

In any authenticated Web-based application, it is necessary to provide a persistent connection between a client and the content server to which it is connected. Because HTTP does not carry any state information for these applications, it is important for the browser to be mapped to the same real server for each HTTP request until the transaction is completed. This ensures that the client traffic is not load balanced mid-session to a different real server, forcing the user to restart the entire transaction.

Persistence-based SLB lets you configure the network to redirect requests from a client to the same real server that initially handled the request. Persistence is an important consideration for administrators of e-commerce Web sites, where a server may have data associated with a specific user that is not dynamically shared with other servers at the site.

In Alteon, persistence can be based on:

- Using Source IP Address, page 584
- Using Cookies, page 584
- Using SSL Session ID, page 584

Using Source IP Address

Using the source IP address as the key identifier was once the only way to achieve TCP/IP session persistence. There are two major conditions which cause problems when session persistence is based on a packet's IP source address:

- **Many clients sharing the same source IP address (proxied clients)**—Proxied clients appear to Alteon as a single source IP address and do not take advantage of SLB. When many individual clients behind a firewall use the same proxied source IP address, requests are directed to the same server, without the benefit of load balancing the traffic across multiple servers. Persistence is supported without the capability of effectively distributing traffic load.

  Also, persistence is broken if you have multiple proxy servers behind Alteon performing SLB. Alteon changes the client's address to different proxy addresses as attempts are made to load balance client requests.
- **Single client sharing a pool of source IP addresses**—When individual clients share a pool of source IP addresses, persistence for any given request cannot be assured. Although each source IP address is directed to a specific server, the source IP address itself is randomly selected, thereby making it impossible to predict which server will receive the request. SLB is supported, but without persistence for any given client.

Using Cookies

Cookies are strings passed via HTTP from servers to browsers. Based on the mode of operation, cookies are inserted by either Alteon or the server. After a client receives a cookie, a server can poll that cookie with a GET command, which allows the querying server to positively identify the client as the one that received the cookie earlier.

Cookie-based persistence solves the proxy server problem and gives better load distribution at the server site. In Alteon, cookies are used to route client traffic back to the same physical server to maintain session persistence.

Note: If the cookie expiration time is greater than the **`/cfg/slb/virt x/service x/ptmout`** value, timed-out requests will not be persistent.

Using SSL Session ID

The SSL session ID is effective only when the server is running SSL transactions. Because of the heavy processing load required to maintain SSL connections, most network configurations use SSL only when it is necessary. Persistence based on the SSL Session ID ensures completion of complex transactions in proxy server environments. However, this type of persistence does not scale on servers because of their computational requirements.

Note: SSL session ID persistence is not supported when SSL offloading is enabled and other more advanced persistency features, such as cookie persistency, are available.

HTTP and HTTPS Persistence Based on Client IP

Alteon lets you use the client IP address to maintain persistence for both HTTP and HTTPS sessions only. The **`pbind clientip`** command maintains persistence for the same service across multiple sessions from the same client, or maintains persistence between different services (for HTTP and

HTTPS traffic only) from the same client to map to the same server, as long as the same group is configured for both services. In Alteon, when the metric configured is hash, phash, or minmisses, persistence may also be maintained to the real server port (rport), in addition to the real server.

You should disable persistence to the rport on the following conditions:

- When there are two different services, such as TCP and UDP, that must maintain persistence to the same real server.
- When client IP-based persistence is not dependent on the load balancing metric.

To configuring Client IP address-based persistence

1. Configure real servers and services for basic SLB:
   - Define each real server and assign an IP address to each real server in the server pool.
   - Define a real server group and set up health checks for the group.
   - Define a virtual server on the virtual port for HTTP (port 80) and HTTPS (port 443) and assign both services to the same real server group. HTTP and HTTPS are supported only on their default service port numbers.
   - Enable SLB.
   - Enable client processing on the port connected to the client.

   For information on how to configure your network for SLB, see Server Load Balancing, page 165
2. If a proxy IP address is not configured on the client port, enable DAM for real servers.

```
>> # /cfg/slb/adv/direct ena
```

3. Select Client IP-based persistence as the persistent binding option for the virtual port.

   If multiple real server ports are configured for this service, you may choose whether to maintain persistence to the rport on the real server.

```
>> # /cfg/slb/virt 1/service <virtual port> pbind clientIP
Current persistent binding mode: disabled
Enter clientip|cookie|sslid|disable persistence mode:     clientIP
Use Rport? (y/n) [y]y
```

4. Enable client processing on the client port.

```
>> # /cfg/slb/port <port number> /client ena
```

Cookie-Based Persistence

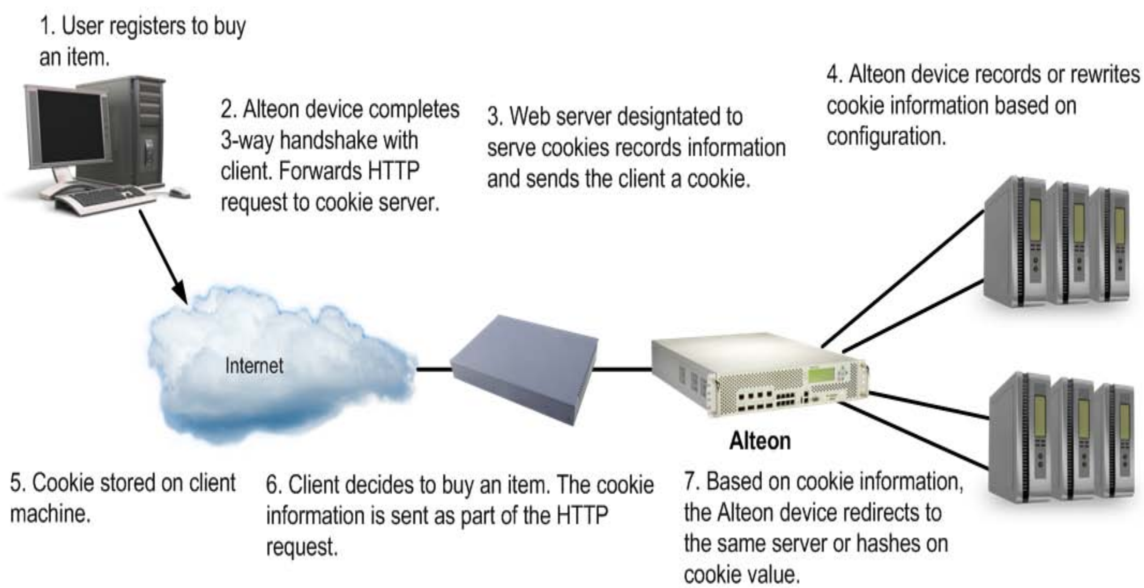
Cookies are a mechanism for maintaining the state between clients and servers. When the server receives a client request, the server issues a cookie, or token, to the client, which the client then sends to the server on all subsequent requests. Using cookies, the server does not require authentication, the client IP address, or any other time-consuming mechanism to determine that the user is the same user that sent the original request.

In the simplest case, the cookie may be just a "customer ID" assigned to the user. It may be a token of trust, allowing the user to skip authentication while his or her cookie is valid. It may also be a key that associates the user with additional state data that is kept on the server, such as a shopping cart and its contents. In a more complex application, the cookie may be encoded so that it actually contains more data than just a single key or an identification number. The cookie may contain the user's preferences for a site that allows their pages to be customized.

Note: If the cookie expiration time is greater than the **/cfg/slb/virt x/service x/ptmout** value, timed-out requests will not be persistent.

Figure 92 - Cookie-Based Persistence, page 586 illustrates how cookie-based persistence works:

Figure 92: Cookie-Based Persistence



The following topics discussing cookie-based persistence are discussed in this section:

Note: When both cookie-based pbind is used and HTTP modifications on the same cookie header are defined, Alteon performs both. This may lead to various application behaviors and should be done with caution.

Permanent and Temporary Cookies

Cookies can either be permanent or temporary. A permanent cookie is stored on the client's browser as part of the response from a Web site's server. It is sent by the browser when the client makes subsequent requests to the same site, even after the browser has been shut down. A temporary cookie is only valid for the current browser session. Similar to a SSL session-based ID, the temporary cookie expires when you shut down the browser. Based on RFC 2109, any cookie without an expiration date is a temporary cookie.

Cookie Formats

A cookie can be defined in the HTTP header (the recommended method) or placed in the URL for hashing. The cookie is defined as a "Name=Value" pair and can appear along with other parameters and cookies. For example, the cookie "**SessionID=1234**" can be represented in one of the following ways:

- In the HTTP Header:

  Cookie: SesssionID=1234
  Cookie: ASP_SESSIONID=POIUHKJHLKHD
  Cookie: name=john_smith

  The second cookie represents an Active Server Page (ASP) session ID. The third cookie represents an application-specific cookie that records the name of the client.
- Within the URL

  http://www.mysite.com/reservations/SessionID=1234

Cookie Properties

Cookies are configured by defining the following properties:

- Cookie names of up to 20 bytes.
- The offset of the cookie value within the cookie string.

  For security, the real cookie value can be embedded somewhere within a longer string. The offset directs Alteon to the starting point of the real cookie value within the longer cookie string.
- Length of the cookie value. This defines the number of bytes to extract for the cookie value within a longer cookie string.
- Whether to find the cookie value in the HTTP header (the default) or the URL.
- Cookie values of up to 64 bytes for hashing. Hashing on cookie values is used only with the passive cookie mode (Passive Cookie Mode, page 590), using a temporary cookie. Alteon mathematically calculates the cookie value using a hash algorithm to determine which real server should receive the request.
- An asterisk (*) in cookie names for wildcards. For example: `Cookie name = ASPsession*`

Client Browsers that Do Not Accept Cookies

Under normal conditions, most browsers are configured to accept cookies. However, if a client browser is not configured to accept cookies, you must use **hash** or **pbind clientip** (for client IP persistence) as the load-balancing metric to maintain session persistence.

With cookie-based persistence enabled, session persistence for browsers that do not accept cookies is based on the source IP address. However, individual client requests coming from a proxy firewall appear to be coming from the same source IP address. Therefore, the requests are directed to a single server, resulting in traffic being concentrated on a single real server instead of load-balanced across the available real servers.

Cookie Modes of Operation

Alteon supports the following modes of operation for cookie-based session persistence: *insert*, *passive*, and *rewrite* mode. Table 49 - Comparison Among the Three Cookie Modes, page 588 shows the differences between these modes:

Table 49: Comparison Among the Three Cookie Modes

Cookie Mode	Configuration Required	Cookie Location	Uses Session Entry
Insert Cookie	Alteon only	HTTP Header	No
Passive Cookie	Server and Alteon	HTTP Header or URL	Yes
Rewrite Cookie	Server and Alteon	HTTP Header	No

Insert Cookie Mode

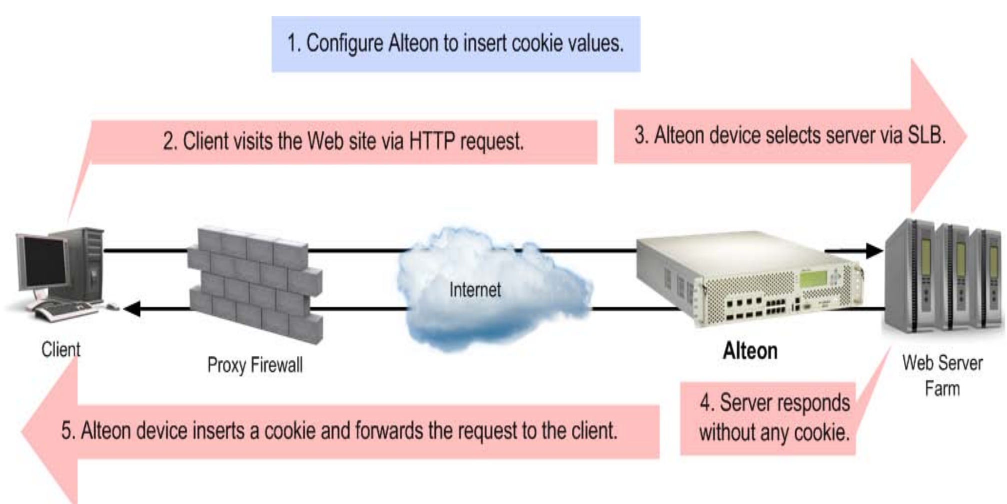
In the insert cookie mode, Alteon generates the cookie value on behalf of the server. Because no cookies are configured at the server, the need to install cookie server software on each real server is eliminated.

An inserted cookie has a value of 28 bytes, containing an 8-byte virtual server IP (VIP) value, an 8-byte client IP (CIP) value, a 4-byte real server port (RPORT) value, and an 8-byte random client ID value.

In this mode, the client sends a request to visit the Web site. Alteon performs load balancing and selects a real server. The real server responds without a cookie. Alteon inserts a cookie and forwards the new request with the cookie to the client.

Figure 93 - Insert Cookie Mode, page 588 illustrates insert cookie mode:

Figure 93: Insert Cookie Mode



Insert Cookie Mode Enhancement

This mode allows you to configure new cookie values, a path, and a secure flag. The configuration options are:

- **Cookie name**—This defaults to "AlteonP".
- **Expiry date and time**—If configured, the client sends cookie only until the expiration time. Otherwise, the cookie expires after the current session.

- **Domain name**—If configured, cookies are sent only to the domain, using the following commands:

```
>> #/cfg/slb/virt x/dname
>> #/cfg/slb/virt x/service y/hname
```

Note: Domain name is taken as "<hname>.<dname>". It defaults to a NULL string.

- **Cookie path**—If the cookie path is configured, the cookie is sent only for URL requests that are a subset of the path. The path defaults to "/".
- **Secure flag**—If the secure flag is set, the client is required to use a secure connection to obtain content associated with the cookie.

Setting Expiration Timer for Insert Cookie

If you configure for insert cookie persistence mode, you are prompted for cookie expiration timer. The expiration timer specifies a date string that defines the valid life time of that cookie. The expiration timer for insert cookie can be of the following types:

- **Absolute timer**—The syntax for the absolute timer is **MM/dd/yy[@hh:mm]**. The date and time is based on RFC 822, RFC 850, RFC 1036, and RFC 1123, with the variations that the only legal time zone is GMT. Once the expiration date is met, the cookie is not stored or given out. For example:

```
>> Enter cookie expiration: 12/31/04@11:59
Current persistent binding for http: disabled
New persistent binding for http: cookie
New cookie persistence mode: insert
Inserted cookie expires on Mon 12/31/04 at 11:59>>
```

- **Relative timer**—This timer defines the elapsed time from when the cookie was created. The syntax for the relative timer is `days[:hours[:minutes]]`. For example:

```
Enter cookie expiration: 32:25:61
Current persistent binding for http: disabled
New persistent binding for http: cookie
New cookie persistence mode: insert
Inserted cookie expires after 33 days 2 hours 1 minutes
```

Alteon adds or subtracts hours according to the time zone settings using the **/cfg/sys/ntp/tzone** command. When the relative expiration timer is used, ensure the **tzone** setting is set correctly. If NTP is disabled (using **/cfg/sys/ntp/off**), the **tzone** setting still applies to the cookie mode.

Note: If the cookie expiration timer is not specified, the cookie will expire when the user's session ends.

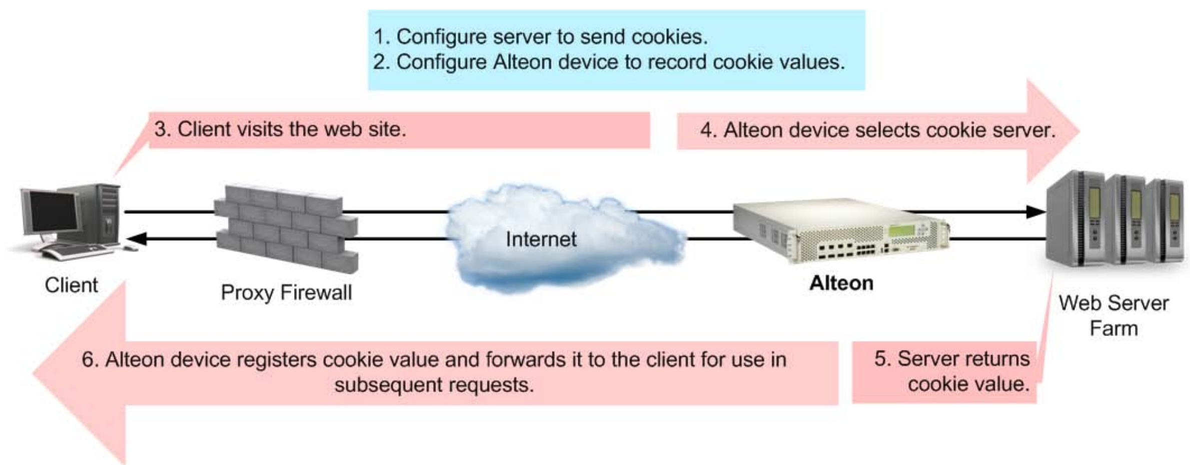
Passive Cookie Mode

In passive cookie mode, when the client first makes a request, Alteon selects the server based on the configured load-balancing metric. The real server embeds a cookie in its response to the client. Alteon records the cookie value and matches it in subsequent requests from the same client.

Note: Radware recommends passive cookie mode for temporary cookies. However, you can use this mode for permanent cookies if the server is embedding an IP address. In this case, a cookie has to be eight characters long, and every two characters represent one byte of IP address encoded in hexadecimal.

Figure 94 - Passive Cookie Mode, page 590 illustrates passive cookie mode operation:

Figure 94: Passive Cookie Mode



Subsequent requests from Client 1 with the same cookie value are sent to the same real server (RIP 1 in this example).

When passive cookie persistence mode is enabled, Alteon creates persistent entries for server returned responses with new cookie values within the same TCP connection.

Rewrite Cookie Mode

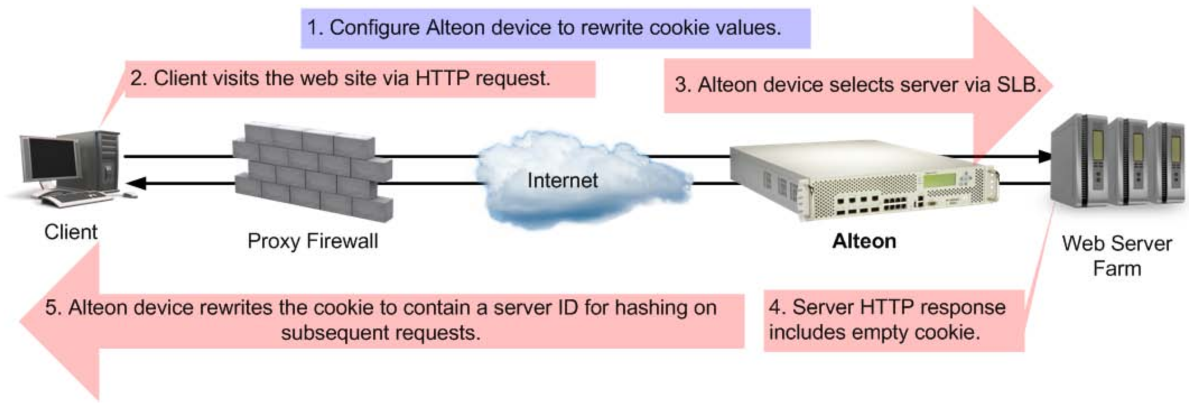
In rewrite cookie mode, Alteon generates the cookie value on behalf of the server, eliminating the need for the server to generate cookies for each client. Instead, the server is configured to return a special persistence cookie which Alteon is configured to recognize. Alteon then intercepts this persistence cookie and rewrites the value to include server-specific information before sending it on to the client. Subsequent requests from the same client with the same cookie value are sent to the same real server.

Rewrite cookie mode requires at least 28 bytes in the cookie header. An additional eight bytes must be reserved if you are using cookie-based persistence with Global Server Load Balancing (GSLB).

Note: Rewrite cookie mode only works for cookies defined in the HTTP header, not cookies defined in the URL.

Figure 95 - Rewrite Cookie Mode, page 591 illustrates the rewrite cookie mode operation:

Figure 95: Rewrite Cookie Mode



Note: When Alteon rewrites the value of the cookie, the rewritten value represents the responding server. This means that the value can be used for hashing into a real server ID or it can be the real server IP address. The rewritten cookie value is encoded.

Configuring Cookie-Based Persistence

The following is an example procedure for configuring cookie-based persistence.

To configure cookie-based persistence

1. Before you can configure cookie-based persistence, configure Alteon for basic SLB:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface.
   - Configure each real server with its IP address, name, weight, and so on.
   - Assign servers to real server groups.
   - Define virtual servers and services.

   For information on basic SLB configuration, see Server Load Balancing, page 165.
2. Either enable Direct Access Mode (DAM), or disable DAM and specify proxy IP addresses on the client ports.
   - Enable DAM.

```
>> # /cfg/slb/adv/direct ena
```

— Disable DAM and specify proxy IP addresses on the client ports.

```
>> # /cfg/slb/adv/direct disable                    (Disable DAM)
>> # /cfg/slb/port 1                                (Select network Port 1)
>> # pip 200.200.200.68                             (Set proxy IP address for Port 1)
```

3. Server processing is not required if using proxy IP addresses, so optionally you can disable it.

```
>> # /cfg/slb/port 1                                (Select Port 1)
>> # server dis                                     (Disable server processing on Port 1)
```

4. Select the appropriate load-balancing metric for the real server group.

```
>> # /cfg/slb/group 2/metric hash
```

— If embedding an IP address in the cookie, select roundrobin or leastconns as the metric.

— If you are not embedding the IP address in the cookie, select hash as the metric in conjunction with a cookie assignment server.

While you may experience traffic concentration using the hash metric with a cookie assignment server, using a hash metric without a cookie assignment server causes traffic concentration on your real servers.

5. Enable cookie-based persistence on the virtual server service.

In this example, cookie-based persistence is enabled for service 80 (HTTP).

```
>> # /cfg/slb/virt 1/service 80/pbind
Current persistent binding mode: disabled
Enter clientip|cookie|sslid|disable persistence mode: cookie
```

After you specify **cookie** as the persistence mode, you are prompted for the following parameters:

```
>> Enter insert|passive|rewrite cookie persistence mode [i/p/r]: p
Enter cookie name: CookieSession1
Enter starting point of cookie value [1-64]: 1
Enter number of bytes to extract [1-64]: 8
Look for cookie in URI [e|d]: d
```

— Cookie-based persistence mode: **insert, passive** or **rewrite**

— Cookie name

— Starting point of the cookie value

— Number of bytes to be extracted

— Look for cookie in the URI [e | d]

If you want to look for a cookie name/value pair in the URI, enter **e** to enable this option. To look for the cookie in the HTTP header, enter **d** to disable this option.

CLI Capture

When you issue the command **/cfg/slb/virt <virtual#>/service <service#>/pbind**, additional inputs taken from the user are listed in the output:

```
>> Virtual Server 10 http Service# /c/sl/vi 10/ser http/pbind
Current persistent binding mode: disabled

New persistent binding mode: cookie

Enter clientip|cookie|sslid|disable persistence mode: cookie

Enter passive|rewrite|insert cookie persistence mode [p/r/i]: i

Enter Cookie Name [AlteonP]:

Enter insert-cookie expiration as either:
...a date <MM/dd/yy[@hh:mm]> (e.g. 12/31/01@23:59)
 ...a duration <days[:hours[:minutes]]> (e.g 45:30:90)
...or none <return>

Enter cookie expiration: 0:0:59

Insert cookie domain name? (y/n) [n]yes

Enter path: "/test/test.html"

Is cookie secure[y/n] [n]yes
```

Cookie-Based Persistence Examples

This section includes the following cookie-based persistence examples:

- Example 1: Setting the Cookie Location, page 593
- Example 2: Parsing the Cookie, page 594
- Example 3: Using Passive Cookie Mode, page 594
- Example 4: Using Rewrite Cookie Mode, page 595

Example 1: Setting the Cookie Location

In this example, the client request has two different cookies labeled "UID". One exists in the HTTP header and the other appears in the URI:

```
GET /product/switch/UID=12345678;ck=1234...
Host: www.radware.com
Cookie: UID=87654321
```

A Look for the Cookie in the HTTP Header

```
>> # /cfg/slb/virt 1/service 80/pbind cookie passive UID 1 8 dis
```

The last parameter in this command answers the "**Look for cookie in URI?**" prompt. If you set this parameter to **disable**, Alteon uses **UID=87654321** as the cookie.

B Look for the Cookie in the URI

```
>> # /cfg/slb/virt 1/service 80/pbind cookie passive UID 1 8 ena
```

The last "Look for cookie in URI?" parameter is set to **enable**. As a result, Alteon uses **UID=12345678** as the cookie.

Example 2: Parsing the Cookie

This example shows three configurations which use the hashing key or wildcards to identify which part of the cookie value should be used for determining the real server. For example, the value of the cookie is defined as follows:

>> Cookie: sid=0123456789abcdef; name1=value1;...

A Select the entire value of the sid cookie as a hashing key for selecting the real server

```
>> # /cfg/slb/virt 1/service 80/pbind cookie passive sid 1 16 dis
```

This command directs Alteon to use the sid cookie, starting with the first byte in the value, and using the full 28 bytes.

B Select a specific portion of the sid cookie as a hashing key for selecting the real server

```
>> # /cfg/slb/virt 1/service 80/pbind cookie passive sid 8 4 dis
```

This command directs Alteon to use the sid cookie, starting with the eighth byte in the value, and using only four bytes. This uses **789a** as a hashing key.

C Using wildcards for selecting cookie names

```
>> # /cfg/slb/virt 1/service 80/pbind cookie passive ASPSESSIONED* 1 16 dis
```

With this configuration, Alteon looks for a cookie name that starts with **ASPSESSIONID. ASPSESSIONID123, ASPSESSIONID456**, and **ASPSESSIONID789** are seen as the same cookie name. If more than one cookie matches, only the first one is used.

Example 3: Using Passive Cookie Mode

If you are using passive cookie mode, Alteon examines the server's **Set-Cookie**: value and directs all subsequent connections to the server that assigned the cookie.

For example, if Server 1 sets the cookie as "**Set-Cookie**: **sid=12345678**", then all traffic from a particular client with cookie **sid=12345678** is directed to Server 1.

```
>> # /cfg/slb/virt 1/service 80/pbind cookie passive sid 1 8 dis
```

Example 4: Using Rewrite Cookie Mode

A Rewrite server cookie with the encrypted real server IP address

In cookie rewrite mode, if the cookie length parameter is configured to be 28 bytes, Alteon rewrites the placeholder cookie value with the encrypted real server IP address:

```
>> # /cfg/slb/virt 1/service 80/pbind cookie rewrite sid dis
```

If the server is configured to include a placeholder cookie, such as `Set-Cookie: sid=alteonpersistence;`, then Alteon rewrites the first 28 bytes of the cookie to include the server's encrypted IP address: `Set-Cookie: sid=alteonpersistence;`.

All subsequent traffic from a specific client with this cookie is directed to the same real server.

B Rewrite server cookie with the encrypted real server IP address and virtual server IP address

If the cookie length is configured to be 28 bytes, Alteon rewrites the cookie value with the encrypted real server IP address and virtual server IP address:

```
>> # /cfg/slb/virt 1/service 80/pbind cookie rewrite sid dis
```

If the server is configured to include a placeholder cookie as `Set-Cookie: sid=alteonpersistence;`, then Alteon rewrites the first 28 bytes of the cookie to include the encrypted real server IP address and virtual server IP address:`Set-Cookie: sid=cdb20f04cdb20f0a;`

All subsequent traffic from a specific client to the particular virtual server IP address with this cookie is directed to the same real server.

Server-Side Multi-Response Cookie Search

Cookie-based persistence requires Alteon to search the HTTP response packet from the server and, if a persistence cookie is found, set up a persistence connection between the server and the client. Alteon looks through the first HTTP response from the server. While this approach works for most servers, some customers with complex server configurations might send the persistence cookie a few responses later. In order to achieve cookie-based persistence in such cases, Alteon lets the network administrator configure Alteon to search through multiple HTTP responses from the server.

In Alteon, the network administrator can modify a response counter to a value from 1 through 16. Alteon looks for the persistence cookie in this number of responses (each of them can be multi-frame) from the server.

Configuring Server-Side Multi-Response Cookie Search

The following is an example procedure for configuring a server-side multi-response cookie search.

To configure the server-side multi-response cookie search

```
>> # /cfg/slb/virt <virtual server> /service 80/http/rcount
Current Cookie search response count:
Enter new Cookie search response count [1-16]:
```

Proxy Support for Insert Cookie

When the insert cookie persistence mode is enabled, Alteon parses through every HTTP request within the same TCP connection to look for the configured cookie name to use for persistency. If the client request arrives without a cookie, the request is forwarded to the existing binded server. When cookie insert persistence mode is enabled, Alteon needs to insert a cookie in the server-returned response for those client requests without a cookie.

If the client request arrives with a cookie, then the cookie is used to check against the persistence binding table.

SSL Session ID-Based Persistence

SSL is a set of protocols built on top of TCP/IP that allows an application server and client to communicate over an encrypted HTTP session, providing authentication, non-repudiation, and security. The SSL protocol *handshake* is performed using clear (unencrypted) text. The content data is then encrypted, using an algorithm exchanged during the handshake, prior to being transmitted.

Using the SSL session ID, Alteon forwards the client request to the same real server to which it was bound during the last session. Because the SSL protocol allows many TCP connections to use the same session ID from the same client to a server, the key exchange needs to be done only when the session ID expires. This reduces server overhead and provides a mechanism, even when the client IP address changes, to send all sessions to the same real server.

Notes

- The SSL session ID can only be read after the TCP three-way handshake. In order to make a forwarding decision, Alteon must terminate the TCP connection to examine the request.
- SSL session ID persistence is not supported when SSL offloading is enabled and other more advanced persistency features, such as cookie persistency, are available.

Some versions of Web browsers allow the session ID to expire every two minutes, thereby breaking the SSL ID persistence. To resolve this issue, use persistency with metric **`hash`** or **`pbind clientip`**.

Note: The destination port number to monitor for SSL traffic is user-configurable.

Alteon also has set of SSL offloading features for manipulating SSL traffic. For more information, see Offloading SSL Encryption and Authentication, page 337.

How SSL Session ID-Based Persistence Works

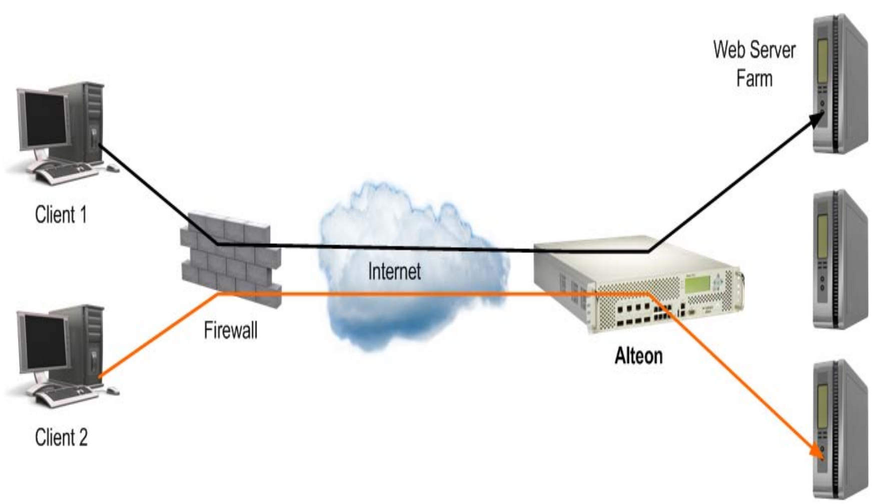
- All SSL sessions that present the same session ID (32 random bytes chosen by the SSL server) are directed to the same real server.
- New sessions are sent to the real server based on the metric selected (hash, roundrobin**,** leastconns, minmisses, response, and bandwidth).
- If no session ID is presented by the client, Alteon picks a real server based on the metric for the real server group and waits until a connection is established with the real server and a session ID is received.
- The session ID is stored in a session hash table. Subsequent connections with the same session ID are sent to the same real server. This binding is preserved even if the server changes the session ID midstream. A change of session ID in the SSL protocol causes a full three-way handshake to occur.

- Session IDs are kept on Alteon until an idle time equal to the configured server timeout (a default of 10 minutes) for the selected real server has expired.

Figure 96 - SSL Session ID-Based Persistence, page 597 illustrates persistence based on the SSL session ID, as follows:

1. An SSL Hello handshake occurs between Client 1 and Server 1 via Alteon.
2. An SSL session ID is assigned to Client 1 by Server 1.
3. Alteon records the SSL session ID.
4. Alteon selects a real server based on the existing SLB settings. As a result, subsequent connections from Client 1 with the same SSL session ID are directed to Server 1.

Figure 96: SSL Session ID-Based Persistence



5. Client 2 appears to have the same source IP address as Client 1 because they share the same proxy firewall.

   However, Alteon does not direct Client 2 traffic to Server 1 based on the source IP address. Instead, an SSL session ID for the new traffic is assigned. Based on SLB settings, the connection from Client 2 is spliced to Server 3. As a result, subsequent connections from Client 2 with the same SSL session ID are directed to Server 3.

Configuring SSL Session ID-Based Persistence

The following is an example procedure for configuring SSL session ID-based persistence.

To configure session ID-based persistence for a real server

1. Configure real servers and services for basic SLB:
   — Define each real server and assign an IP address to each real server in the server pool.
   — Define a real server group and set up health checks for the group.
   — Define a virtual server on the virtual port for HTTPS (for example, port 443), and assign a real server group to service it.
   — Enable SLB.
   — Enable client processing on the port connected to the client.

For information on how to configure your network for SLB, see Server Load Balancing, page 165.

2. If a proxy IP address is not configured on the client port, enable DAM for real servers.

```
>> # /cfg/slb/adv/direct ena
```

3. Select session ID-based persistence as the persistent binding option for the virtual port.

```
>> # /cfg/slb/virt <virtual server number> /service <virtual port> pbind sslid
```

4. Enable client processing on the client port.

```
>> # /cfg/slb/port <port number> /client ena
```

Windows Terminal Server Load Balancing and Persistence

Windows Terminal Services refers to a set of technologies that allow Windows users to run Windows-based applications remotely on a computer running as the Windows Terminal Server. Alteon includes load balancing and persistence options designed specifically for Windows Terminal Services.

In a load-balanced environment, a group of terminal servers have incoming session connections distributed in a balanced manner across the servers in the group. The Windows session director is used to keep a list of sessions indexed by username. This allows a user to reconnect to a disconnected user session.

The session director provides functionality that allows a group of terminal servers to coordinate the reconnection of disconnected sessions. The session director is updated and queried by the terminal servers whenever users log on, log off, or disconnect their sessions while leaving their applications active.

The client can be reconnected to the terminal server where the user's disconnected session resides using the routing token information. The session director passes the routing token information to the client with the correct server IP address embedded. The client presents this routing token to the load balancer when it reconnects to the virtual IP address. The load balancer deciphers the token and sends the client to the correct terminal server.

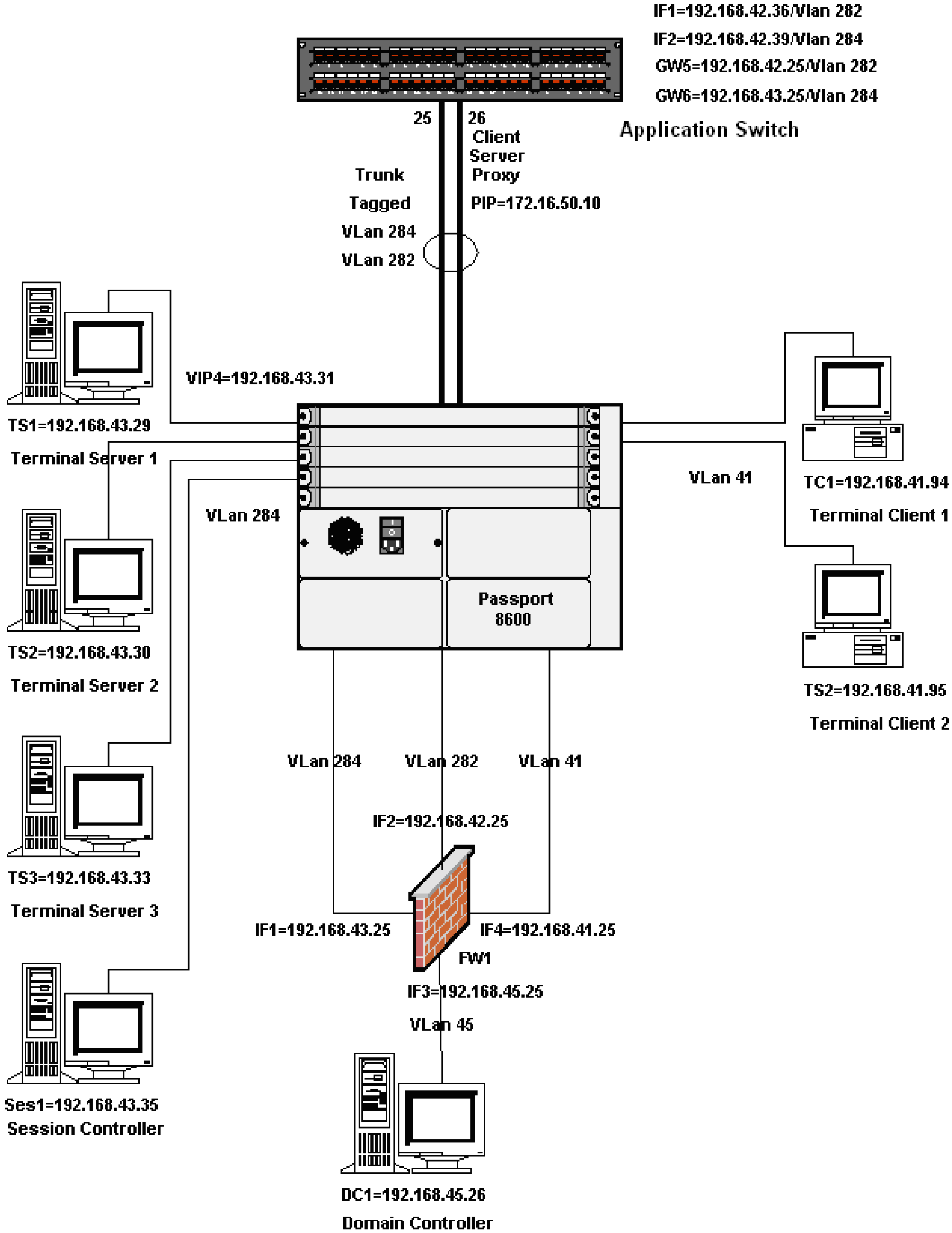
In some instances, a dedicated session director may not exist. If this is the case, enable the *userhash* functionality to perform the terminal server binding operation based on user name hashing.

By default, Windows Terminal Server traffic uses TCP port 3389 but it can configured to work on any non-standard port.

For further information regarding Windows Terminal Services, refer to the Microsoft Web site.

Figure 97 - Windows Terminal Server Load Balancing Network Topology, page 599 illustrates a sample Windows Terminal Server Load Balancing network topology:

Figure 97: Windows Terminal Server Load Balancing Network Topology



Configuring Windows Terminal Server Load Balancing and Persistence

When using Windows Terminal Server load balancing and persistence, ensure that either DMA is enabled or a proxy IP address has been configured.

To configure Windows Terminal Server load balancing and persistence

1. Access the *Windows Terminal Server* menu.

```
>> Main# /cfg/slb/virt <virtual server number> /service 3389/wts
```

2. Enable the Windows Terminal Server feature.

```
>> WTS Load Balancing# ena
```

3. Optionally, enable the WTS userhash.

Note: If the dedicated session director does not exist to relate users to disconnected sessions, Radware recommends enabling the userhash functionality to perform this task.

```
>> WTS Load Balancing# userhash enable
```

Chapter 21 – Advanced Denial of Service Protection

This chapter describes the Advanced Denial of Service (DoS) protection features that can be used to prevent a wide range of network attacks. The commands to execute these features are located in the *Security* menu, and are enabled via a separately purchased license key.

Note: If you purchased the Advanced DoS protection option, enable it by typing `/oper/swkey` and entering its software key.

Background

The Advanced DoS feature set extends the Alteon functionality to act as an application-intelligent firewall. You can use these features to perform deep inspection and blocking of malicious content. For example, many newer viruses, worms, malicious code, applications with security bugs, and cyber attacks have targeted application and protocol weaknesses by tunneling through a firewall over HTTP port 80, or by encapsulating attacks into SSL tunnels. Such packets can pass undetected through standard network firewalls, which are configured only to open or close access to HTTP port 80. Many of the attacks (such as nullscan, xmascan, scan SYNFIN) are created with purposely malformed packets that include illegal fields in the IP headers.

Security Inspection Workflow

A typical Alteon workflow to handle security inspection is as follows:

1. Alteon is configured with a predefined set of rules.

   To increase the performance of the inspection, complex pattern inspection rules can be defined with an offset value so that the inspection engine can go directly to the location to be inspected. A virus pattern often is a combination of multiple patterns within the IP payload. Alteon can be configured to inspect multiple patterns and locate them at different offsets within the payload.
2. Packets enter Alteon.
3. Alteon inspects the packet by comparing the rules to the content of the packet.

4. When an attack pattern is matched, Alteon drops this packet, and creates a session so that subsequent packets of the same session (if it is TCP) are also dropped without going through additional rule inspection.

Other Types of Security Inspection

Alteon can use its inspection engine to provide rate limiting capability to complex protocols such as those used in the peer-to-peer programs that use dynamic ports to establish communication between clients. Standard firewalls are unable to detect these programs, because the protocol signatures do not appear at the Layer 4 port level. Many of these protocols have signatures that are embedded in the HTTP header or, in some cases, embedded in the data payload itself. For more information, see TCP or UDP Pattern Matching, page 618.

Alteon can also rate limit the amount of the total traffic generated by these programs. This is especially useful in Cable ISP and universities where peer-to-peer programs can reach as much as 70% of the total traffic. For more information, see Protocol-Based Rate Limiting, page 611.

IP Address Access Control Lists

Alteon can be configured with IP access control lists (ACLs) composed of ranges of client IP addresses that are to be denied access to Alteon. When traffic ingresses Alteon, the client source or destination IP address is checked against this pool of addresses. If a match is found, then the client traffic is blocked.

ACLs versus Filters

ACLs are used to control which IP addresses are allowed access to a network. Unlike a filter, the IP ACL feature can only perform a *deny* action. The decision about whether to deny traffic is based solely on whether a match is found between the client IP and the ACL. The IP access control list (**ipacl**) commands can be used to configure a pool of up to 8192 blockable IP addresses (5120 configured source IP addresses, 1024 configured destination IP addresses, 1024 operationally added source IP addresses, and 1024 operationally added destination IP addresses).

While filters can perform the same function by blocking IP addresses ranges, they contain additional information which also must be matched on ingress traffic before determining whether to allow, deny, or redirect traffic.

How IP ACL Works

IP ACL uses a hash table to effectively block a configured range of IP addresses. The ACL is a global list which is by default disabled. It is enabled on a per-port basis.

When a packet ingresses a port that has been enabled with IP ACK processing, Alteon compares the client source or destination IP address with internal hash tables containing the IP addresses. If a match is found, the packet is dropped. If no match on the address is found in any of the hash tables, the packet is allowed to pass.

Configuring Blocking with IP Access Control Lists

The following is an example procedure for configuring blocking with IP access control lists.

To configure blocking with IP ACLs

1. Add the IP addresses that you want to block.
   — The following example blocks source addresses 192.168.40.0-255:

`>> Main # /cfg/security/ipacl`	(Select the *IP ACL* menu)
`>> IP ACL# add 192.168.40.0`	(Enter a network address)
`Enter IP subnet mask [default is 255.255.255.255]: 255.255.255.0`	(Enter the appropriate mask)

   — The following example blocks destination addresses 192.180.11.0-255:

`>> Main# /cfg/security/ipacl`	(Select the *IP ACL* menu)
`>> IP ACL# dadd 192.180.11.0`	(Enter a network address)
`Enter IP subnet mask [default is 255.255.255.255]: 255.255.255.0`	(Enter the appropriate mask)

2. Repeat step 1 to configure any other IP addresses that should be dropped.
3. Enable IP ACL processing on the ingress port.

```
>> Main# /cfg/security/port <x> /ipacl ena
Current IP ACL processing: disabled
New IP ACL processing: enabled
```

4. Apply and save the configuration.

Viewing IP ACL Statistics

You can view the accumulated blocked packets for each IP address /mask pair by entering the following command:

```
>> /stats/security/ipacl/dump

IP ACL stats:
Source IP ACL hits: 3
Source IP Addr  Mask            Type
--------------- --------------- -----
192.168.1.0     255.255.255.0   cfg

Destination IP ACL hits: 0
Dest IP Addr    Mask            Type
--------------- --------------- -----
No destination IP ACL's created
```

Protection Against Common Denial of Service Attacks

Alteon can protect ports against a variety of DoS attacks, including Port Smurf, LandAttack, Fraggle, Blat, Nullscan, Xmascan, PortZero, and Scan SynFin, among many others. Enable DoS protection on any ports connected to unsafe networks.

Configuring Ports with DoS Protection

The following is an example procedure for configuring ports with DoS protection.

To enable DoS protection on any port that is connected to an unsafe network

Once enabled, this feature detects and drop packets containing any of the supported types of DoS attack.

1. Enable DoS protection on the ports.

```
>> Main# /cfg/security/port 1/dos enable
>> Current Protocol anomaly and DOS attack prevention: disabled
New Protocol anomaly and DOS attack prevention: enabled
```

2. Add a DoS attack type to guard against.

```
>> Main# /cfg/security/port 1/dos/add <DoS attack type>
```

Note: To determine which DoS attack types a port is guarding against, view the current settings by using the command /cfg/security/port <port number>/cur.

3. Optionally, remove a DoS attack type from a port:

```
>> Main# /cfg/security/port 1/dos/rem <DoS attack type>
```

4. Repeat step 1 and step 2 to apply DoS protection to any other ports.
5. Apply and save the configuration.

Viewing DoS Statistics

You can view the number of times packets are dropped when a DoS attack is detected on Alteon or on a specific port.

When an attack is detected, Alteon generates a message similar to the following:

```
>> Jun 18 22:33:32 ALERT   security: DoS Attack:Fraggle
sip:192.115.106.200 dip:192.115.106.255 ingress port:1
```

To shows DoS statistics on all ports where DoS protection is enabled

```
>> /stats/security/dos/dump
---------------------------------------------------------------------------
Protocol anomaly and DoS attack prevention statistics for port 1:
Protocol anomaly and DoS attack prevention statistics for port 8
broadcast    :          1
loopback     :          8
land         :          1
ipptl        :          1
ipprot       :          1
fragmoredont:           1
fragdata     :          2
fragboundary:           2
fraglast     :          1
fragdontoff :           1
fragoff      :          1
fragoversize:           1
tcplen       :          4
tcpportzero :           2
blat         :          1
nullscan     :          1
fullxmasscan:           1
finscan      :          1
vecnascan    :          5
xmasscan     :          1
synfinscan   :          1
synfrag      :          1
ftpport      :          1
dnsport      :          1
seqzero      :          1
ackzero      :          1
udplen       :          2
udpportzero :           2
fraggle      :          1
snmpnull     :          1
icmplen      :          2
smurf        :          1
icmpdata     :          1
igmplen      :          2
igmpfrag     :          1
arpnbcast    :         21
------
Totals       :         77
```

Specific subtotals are given for only those ports that are seeking attack traffic.

Viewing DoS Statistics Per Port

The following is an example procedure for viewing DoS statistics per port.

To display DoS protection statistics for a specified port

```
>> /stats/security/dos/port <port>
```

Understanding the Types of DoS Attacks

This section includes an explanation of the different types of DoS attacks.

To obtain a brief explanation of each type of detected DoS attack

```
>> /stats/security/dos/help
```

Once DoS protection is enabled on the appropriate ports, Alteon performs checks on incoming packets, as described in Table 50.

Table 50: DoS Attacks Detected by Alteon

DoS Attack	Description	Action
IPLen	An IPv4 packet is sent with an invalid payload or IP header length.	Alteon checks for malformed packets that have either an IP header length less than 20 bytes, an IP total packet length less than the IP header length, or an actual packet length less than the IP total length, and drops any matching packets.
IPVersion	An IPv4 packet is sent with an invalid IP version.	Alteon checks for IPv4 packets marked with a version other than version 4, and drops any matching packets.
Broadcast	An IPv4 packet with a broadcast source or destination IP address.	Alteon checks for IPv4 packets with a broadcast source or destination IP address (0.0.0.0,255.255.255.255), and drops any matching packets.
LoopBack	An IPv4 packet with a loopback source or destination IP address.	Alteon checks for IPv4 packets with a loopback source or destination IP address (127.0.0.0/8), and drops any matching packets.
LandAttack	Packets with source IP (sip) equal to destination IP (dip) address.	Alteon checks for a sip equal to the dip in the packet, and drops any matching packets.
IPReserved	An IPv4 packet with the reserved IP bit set.	Alteon checks for IPv4 packets with the reserved IP bit set, and drops any matching packets.
IPTTL	An IPv4 packet with a small IP TTL.	Alteon checks for IPv4 packets with a small IP TTL, and drops any matching packets.

Table 50: DoS Attacks Detected by Alteon

DoS Attack	Description	Action
IPProt	An IPv4 packet with an unassigned or reserved IP protocol.	Alteon checks for IPv4 packets with an unassigned or reserved IP protocol, and drops any matching packets.
IPOptLen	An IPv4 packet with an invalid IP options length.	Alteon checks for IPv4 packets with an invalid IP options length set, and drops any matching packets.
FragMoreDont	An IPv4 packet with the “more” fragments and “don't” fragment bits set.	Alteon checks for IPv4 packets with both the “more” fragments and “don't” fragments bits set, and drops any matching packets.
FragData	An IPv4 packet with the “more” fragments bit set but a small payload.	Alteon checks for IPv4 packets with the “more” fragments bit set but exhibiting a small payload, and drops any matching packets.
FragBoundary	An IPv4 packet with the “more” fragments bit set but a payload not at an 8-byte boundary.	Alteon checks for IPv4 packets with the more fragments bit set but whose payload is not at an 8-byte boundary, and drops any matching packets.
FragLast	An IPv4 packet that is the last fragment but no payload.	Alteon checks for IPv4 packets with the last fragment bit set but no payload, and drops any matching packets.
FragDontOff	An IPv4 packet with a non-zero fragment offset and the “don't” fragment bits set.	Alteon checks for IPv4 packets with a non-zero fragment offset and the “don't” fragment bits set, and drops any matching packets.
FragOpt	An IPv4 packet with a non-zero fragment offset and IP options bits set.	Alteon checks for IPv4 packets with a non-zero fragment offset and the IP options bits set, and drops any matching packets.
FragOff	An IPv4 packet with a small non-zero fragment offset.	Alteon checks for IPv4 packets with a small non-zero fragment offset, and drops any matching packets.
FragOverSize	An IPv4 packet with a non-zero fragment offset and an oversized payload.	Alteon checks for IPv4 packets with a non-zero fragment offset and an oversized payload, and drops any matching packets.
TCPLen	A TCP packet with a TCP header length less than 20 bytes and an IP data length less than the TCP header length.	Alteon checks for TCP packets with a TCP header length less than 20 bytes and an IP data length less than the TCP header length, and drops any matching packets.
TCPPortZero	A TCP packet with a source or destination port of zero.	Alteon checks for TCP packets with a source or destination port of zero, and drops any matching packets.
TCPReserved	A TCP packet with the TCP reserved bit set.	Alteon checks for TCP packets with the TCP reserved bit set, and drops any matching packets.
NULLscan	A TCP packet with a sequence number of zero or all of the control bits are set to zero.	Alteon checks for TCP packets with a sequence number or zero or with all control bits set to zero, and drops any matching packets.

Table 50: DoS Attacks Detected by Alteon

DoS Attack	Description	Action
FullXmasScan	A TCP packet with all control bits set.	Alteon checks for TCP packets with all of the control bits set, and drops any matching packets.
FinScan	A TCP packet with only the FIN bit set.	Alteon checks for TCP packets with only the FIN bit set, and drops any matching packets.
VecnaScan	A TCP packet with only the URG, PUSH, URG|FIN, PSH|FIN, or URG|PSH bits set.	Alteon checks for TCP packets with only the URG, PUSH, URG|FIN, PSH|FIN, or URG|PSH bits set and drops any matching packets.
Xmascan	Sequence number is zero and the FIN, URG, and PSH bits are set.	Alteon checks for any TCP packets where the sequence number is zero and the FIN, URG, and PSH bits are set, and drops any matching packets.
SYNFIN Scan	SYN and FIN bits set in the packet.	Alteon checks for TCP packets with the SYN and FIN bits set, and drops any matching packets.
FlagAbnormal	A TCP packet with an abnormal control bit combination set.	Alteon checks for an abnormal control bit combination, and drops any matching packets.
SynData	A TCP packet with the SYN bit set and that also has a payload.	Alteon checks for TCP packets with the SYN bit set and that also has a payload, and drops any matching packets.
SynFrag	A TCP packet with the SYN and more fragments bits set.	Alteon checks for TCP packets with the SYN and more fragments bits set, and drops any matching packets.
FTPPort	A TCP packet with a source port of 20, a destination port of less than 1024 and the SYN bit set.	Alteon checks for TCP packets with a source port of 20, a destination port of less than 1024, and the SYN bit set, and drops any matching packets.
DNSPort	A TCP packet with a source port of 53, a destination port of less than 1024 and the SYN bit set.	Alteon checks for TCP packets with a source port of 53, a destination port of less than 1024, and the SYN bit set and drops any matching packets.
SeqZero	A TCP packet with a sequence number of zero.	Alteon checks for TCP packets with a sequence number of zero, and drops any matching packets.
AckZero	A TCP packet with an acknowledgement number of zero and the ACK bit set.	Alteon checks for TCP packets with an acknowledgement number of zero and the ACK bit set, and drops any matching packets.
TCPOptLen	A TCP packet with a TCP options length of less than two or where the TCP options length is greater than the TCP header length.	Alteon checks for TCP packets with a TCP options length of less than two or where the TCP options length is greater than the TCP header length, and drops any matching packets.
UDPLen	An UDP packet with a UDP header length of less than 8 bytes or where the IP data length is less than the UDP header length.	Alteon checks for UDP packets with a UDP header length of less than 8 bytes or where the IP data length is less than the UDP header length, and drops any matching packets.

Table 50: DoS Attacks Detected by Alteon

DoS Attack	Description	Action
UDPPortZero	An UDP packet with a source or destination port of zero.	Alteon checks for UDP packets with a source or destination port of zero, and drops any matching packets.
Fraggle	Similar to a smurf attack, attacks are directed to a broadcast address, except that the packets sent are UDP, not ICMP.	Deny all the UDP packets with destination address set to a broadcast address. **User action**: Configure UDP and ICMP Rate Limiting, page 613.
Pepsi	An UDP packet with a source port of 19 and destination port of 7, or vice versa.	Alteon checks for UDP packets with a source port of 19 and destination port of 7, or vice versa, and drops any matching packets.
RC8	An UDP packet with a source and destination port of 7.	Alteon checks for UDP packets with a source and destination port of 7, and drops any matching packets.
SNMPNull	An UDP packet with a destination port of 161 and no payload.	Alteon checks for UDP packets with a destination port of 161 and no payload and drops any matching packets.
ICMPLen	An ICMP packet with an improper ICMP header length.	Alteon checks for ICMP packets with an improper ICMP header length and drops any matching packets.
Smurf	The attacker sends ICMP ping requests to multiple broadcast destination IP (*x.x.x*.255). The packet contains spoofed source IP of the victim.	Alteon checks every packet for destination IP set to a broadcast address in a filter, and drops any matching packet.
ICMPData	An ICMP packet with a zero fragment offset and a large payload.	Alteon checks for ICMP packets with a zero fragment offset and a large payload, and drops any matching packets.
ICMPOff	An ICMP packet with a large fragment offset.	Alteon checks for ICMP packets with a large fragment offset, and drops any matching packets.
ICMPType	An ICMP packet where the type is unassigned or reserved.	Alteon checks for ICMP packets where the type is unassigned or reserved, and drops any matching packets.
IGMPLen	An IGMP packet with an improper IGMP header length.	Alteon checks for IGMP packets with an improper IGMP header length, and drops any matching packets.
IGMPFrag	An IGMP packet with the more fragments bit set and a non-zero fragment offset.	Alteon checks for IGMP packets with the more fragments bit set and a non-zero fragment offset, and drops any matching packets.
IGMPType	An IGMP packet with the type of unassigned or reserved.	Alteon checks for IGMP packets with the type of unassigned or reserved, and drops any matching packets.
ARPLen	An ARP request or reply packet with an improper length.	Alteon checks for ARP request or reply packets with an improper length, and drops any matching packets.
ARPNBCast	An ARP request packet with a non-broadcast destination MAC address.	Alteon checks for ARP request packets with a non-broadcast destination MAC address, and drops any matching packets.

Table 50: DoS Attacks Detected by Alteon

DoS Attack	Description	Action
ARPNUCast	An ARP reply packet with a non-unicast destination MAC address.	Alteon checks for ARP reply packets with a non-unicast destination MAC address, and drops any matching packets.
ARPSpoof	An ARP request or reply packet with a mismatched source with sender MAC addresses or destination with target MAC addresses.	Alteon checks for ARP request or reply packets with a mismatched source with sender MAC addresses, or destination with target MAC addresses, and drops any matching packets. **Note:** VRRP enabled gateways can produce a false positive for arpspoof.
GARP	An ARP request or reply packet with the same source and destination IP.	Alteon checks for ARP request or reply packets with the same source and destination IP, and drops any matching packets.
IP6Len	An IPv6 packet with an improper header length.	Alteon checks for IPv6 packets with an improper header length, and drops any matching packets.
IP6Version	An IPv6 packet with the IP version set to a value other than 6.	Alteon checks for IPv6 packets with the IP version set to a value other than 6, and drops any matching packets.
Blat	TCP packets with a source IP (sip) not equal to a destination IP (dip), but a source port (sport) equal to the destination port (dport).	Alteon checks for source IP not equal to destination IP and sport equal to dport, and drops any matching packets.

DoS Attack Prevention Configuration

Many of the DoS attacks that Alteon guards against have configurable values associated with them. These values allow Alteon to determine if the packets under inspection are DoS attacks based on additional administrator input.

Table 51 outlines these DoS attacks and their associated commands.

Table 51: DoS Attack Prevention Commands

DoS Attack	Command
IPTTL	`/cfg/security/dos/ipttl <smallest allowable IP TTL>`
IPProt	`/cfg/security/dos/ipprot <highest allowable protocol>`
FragData	`/cfg/security/dos/fragdata <smallest allowable IP fragment payload>`
FragOff	`/cfg/security/dos/fragoff <smallest allowable IP fragment offset>`
SynData	`/cfg/security/dos/syndata <largest allowable TCP SYN payload>`
ICMPData	`/cfg/security/dos/icmpdata <largest allowable ICMP payload>`
ICMPOff	`/cfg/security/dos/icmpoff <largest allowable ICMP offset>`

To view the current values associated with these DoS attacks

Use of the of the following commands:

```
>> Main# /cfg/security/dos/cur
>> Main# /info/security/dos
```

To display a brief explanation of any of the DoS attacks that Alteon guards against

```
>> Main# /cfg/security/dos/help
```

Preventing Other Types of DoS Attacks

Table 52 describes how to prevent other types of DoS attacks.

Table 52: DoS Attack Prevention Commands

DoS Attack	Description	User Action
Ping Flood	Flood of ICMP packets intentionally sent to overwhelm servers. The server is removed from service while it attempts to reply to every ping.	Configure 4: A Rate Limiting Filter to Thwart Ping Flooding, page 617 to limit ICMP packets.
Ping of Death	A ping of death attack sends fragmented ICMP echo request packets. When these packets are reassembled, they are larger than the 65536 byte packets allowed by the IP protocol. Oversized packets cause overflows in the server's input buffer, and can cause a system to crash, hang, or reboot.	Configure FragOversize or Matching and Denying Large Packets—ICMP Ping of Death Example, page 623.

Protocol-Based Rate Limiting

Alteon lets you detect and block certain kinds of protocol-based attacks. These attacks can flood servers with enough traffic to severely affect their performance or bring them down altogether. Protocol-based rate limiting is implemented via filters. Alteon currently supports rate limiting on TCP, UDP, and ICMP protocols. Each filter is configured with one of the above protocols, and then rate limiting is enabled or disabled in the *Filtering Advanced* menu.

- **TCP Rate Limiting**—Limits new TCP connection requests or SYN packets. Alteon monitors the rate of incoming TCP connection requests to a virtual IP address and limits the client requests with a known set of IP addresses. For more information, see TCP Rate Limiting, page 613.

- **UDP and ICMP Rate Limiting**—Counts all received packets from a client and compares against the configured maximum threshold. When the maximum configured threshold has been reached before the time window expires, Alteon drops until the configured holddown period expires. For more information, see UDP and ICMP Rate Limiting, page 613.

Time Windows and Rate Limits

A *time window* is a configured period of time, in seconds, during which packets are allowed to be received. A *rate limit* is defined as the maximum number of TCP connection requests (for TCP rate limiting), or the maximum number of UDP or ICMP packets, that have been received from a particular client within a configured time window.

- When the **fastage** value is configured, the total desired **timewin** is in seconds and the total desired **holddur** is in minutes. Alteon determines the multiple. For more information on these values, see the *Alteon Application Switch Operating System Command Reference*. The total time window is the outcome of the timewin value multiplied by the fastage value.
- When the holddown is not triggered, the session time limit value starts with the total time window and it is decremented by one second until the value is zero (0). When the value is zero, the session time limit value resets to the next total time window value.
- When the holddown is triggered, the session time limit starts with holddown time, and it is decremented after every *x* minutes, where *x* = 2 * 2^*slowage*.

Holddown Calculation

hold_down = *holddur* X *slowage_time*

where

- *holddur* = the value entered using `/cfg/slb/filt <filter number> /adv/security/ratelim/holddur`
- *slowage_time* = 2 X 2^*slowage*

Time Window Calculation

Total_time_window = *timewin* X 2^(-*x*)

where *x* is the fastage value. By default, the fastage value is 0.

Holddown Periods

Alteon monitors the number of new TCP connections (for TCP rate limiting) or UDP/ICMP packets received (for UDP/ICMP rate limiting). When the number of new connections or packets exceeds the configured limit, any new TCP connection requests or UDP/ICMP packets from the client are blocked. When blocking occurs, the client is said to be *held down*. The client is held down for a specified number of minutes, after which new TCP connection requests or packets from the client are allowed once again to pass through.

Note: The time window and hold duration can be configured individually on a per-filter basis.

The *holddown period* is a multiple of the **slowage** and **holddur** values. For more information about these values, see the *Alteon Application Switch Operating System Command Reference*. The total holddown period is the result of the **holddur** value multiplied by the **slowage** value.

UDP and ICMP Rate Limiting

Alteon filters can be configured to perform rate limiting on UDP and ICMP traffic. Because UDP and ICMP are stateless protocols, the maximum threshold (the **maxcon** command) should be interpreted as the maximum number of packets received from a particular client IP address.

When the maximum threshold has been reached before the time window has expired, all packets of the configured protocol are dropped until the configured holddown (**holddur**) period has expired.

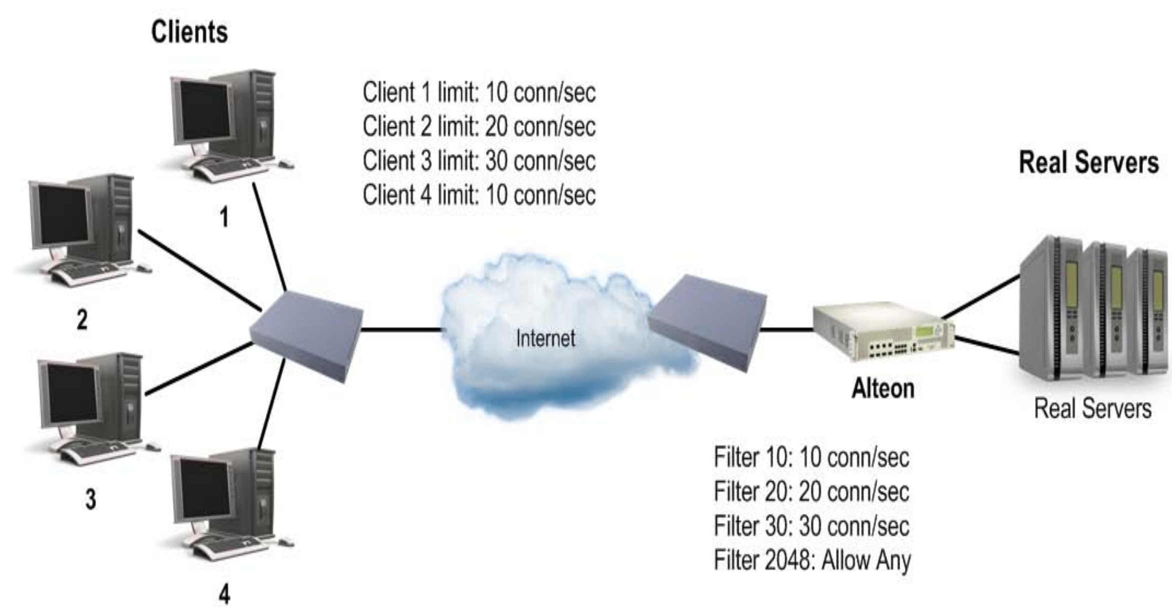
TCP Rate Limiting

Alteon monitors new TCP connections by looking for incoming SYN packets that match a specified TCP rate filter. The first SYN packet to match the filter creates a TCP rate session in the session table. Subsequent SYN packets from the same client that match the same filter increment the TCP rate session counter. If the counter reaches the threshold value before the TCP rate session ages out, then a holddown period is reached. During the holddown period, no new TCP sessions from this client that match this filter are allowed. After the holddown period ends, the next SYN packet is allowed, and a new TCP rate session is created.

Figure 98 - Configuring Clients with Different Rates, page 613 shows four clients configured for TCP rate limits based on source IP address. Clients 1 and 4 have the same TCP rate limit of 10 connections per second. Client 2 has a TCP rate limit of 20 connections per second. Client 3 has a TCP rate limit of 30 connections per second.

When the rate of new TCP connections from clients 1, 2, 3, and 4 reach the configured threshold, any new connection request from the client is blocked for a pre-determined amount of time. If the client's IP address and the configured filter do not match, then the default filter is applied. The default filter 2048 configured for **Any** is applied for all other connection requests.

Figure 98: Configuring Clients with Different Rates



Configuring Protocol-Based Rate Limiting Filters

Rate limiting filters are supported on TCP, UDP, or ICMP protocols only. Protocol-based rate limiting can be configured for all filter types (allow, deny, redir, sip, and dip) and parameters. Specify the source IP address and mask options in the *Filter Configuration* menu to monitor a client or a group of clients. The destination IP address and mask options are used to monitor connections to a virtual IP address or a group of virtual IP addresses.

The following examples work for any supported protocol-based rate limiting configuration. To specify a rate limiting filter for TCP, UDP, or ICMP, set the protocol on the filter itself, then go into the *Filtering Advanced* menu to set the rate limiting parameters.

Example 1: A Basic Rate Limiting Filter

The following example illustrates how to configure rate limiting for Filter 10.

1. Set the protocol used for the rate limiting filter. Only UDP, ICMP, and TCP protocols are supported for rate limiting.

```
>> Main /cfg/slb/filt 10
>> Filter 10 # proto <any|<number>|<name>>
```

2. Enable rate limiting for the filter.

```
>> # /cfg/slb/filt 10/adv/security/ratelim/ena
```

3. Configure maximum number of connections. The value of 1 indicates a total of 10 TCP connections (or sessions).

```
>> Rate Limiting Advanced# maxconn 3
```

4. Set the time window in seconds.

```
>> Rate Limiting Advanced# timewin 3
```

Note: The rate limit defined in step 3 and step 4 as the maximum number of connections over a specified time window results in 30 TCP connections for every three seconds (or 10 TCP connections per second).

5. Set the **holddur** parameter in minutes.

```
>> Rate Limiting Advanced# holddur 4
```

If a client exceeds the rate limit, then the client is not allowed to make any new TCP connections or UDP/ICMP packets for 4 minutes. The following two configuration examples illustrate how to use protocol-based rate limiting to limit user access based on source IP address and virtual IP address.

6. Repeat step 1 through step 5 to configure other filters.
7. Apply and save the configuration.

Example 2: A Rate Limiting Filter Based on Source IP Address

This example illustrates how to define a filter that limits clients with IP address 30.30.30.*x* to a maximum of 150 TCP connections or 150 UDP or ICMP packets per second.

1. Configure the filter as follows.

<table>
<tr><td>>> # /cfg/slb/filt 100/ena</td><td>(Enable the filter)</td></tr>
<tr><td>>> Filter 100 # sip 30.30.30.0</td><td>(Specify the source IP address)</td></tr>
<tr><td>>> Filter 100 # smask 255.255.255.0</td><td>(Specify the source IP address mask)</td></tr>
<tr><td>>> Filter 100 # proto <any|<number>|<name>></td><td>(Specify TCP, UDP or ICMP protocol)</td></tr>
<tr><td>>> Filter 100 # adv/security/ratelim</td><td>(Select the Rate Limiting Advanced menu)</td></tr>
<tr><td>>> Rate Limiting # ena</td><td>(Enable rate limiting on TCP)</td></tr>
<tr><td>>> Rate Limiting # maxconn 15</td><td>(Specify the maximum connections in multiples of 10)</td></tr>
<tr><td>>> Rate Limiting # timewin 1</td><td>(Set the time window in seconds)</td></tr>
<tr><td>>> Rate Limiting # holddur 10</td><td>(Set the hold duration in minutes)</td></tr>
</table>

— Time window = 1 second

— Hold duration = 10 minutes

— Max rate = maxconn/timewin = 150 connections/1 second = 150 connections/second

2. Apply and save the configuration.

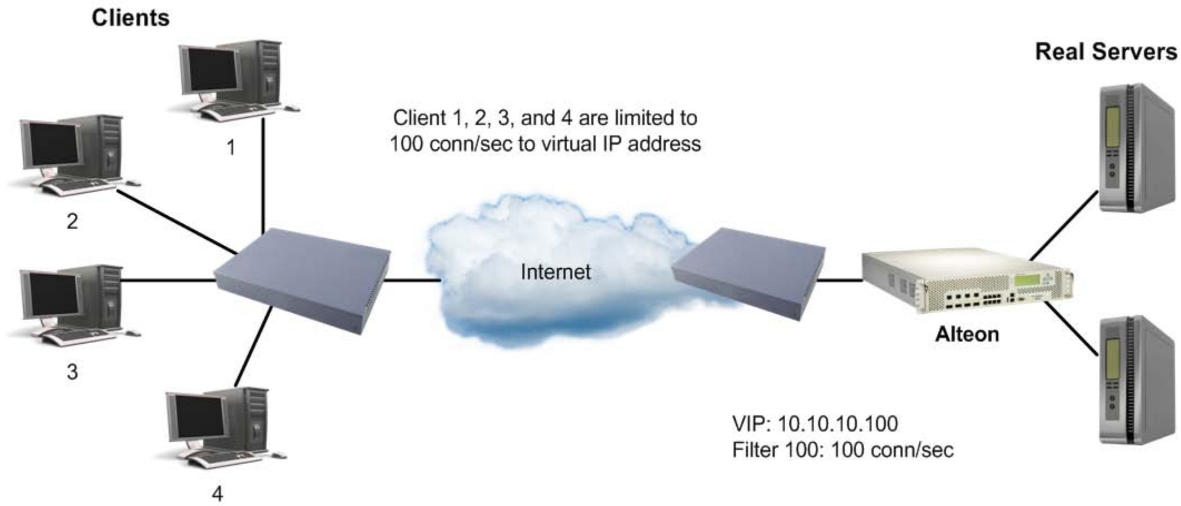
Any client with source IP address equal to 30.30.30.*x* is allowed to make 150 new TCP connections (or UDP/ICMP packets) per second to any single destination. When the rate limit of 150 is met, the hold duration takes effect. The client is not allowed to transmit sessions or connections to the same destination for 10 minutes.

Example 3: A Rate Limiting Filter Based on Virtual Server IP Address

This example defines a filter that limits clients to 100 TCP connections per second or 100 UDP or ICMP sessions per second to a specific destination (VIP 10.10.10.100). Once a client exceeds that limit, the client is not allowed to initiate new TCP connection requests or send UDP or ICMP traffic to that destination for 40 minutes.

Figure 99 - Limiting User Service to a Server, page 616 illustrates how to use this feature to limit client access to a specific destination:

Figure 99: Limiting User Service to a Server



1. Configure the following:

`>> # /cfg/slb/filt 100/ena`	(Enable the filter)
`>> Filter 100 # dip 10.10.10.100`	
`>> Filter 100 # dmask 255.255.255.255`	
`>> Filter 100 # proto <any|<number>|<name>>`	(Specify TCP, UDP or ICMP protocol)
`>> Filter 100 # adv/security`	(Select the *Security* menu)
`>> Security# ratelim ena`	(Enable rate limiting)
`>> Security# maxconn 20`	(Specify the maximum connections in multiples of 10)
`>> Security# timewin 2`	(Set the time window for the session)
`>> Security# holddur 40`	(Set the hold duration for the session)

— Time window = 2 seconds
— Holddown time = 40 minutes
— Max rate = maxconn/time window = 100 connections/second
— 200 connections/2 seconds = 100 connections/second

This configuration limits all clients to 100 new TCP (or UDP/ICMP packets) per second to the server. If a client exceeds this rate, then the client is not allowed to transmit sessions or connections to the virtual server for 40 minutes.

2. Add the filter to the ingress port.

```
>> Rate Limiting # /cfg/slb/port 2/filt ena/add 100
```

3. Apply and save the configuration.

Example 4: A Rate Limiting Filter to Thwart Ping Flooding

This example shows how to define a filter that limits the amount of ICMP pings to any destination behind Alteon. A ping flood attempts to overwhelm servers with ping packets, thus removing it from service while it attempts to reply to every ping.

1. Configure the following filter.

`>> # /cfg/slb/filt 30/ena`	
`>> Filter 30 # proto icmp`	(Specify ICMP protocol)
`>> Filter 30 # action allow`	(Allow ICMP traffic)
`>> Filter 30 # adv/security`	(Select the *Security* menu)
`>> Security# ratelim ena`	(Enable rate limiting)
`>> Security# maxcon 10`	(Specify the maximum connections in multiples of 10)

2. Add the filter to the ingress port.

`>> Rate Limiting # /cfg/slb/port 2`	(Select the appropriate ingress port)
`>> SLB port 2# filt ena`	(Enable filtering on the port)
`Current port 2 filtering: disabled`	
`>> New port 2 filtering: enabled`	
`>> SLB port 2# add 30`	(Add the rate limit filter to the port)
`>> Security# maxcon 10`	

3. Apply and save the configuration.

Protection Against UDP Blast Attacks

Malicious attacks over UDP protocol ports are a common way to bring down real servers. Alteon can be configured to restrict the amount of traffic allowed on any UDP port, thus ensuring that back-end servers are not flooded with data.

In the CLI, you specify a series of UDP port ranges and the allowed packet limit for that range. When the maximum number of packets per second is reached, UDP traffic is shut down on those ports.

To configure UDP blast protection

1. Configure the UDP port numbers or ranges of UDP ports that you want to protect against UDP attacks.

   For example, configure UDP ports 1001-2000 @ 1000pps, UDP ports 2001-4000 @2000pps, and UDP ports 4001-6000 @5000pps.

```
>> /cfg/security/udpblast
>> UDP Blast Protection# add
Enter UDP port number (1 to 65535) or range (first-last): 1001-2000
Enter max packet rate per second (1 to 20000000):         1000
>> UDP Blast Protection# add
Enter UDP port number (1 to 65535) or range (first-last): 2001-4000
Enter max packet rate per second (1 to 20000000):         2000
>> UDP Blast Protection# add
Enter UDP port number (1 to 65535) or range (first-last): 4001-6000
Enter max packet rate per second (1 to 20000000):         5000
```

   Alteon supports up to 5000 UDP port numbers, using any integer from 1 to 65535. For the entire port range, the difference between the highest port number and the lowest port number must be less than or equal to 5000.

2. Enable UDP blast protection on the ports that are connected to unsafe networks.

```
>> /cfg/security/port 1/udpblast ena
```

3. Apply and save the configuration.

TCP or UDP Pattern Matching

This feature provides the capability to scan ingressing packets for patterns contained in some well-known TCP or UDP attacks on back-end servers. Alteon can be configured with one or more filters that scan the first IP packet, and drop if it finds one or all of the configured patterns. If no match is found, the packets are allowed through.

Pattern matching is constructed much in the same way as any other filter configured to examine Layer 7 content.

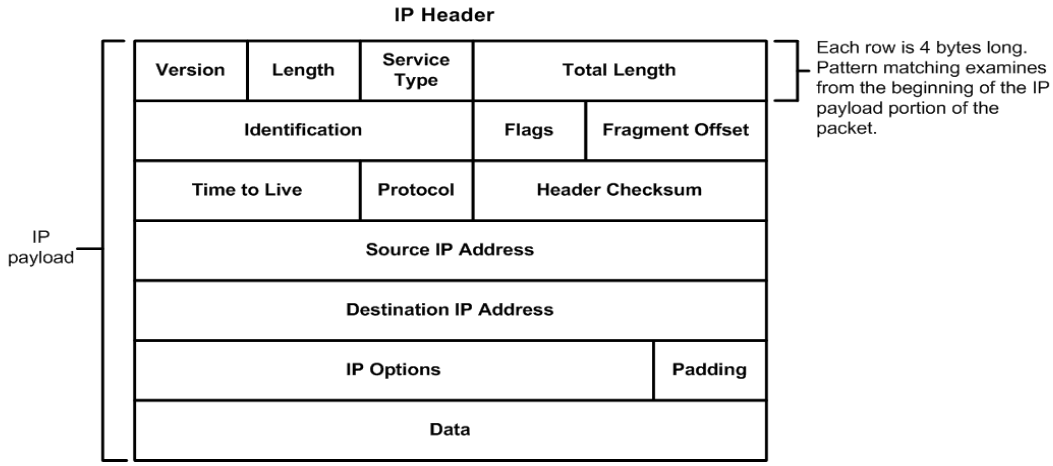
Note: The ability to match and perform filter action on a pattern or group of patterns is available only when you enable the Security Pack software.

Pattern Criteria

Many TCP or UDP attacks contain common signatures or patterns in the IP packet data. Alteon can be configured to examine an IP packet from either the beginning, from a specific offset value (starting point) within the IP packet, and/or from a specified depth (number of characters) into the IP packet. It then performs a matching operation.

Figure 100 - IP Packet Format, page 619 illustrates an IP packet format. Alteon is able to track from the beginning of the IP packet (at the IP version number), through an IP packet payload of 1500 bytes. Each row in an IP packet is four bytes.

Figure 100: IP Packet Format



To enter pattern criteria

```
>> /cfg/slb/layer7/slb/addstr
```

Table 53 includes an explanation of values you are prompted to provide:

Table 53: Pattern Criteria Values

Value	Description
Pattern	A pattern can be a regular expression string pattern in ASCII characters, or a binary pattern in hexadecimal notation. For more information on using regular expressions to match pattern data, see Regular Expression Matching, page 803. If the pattern is binary, specify the binary pattern in hexadecimal notation. For example, to specify the binary pattern **1111 1100 0010 1101**, enter **FC2D**.
Offset	An offset value is the byte count from the start of the IP header, from which a search or compare operation is performed. An offset value is always required when the creating pattern strings, even if the desired value is zero (0). For example, if an offset of 12 is specified, Alteon starts examining the hexadecimal representation of a binary string from the 13th byte. In the IP packet, the 13th byte starts at the source IP address portion of the IP payload.

Table 53: Pattern Criteria Values

Value	Description
Depth	Depth is the number of bytes in the IP packet that should be examined from either the beginning of the packet or from the offset value. For example, if an offset of 12 and a depth of 8 is specified, the search begins at the 13th byte in the IP packet, and matches 8 bytes. An offset of 12 and depth of 8 encompasses the source IP address and destination IP address fields in the IP payload. If no depth is specified in ASCII matches, the exact pattern is matched from the offset value to the end of the pattern. A depth must be specified for binary matches that are larger than the pattern length in bytes.
Operation	An operation tells Alteon how to interpret the pattern, offset, and depth criteria. • For a *string pattern*, use the operation eq (equals) to match the content of the string. • Use the operations to find values lt (less than), gt (greater than), or eq (equals) to the specified binary value. If no operation is specified, the pattern is invalid. The lt and gt operators can be used for certain attack signatures in which one or more bytes are less than or greater than a certain value.

Matching Groups of Patterns

When a virus or other attack contains multiple patterns or strings, it is useful to combine them into one group and give the group a name that is easy to remember. When a pattern group is applied to a deny filter, Alteon matches any of the strings or patterns within that group before denying and dropping the packet. Up to five (5) patterns can be combined into a single pattern group. Configure the binary or ASCII pattern strings, group them into a pattern group, name the pattern group, and then apply the group to a filter.

The filtering commands enable the administrator to define groups of patterns and place them into groups. By applying the patterns and groups to a deny filter, the packet content can be detected and thus denied access to the network.

Alteon supports up to 1024 pattern groups.

Note: The pattern group matching feature is available only if you have purchased and enabled the Advanced Denial of Service Protection software key.

Alteon supports multi-packet inspection. This allows for the inspection of multiple patterns across multiple packets in a session. Filtering actions will be taken only after matching all the patterns in the same given sequence.

For example, assume a chain consisting of multiple patterns numbered 1 through 4. The incoming packets of the session are first searched for pattern 1. Once pattern 1 of the chain is matched, subsequent packets of the session are searched for pattern 2 and, if matched, pattern 3 is searched for and so on, until all the patterns in the chain are matched. The filter action is taken after patterns 1 through 4 are matched.

Note: A *reset* frame is sent to the destination device when a Layer 7 deny filter is matched instead of waiting for a server side timeout. This releases the TCP connection in the destination device. Similarly, any time a TCP packet is denied, a reset frame is sent.

Matching and Denying a UDP Pattern Group

The following is an example configuration for matching an denying a UDP pattern group.

To match and deny a UDP pattern group

1. Configure a list of SLB strings containing binary patterns and offset pairs.

   This example illustrates adding one binary pattern and one ASCII string pattern. The binary pattern is written in hexadecimal notation.

`>> /cfg/slb/layer7/slb/addstr`	
`Enter type of string [l7lkup|pattern]: pattern`	(Add the first pattern)
`Enter match pattern type [ascii|binary]: binary`	(Select binary matching)
`Enter HEX string:  014F`	(For this binary pattern)
`Enter offset in bytes from start of IP frame (0-1500): 2`	(Starting from third byte)
`Enter depth in bytes to search from offset (0-1500): 0 )`	(Search length of the pattern
`Enter operation (eq|gt|lt): eq`	(For values equal to this binary pattern)
`>> Server Loadbalance Resource# add`	(Add the second pattern)
`Enter type of string [l7lkup|pattern]: pattern`	
`Enter match pattern type [ascii|binary]: ascii`	(Select ASCII matching)
`Enter ASCII string:   /default.htm )`	
`Enter offset in bytes from start of IP frame (0-1500): 44`	(Match this ASCII string)
`Enter depth in bytes to search from offset (0-1500): 30`	(Search from 45th byte)
	(through the 30th byte)

2. Identify the IDs of the defined strings.

```
>> Server Loadbalance resource# cur
```

   The strings in **bold** are used in this example. Number of entries: 10

ID	SLB String
1	ida
2	%c1%9c
3	%c0%af
4	playdog.com
6	HTTPHDR:Host:www.playdog.com
7	HTTPHDR:SoapAction=*
8	BINMATCH=014F, offset=2, depth=0, op=eq, cont 256
9	STRMATCH=/default.htm offset=44, depth=30, op=eq, cont 256

3. From the *Security* menu, configure a pattern group and name it something relevant and easy to remember.

`>> /cfg/security/pggroup 1/name`	(Name Pattern Group 1)
`>> /cfg/security/pggroup 1/name`	
`>> /cfg/security/pggroup 1/name`	(Name the group)

4. Add the new pattern/offset pairs to the pattern group using their ID numbers.

   Refer back to step 2, where you typed the `cur` command, if you need to recall the ID number associated with the SLB string.

`>>Pattern Match Group 1# add 8`	(Add the first binary pattern)
`>>Pattern Match Group 1# add 8`	(Add the ASCII string pattern)

5. Configure a filter and its appropriate protocol in which the patterns are found.

```
>>/cfg/slb/filt 90
>>Filter 90  # proto tcp
```

6. Configure the filter source and destination ports.

```
>>Filter 90  # sport any
>>Filter 90  # dport http
```

7. Configure the filter to deny.

```
>>Filter 90 # action deny
Current action: none
Pending new action:     deny
```

8. Apply the pattern group you configured in step 3 and step 4 to the filter.

```
>>Main# /cfg/slb/filt 90/adv/security/addgrp 1
>>Group ID 1 added.
```

9. Enable pattern matching on the filter. This command enables Layer 7 lookup on the filter.

```
>>/cfg/slb/filt 90/adv/security/pmatch enable
Current Pattern Match: disabled
New Pattern Match:     enabled
```

10. Apply the filter to the client port. If the incoming client requests enter Alteon on port 3, then add this filter to port 3.

`>> # /cfg/slb/port 3`	(Select the client port)
`>> SLB Port 3# filt ena`	(Enable filtering on the client port)
`>> SLB Port 3# add 90`	(Add Filter #90 to the client port)

11. Apply and save the configuration.

Matching All Patterns in a Group

Alteon is capable of matching on all patterns in a pattern group before the filter denies a packet. Use the **matchall** command to instruct the filter to match all patterns in the group before performing the deny action.

Note: The matchall command is configurable only for binary or ASCII patterns added to pattern groups (pgroup). It does not apply to l7lkup filter strings configured with the **/cfg/slb/layer7/slb/addstr** command.

To match all patterns in a group

1. Use the base configuration in Matching and Denying a UDP Pattern Group, page 621.
2. In the *Filter* menu, enable the matching of all criteria.

```
>> /cfg/slb/filt 90/adv/security/matchall ena
>> SLB Port 3# add 90
```

Now, both patterns configured in Matching and Denying a UDP Pattern Group, page 621 must be matched before a packet is denied and dropped.

ID	SLB String
8 9	BINMATCH=014F, offset=2, depth=0, op=eq, cont 256 STRMATCH=/default.htm offset=44, depth=30, op=eq, cont 256

3. Apply and save the configuration.

Matching and Denying Large Packets—ICMP Ping of Death Example

A ping of death attack sends fragmented ICMP echo request packets. When these packets are reassembled, they are larger than the 65536 byte packets allowed by the IP protocol. Oversized packets cause overflows in the server's input buffer, and can cause a system to crash, hang, or reboot.

Large ICMP packets, such as in an ICMP ping of death attack, can be blocked using a deny filter combined with binary patterns used to filter non-zero IP offsets or More-Fragment bits sent in the IP flags.

An IP packet is determined to be an IP fragment if one the following occurs:

- The 13-bit fragment offset field in the IP header is non-zero
- The More-Fragments bit in the 3-bit flags field in the IP header is set.

The flags field begins at the seventh byte of the IP packet, and the fragment offset is right after this field. The two fields taken together occupy a total of two (2) bytes. By searching for values greater than 0000 and less than 4000, Alteon searches for either of these conditions, or both.

To match and deny large packets

This configuration is similar to the examples in Matching and Denying a UDP Pattern Group, page 621 and Matching All Patterns in a Group, page 623.

1. Create an SLB string pattern that filters non-zero IP offsets. Enter the value in hexadecimal notation.

```
>> /cfg/slb/layer7/slb/addstr
Enter type of string [l7lkup|pattern]: pattern          (Add the pattern)
Enter match pattern type [ascii|binary]: binary         (Select binary matching)
Enter HEX string:    0000                     (non-zero IP offset)
Enter offset in bytes from start of IP frame (0-1500):  (Search from seventh byte)
6
Enter depth in bytes to search from offset (0-1500): 0  (Through end of pattern)
Enter operation (eq|gt|lt): gt                          (For values greater than 0000)
```

2. Create another SLB string pattern that filters More-Fragments.

```
>> Server Loadbalance Resource# add
Enter type of string [l7lkup|pattern]: pattern          (Add the pattern)
Enter match pattern type [ascii|binary]: binary         (Select binary matching)
>> Enter HEX string: 4000                               (More-Fragments bit set)
Enter offset in bytes from start of IP frame (0-1500):  (Search from seventh byte)
6
Enter depth in bytes to search from offset (0-1500): 0  (Through end of pattern)

Enter operation (eq|gt|lt):  lt                         (For values less than 4000)
```

3. Apply the new configuration.

```
>> Server Loadbalance Resource# apply
```

4. Identify the IDs of the defined patterns.

```
>> Server Loadbalance Resource# apply
```

The strings in **bold** are used in this example. Number of entries: 11

ID	SLB String
1	ida
2	%c1%9c
3	%c0%af
4	playdog.com
6	HTTPHDR:Host:www.playdog.com
7	HTTPHDR:SoapAction=*

ID	SLB String
8	BINMATCH=014F, offset=2, depth=0, op=eq, cont 256
9	STRMATCH=/default.htm offset=44, depth=30, op=eq, cont 256
10	BINMATCH=0000, offset=6, depth=0, op=gt, cont 256
11	BINMATCH=4000, offset=6, depth=0, op=lt, cont 256

5. In the *Security* menu, configure a pattern group and name it something relevant and easy to remember.

```
>> /cfg/security/pgroup 2/name
Current pattern group name:
Enter new pattern group name: pingofdeath
```

6. Add the defined patterns to the pattern group.

```
>> Pattern Match Group 2# add 10
>> Pattern Match Group 2# add 11
```

7. Configure a filter and its appropriate protocol in which the patterns are found. In this case, the ICMP protocol should be specified.

```
>> /cfg/slb/filt 190
>> Filter 190 # proto icmp
```

8. Set the filter action to deny.

```
>> Filter 190 # action deny
Current action: none
Pending new action:     deny
```

9. Set the ICMP message type. Ping of Death uses the ICMP message type **echoreq**.

```
>> Filter 190  # adv/icmp
>> Filter 190 Advanced# icmp
Current ICMP message type:      any
Enter ICMP message type or any: echoreq
```

10. Apply the pattern group you configured in step 5 and step 6 to the filter.

```
>> Filter 190 # security/addgrp 2
Group ID 2 added.
```

11. Enable pattern matching on the filter.

```
>> /cfg/slb/filt 190/adv/security/pmatch enable
Current Pattern Match: disabled
New Pattern Match:     enabled
```

12. Enable matchall criteria so that the filter matches on all patterns in the pattern group.

```
>> Security# matchall ena
Current Match-all Criteria: disabled
New Match-all Criteria:     enabled
```

13. Apply the filter to the client port. This example assumes a client connection on port 22.

`>> # /cfg/slb/port 22`	(Select the client port)
`>> SLB Port 22# filt ena`	(Enable filtering on the client port)
`>> SLB Port 22# add 190`	(Add Filter #190 to the client port)

14. Apply and save the configuration.

FlexiRules for SIP over UDP Traffic

FlexiRules control the SIP over UDP traffic going through Alteon, and enhances the SIP security in the network. They enable administrators to customize the security policies and set rules. These rules monitor the SIP calls and gives the SIP engine the ability to dynamically filter SIP traffic. FlexiRules work along with filters to provide in-depth security to SIP over UDP application servers.

The following are the functions of the SIP UDP rules:

- Deny traffic based on content match
- Rate limit based on content match
- Monitor SIP Uniform Resource Identifiers (URI)

FlexiRules for SIP over UDP are advanced pattern match filters. Multiple rules can be configured. The severity level can be set from 1 to 5, where 1 is the highest severity. Selection is based on severity when multiple rules are hit.

The following inputs define FlexiRules for SIP over UDP:

- Header field name and content
- Bandwidth Management (BWM) contract for the rule
- Alert message display
- Severity
- Dependent rules

There are two modes set by the SIP rules in a session entry:

- Monitor Mode, page 626
- Dependent Mode, page 627

Monitor Mode

In monitor mode, Alteon dumps the SIP header information to the Management Processor (MP) for analysis. This dump can be used for troubleshooting.

To enable monitor mode

You enable the monitor in the contract.

```
/cfg/bwm/cont <x>/mononly ena
```

The following is an example set of monitoring messages that are displayed on the console:

```
10:10.1.1.10:5060->10.1.1.21 mrid 1 from_has_bob
cid 54A5E6ED-B154-4A22-A59B-E
f sam <sip:sam@ocs2007.com>
t <sip:bob@ocs2007.com>
```

Dependent Mode

You can configure two dependent rules for a rule. When rules contain dependent rules, the rule is matched only when its dependent rules are matched. It checks only the dependent rules for a match.

Alteon is in the inspection path until it finds a match. When multiple rules are matched, Alteon takes the action of the highest severity rule. If the highest severity rule contains dependent rules, and if the dependent rules are not matched, Alteon takes the action of the next highest severity rule that does not contain dependent rules. Alteon takes the action of the highest severity rule only when all its dependent rules are matched.

Configuring the FlexiRules

The following is an example configuration FlexiRules.

To configure FlexiRules

1. Create the rule.

```
/cfg/slb/layer7/rule <1 to 100>
```

2. Define the rule.

```
/cfg/slb/layer7/rule 1/hdrfld
from|to|replyto|via|method|reqline|callid|cseq|contact|expires|contentlen|sdpco
ntent
```

3. Define the content of the header field name.

```
/cfg/slb/layer7/rule 1/content bob
```

4. Define the severity (1 to 5)

```
/cfg/slb/layer7/rule 1/severity 1
```

5. Assign contract for this rule (1 to 1024). For information about creating contracts, see Bandwidth Management, page 761.

```
/cfg/slb/layer7/rule 1/contract 2
```

6. Define the message. This message appears in the log when the rule is matched.

```
/cfg/slb/layer7/rule 1/message "from Bob"
```

7. Enable the rule.

```
/cfg/slb/layer7/rule 1/ena
```

8. Enable SIPs in the filter.

```
/cfg/slb/filt/adv/layer7/sip/sips ena
```

9. Enable pattern matching in the filter.

```
/cfg/slb/filt/adv/security/pmatch ena
```

10. Add the filter on the port. Enable filter on the server port if reverse lookup for SIP UDP rule is configured.

```
/cfg/slb/port <port number>/filt ena/add <filter number>
```

Example Configuration of FlexiRules

1. Configure contracts.

`/cfg/bwm`	(Select BWM)
`on`	(Enable BWM)
`/cfg/bwm/cont 1`	(Select the contract)
`ena`	(Enable the contract)
`pol 1`	(Set contract policy)
`/cfg/bwm/pol 1`	(Select the policy)
`hard 0k`	(Set the hard limit)
`soft 0k`	(Set the soft limit)
`resv 0k`	(Set the reservation limit)
`userlim 0k`	(Set the user limit)

2. Create Rule 1.

`/cfg/slb/layer7/rule 1`	(Select Rule 1)
`ena`	(Enable Rule 1)
`hdrfld from`	(Enter the header field name)
`content “bob”`	(Enter the content of the header field)
`message “from_has_bob”`	(Enter the alert message)
`contract 1`	(Select the contract)
`severity 3`	(Select the highest severity)

3. Create rule 99.

`/cfg/slb/layer7/rule 99`	(Select Rule 99)
`ena`	(Enable Rule 99)
`hdrfld to`	(Enter the header field name)
`content “Sam”`	(Enter the content of the header field)
`message “to_is_sam”`	(Enter the alert message)
`severity 5`	(Select the severity)

4. Create rule 100.

`/cfg/slb/layer7/rule 100`	(Select Rule 100)
`ena`	(Enable Rule 100)
`hdrfld sdpcontent`	(Enter the header field name)
`content "string"`	(Enter the content of the header field)
`message "domain is alteon"`	(Enter the alert message)
`severity 4`	(Select the severity)

5. Add dependent rules 99 and 100 to rule 1.

```
addrule 100
addrule 99
```

After creating the rules, when Bob calls Sam, Rule 1 and Rule 99 are matched and Alteon takes the action of Rule 99. Alteon takes the action of Rule 1 only when Rule 100 is also matched. Until rule 100 is matched in the return traffic, Alteon rate limits the traffic according to Rule 99.

The following is an example of the logs:

```
Nov 12 19:27:33 NOTICE  security: 10:10.1.1.10:5060->10.1.1.21 rid1 deny
from_has_bob
```

Chapter 22 – WAN Link Load Balancing

WAN link load balancing lets you configure Alteon to balance user session traffic among a pool of available WAN Links. The following sections in this chapter provide conceptual information on WAN Link Load balancing:

- Multi-homing, page 631—Provides an overview of WAN link load balancing and its benefits.
- How WAN Link Load Balancing Works, page 633—Discusses in detail the path of the outbound and inbound traffic in a WAN link load balancing environment.
- Configuring WAN Link Load Balancing, page 637—Provides step-by-step procedures to configure Alteon for load balancing the WAN links in different environments, including:
  - Example 1: Simple WAN Link Load Balancing, page 639
  - Example 2: WAN Link Load Balancing with Server Load Balancing, page 646
- Health Checking and Multi-homing, page 655—Discusses the interaction between health checking WAN link load balancing, and the steps needed to avoid service disruption.

For additional information on WAN link commands see the *Alteon Application Switch Operating System Command Reference*.

Although WAN link load balancing supports most IPv4 protocols, the following protocols may not be supported in typical implementations:

- IPv6-related protocols
- IP-within-IP Encapsulation Protocol (IPIP)
- Generic Routing Encapsulation (GRE)
- Encap Security Payload/Authentication Header (ESP/AH)

WAN link load balancing supports the following metrics:

- Response time
- Bandwidth
- Least connections
- Round-robin

When the response time or bandwidth metrics are used, Alteon calculates weights and uses the round-robin metric for selecting the Internet service providers (ISPs).

Multi-homing

WAN link load balancing enables Alteon to provide gigabit connectivity from corporate resources to multiple ISP links to the Internet.

To handle the high volume of data on the Internet, corporations may use more than one ISP as a way to increase reliability of Internet connections. Such enterprises with more than one ISP are referred to as being *multi-homed*. In addition to reliability, a multi-homed network architecture enables enterprises to distribute load among multiple connections and to provide more optimal routing.

Multi-homing has become essential for reliable networks, providing customers with protection against connectivity outages and unforeseen ISP failures. Multi-homing also presents other clear opportunities for enterprises to intelligently manage how WAN links are used. With link load balancing, organizations have greater flexibility to scale bandwidth and reduce spending for corporate connectivity.

Alteon provides a solution for enterprises to optimize use of Internet connectivity. This comprehensive solution helps enterprises to direct traffic over the best connection to maximize performance, maximize corporate bandwidth investments, and effectively remove existing deployment and management barriers for multi-homed networks.

Benefits of WAN Link Load Balancing

Traditionally, corporations have used Border Gateway Protocol (BGP) to determine the optimal path of the WAN link for load balancing traffic. However, Table 54 shows the advantages of implementing WAN link load balancing versus using BGP.

Table 54: WAN Link Load Balancing Versus BGP

WAN Link Load Balancing	BGP
• Easy to configure • Redundancy—If one of the ISP links go down, then the other ISP link takes over. • Backup—You can use a low speed ISP link as a backup for a high speed ISP link. • If ISP reaches its session limit, Alteon deletes it from the group. • Easy to manage	• Complex to implement • Laborious to manage • Difficult to get an autonomous system number • Does not allow you to monitor the WAN links for load, speed, or health of devices on the other end of the link

WAN link load balancing benefits your network in a number of ways:

- **Performance is improved by balancing the request load across multiple WAN links**—More WAN links can be added at any time to increase processing power.
- **Increased efficiency for WAN link use and network bandwidth**—Alteon is aware of the shared services provided by your WAN link pool and can then balance user traffic among the available WAN links. Important WAN link traffic gets through more easily, reducing user competition for connections on overused links. For even greater control, traffic is distributed according to a variety of user-selectable rules.
- **Increased reliability**—Reliability is increased by providing multiple paths from the clients to Alteon and by accessing a pool of WAN links. If one WAN link fails, the others can take up the additional load.
- **Increased scalability of services**—As traffic increases and the WAN link pool's capabilities are saturated, new WAN links can be added to the pool transparently.
- **Ease of maintenance**—WAN links can be added or removed dynamically, without interrupting traffic.

Identifying Your Network Needs

WAN link load balancing addresses the following vital network concerns:

- A single WAN link no longer meets the demand for increased traffic.
- The connection from your LAN to the Internet overloads the WAN link's capacity.
- Your WAN links must remain available even in the event of a link failure.
- Your WAN links are being used as a way to do business and for taking orders from customers. It must not become overloaded or unavailable.
- You want to use multiple WAN links or hot-standby WAN links for maximum server uptime.
- You must be able to scale your applications to meet client and LAN request capacity.
- You cannot afford to continue using an inferior load-balancing technique, such as DNS round-robin or a software-only system.

What is Load Balancing?

Alteon acts as a front-end to the WAN links, interpreting user session requests and distributing them among the available WAN links. Load balancing in Alteon can be done in the following ways:

- **Filtered-based load balancing**—A filter allows you to control the types of traffic permitted through Alteon. Filters are configured to allow, deny, or redirect traffic according to the IP address, protocol, or Layer 4 port criteria. In filtered-based load balancing, a filter is used to redirect traffic to a real server group. If the group is configured with more than one real server entry, redirected traffic is load balanced among the available real servers in the group.

  WAN links use redirection filters to load balance outbound traffic. For more information, see Outbound Traffic, page 633.
- **Virtual server-based load balancing**—This is the traditional load balancing method. Alteon is configured to act as a virtual server and is given a virtual server IP address (or range of addresses) for each collection of services it will distribute. There can be as many as 1024 virtual servers, each distributing up to eight different services (up to a total of 1023 services).

  Each virtual server is assigned a real server. When the user stations request connections to a service, they will communicate with an Alteon virtual server. When Alteon receives the request, it binds the session to the IP address of the corresponding real server and remaps the fields in each frame from virtual addresses to real address.

  This method of load balancing is used to load balance inbound traffic. For more information, see Inbound Traffic, page 634.

How WAN Link Load Balancing Works

To effectively use multiple ISP links, Radware recommends that both outbound and inbound traffic is load balanced using Alteon. Alteon can be configured to load balance up to eight ISP links. Alteon regularly checks the health of the upstream routers and measures the condition of the link. When traffic is to be sent to the link, Alteon chooses the most optimal link for that session.

This section explains how WAN link load balancing works differently for:

- Outbound Traffic, page 633
- Inbound Traffic, page 634

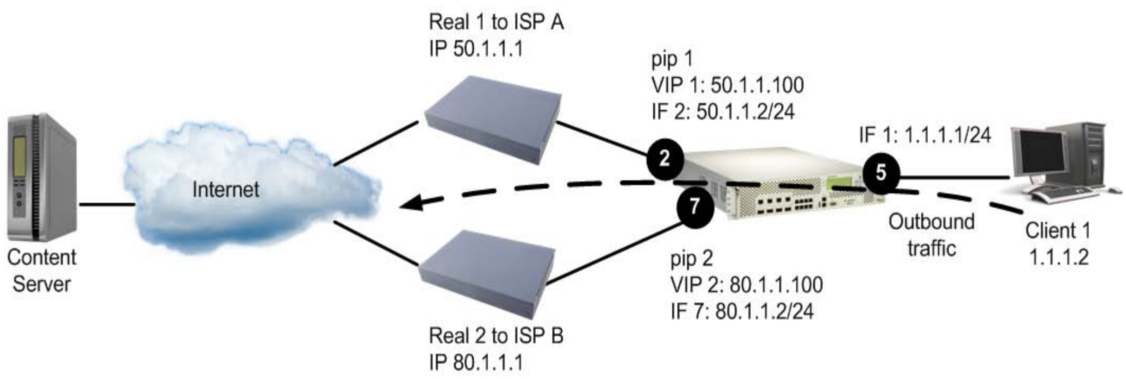
Outbound Traffic

Outbound traffic is data from the intranet that accesses content across the Internet. Alteon load balances outbound traffic using redirection filters to redirect traffic initiated from within the user's network to a group of devices that exist at the other end of the WAN link. These filters determine which link is the best at the time the request is generated.

The design of outbound WAN link load balancing is identical to standard redirection, except that Alteon substitutes the source IP address of each frame with the proxy IP address of the port to which the WAN link is connected. This substitution ensures that the returning response traverses the same link.

In Figure 101 - WAN Link Load Balancing for Outbound Traffic, page 634, client 1 at IP address 1.1.1.2 sends an HTTP request to the Internet. Outbound traffic from client 1 reaches port 5 on the Alteon which is configured with a redirection filter for link load balancing. The traffic is load balanced between ports 2 and 7 depending on the metric of the WAN group (configured as real servers 1 and 2).

Figure 101: WAN Link Load Balancing for Outbound Traffic



The outbound traffic resolution in Figure 101 - WAN Link Load Balancing for Outbound Traffic, page 634 is described as follows:

1. Client 1 makes a data request for content on the Internet.
2. When the request reaches port 5, the redirection filter is triggered and Alteon selects the optimal WAN link.
3. Before the packets leave the WAN link ports, the client IP address is substituted with the configured proxy IP address on port 2 or 7. Proxy IP address maintains persistency for the returning request.
4. Alteon sends the request to the destination IP address.
5. The returning request from the Internet uses the same WAN link because the destination IP address responds to the proxy IP address, thereby maintaining persistency. The selected ISP processes the packet.
6. Alteon converts the proxy IP address to the client IP address and the request is returned to the client.

Inbound Traffic

Inbound traffic is data from an external client on the Internet that enters Alteon to access an internal service, such as corporate Web servers or FTP servers.

Alteon lets you load balance the inbound traffic by providing access to the external client with the best available WAN link.

Note: For load balancing inbound traffic, you must have the Inbound Link Load Balancing license installed. For more information on installing licenses see the section on the **/oper/swkey** command in the *Alteon Application Switch Operating System Command Reference*, and the *Radware Alteon Installation and Maintenance Guide*.

This is implemented by configuring Alteon as an authoritative name server. Alteon dynamically determines the best ISP link to use at the time the request is generated. The best link is determined by the configured metric, the load on the ISP, and periodic health checks on the upstream routers. For more information on load-balancing metrics, see Metrics for Real Server Groups, page 180. Real server weighting can also be used to determine the best link when using the hash metric for load balancing inbound WAN links. For more information on real server weighting, see Weights for Real Servers, page 184.

When the external client makes a DNS request, Alteon responds with the IP address of the best available WAN link (ISP).

Tracing the Data Path

In Figure 102 - External Client Accessing Data from a Non-SLB Group, page 635, the client request enters Alteon via ISP A or ISP B. ISP A is configured as real server 1 and ISP B is configured as real server 2. A virtual server IP address is configured for each ISP and each domain. The virtual server IP addresses for each ISP must be configured in the ISP's address range.

As shown in Figure 102 - External Client Accessing Data from a Non-SLB Group, page 635, two virtual server IP addresses (virtual server IP address 1 and virtual server IP address 2) are configured for **radware.com** in each of the ISP's address ranges. Once Alteon responds with the best virtual server IP address, all subsequent traffic from the clients to this domain is sent to the same virtual server IP address, thereby passing through the same ISP.

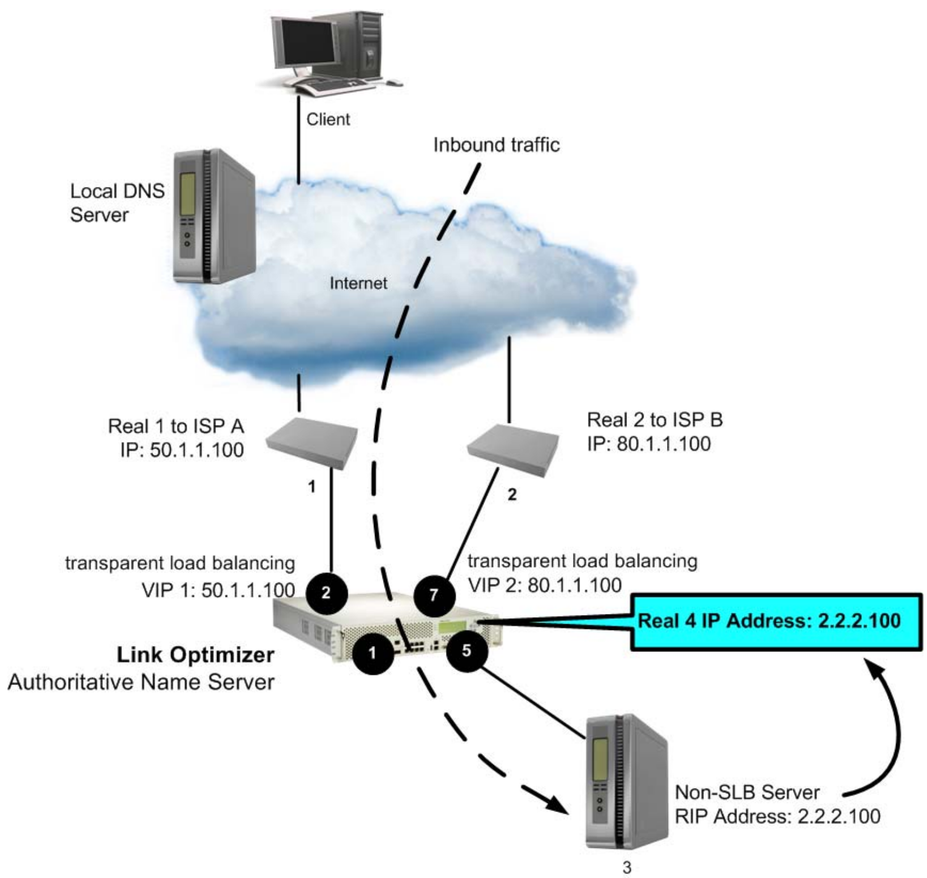
External client request can be one of the following ways:

- External Client Accessing Data from a Non-SLB Group, page 635
- External Client Accessing Data from an SLB Group, page 636

External Client Accessing Data from a Non-SLB Group

In Figure 102 - External Client Accessing Data from a Non-SLB Group, page 635, a client request for **http://www.radware.com** enters Alteon via an ISP. The non-SLB server (real server 3) can be directly or indirectly connected to Alteon. A real server 4 is configured on the Alteon with the IP address of real server 3. Real server 4 is added to a server group and that group is advertised in VIP 1 and VIP 2.

Figure 102: External Client Accessing Data from a Non-SLB Group



The inbound traffic resolution in Figure 102 - External Client Accessing Data from a Non-SLB Group, page 635 is described as follows:

1. The client makes a request to **www.radware.com**.
2. The client query does not exist in the local DNS database. Local DNS queries the Domain Name Server on Alteon.
3. Alteon monitors WAN links and responds with the virtual IP address of the optimal ISP.

Note: Radware recommends default gateways for each ISP VLAN to avoid asymmetric routing.

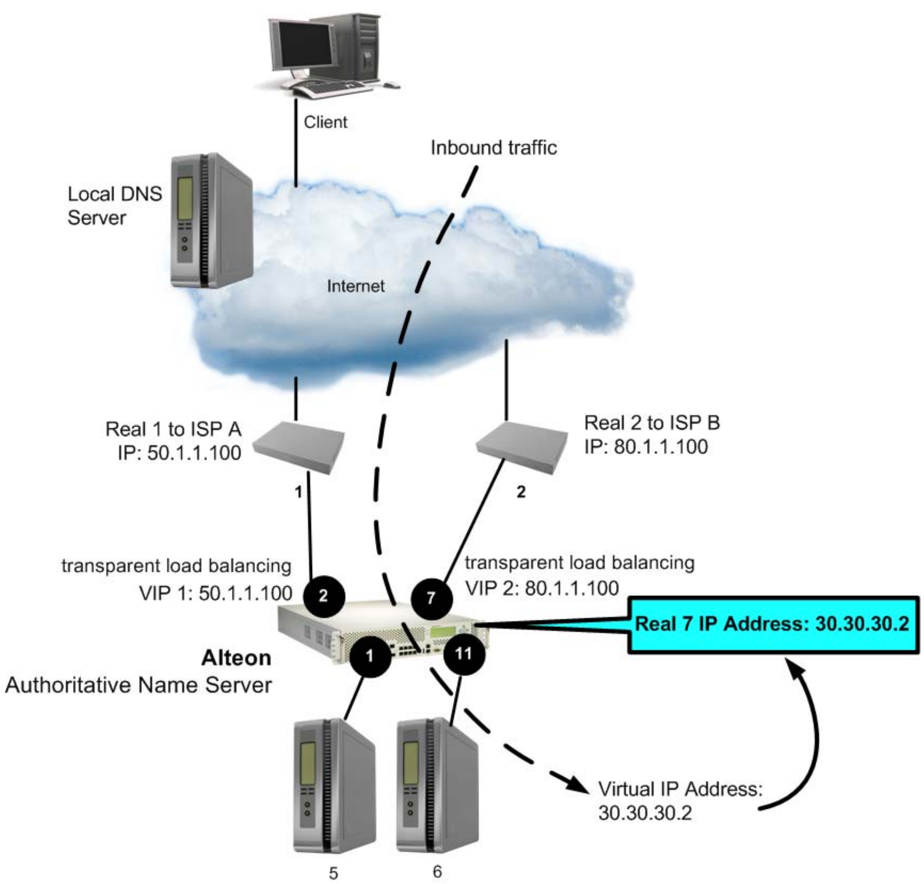
4. The client again requests with the provided virtual IP address.
5. The server responds to the content request.

   An allow filter at port 5 processes the data for the services configured on the server. For example, if the client sends an HTTP request to server 3, then the allow filter should be configured for source port 80. Similarly, if the client sends an SMTP request to server 3, then the allow filter should be configured for source port 25.
6. The transparent load balancing feature on the WAN ports maintains persistency, so that the traffic returns via the same ISP.

External Client Accessing Data from an SLB Group

In Figure 103 - External Client Accessing Data from an SLB Group, page 636, the client request is for **www.radware.com**. The client request should be load balanced between SLB servers 5 and 6.

Figure 103: External Client Accessing Data from an SLB Group



The inbound traffic resolution in Figure 103 - External Client Accessing Data from an SLB Group, page 636 is described as follows:

1. The client makes a request to **www.radware.com**.
2. The client query does not exist in the local DNS database. Local DNS queries the Domain Name Server on Alteon.
3. Alteon monitors WAN links and responds with the virtual IP address of the optimal ISP.
4. The client makes the request again to **www.radware.com** with the provided virtual IP address.
5. The SLB servers respond to the content request, because real server 7 IP address on Alteon is the virtual server address of **www.radware.com**.
6. The session request egresses from port 1 and port 11 of Alteon where it is then load balanced between the SLB servers. The virtual server IP address for the SLB servers on Alteon are configured as a real server IP address (Real 7 IP: 30.30.30.2). Real 7 is added to a group.
7. The returning data from the SLB server reaches port 1, which is enabled for server processing.

   For information on server processing, see Network Topology Requirements, page 169. The transparent load balancing feature on the WAN ports maintains persistency, so that the traffic returns via the same ISP.

Configuring WAN Link Load Balancing

This section describes how to configure Alteon for load balancing the WAN links in different environments:

Before You Begin

The following is required prior to configuration:

- Log into the CLI as the administrator.
- Connect each WAN link to a separate port on Alteon.

  Note: Do not connect two or more WAN links to the same Alteon port using a Layer 2 switch. WAN link load balancing uses the proxy IP address of the destination port when translating the source IP address of the requests.

- Do not configure your WAN link ports into trunk groups.

Configuration Summary

Table 55 summarizes the steps for configuring WAN link load balancing:

Table 55: Configuration Summary

Configuring Outbound Traffic	Configuring Inbound Traffic
1. Configure the basic parameters.This includes configuring VLAN, IP interfaces, and defining gateways per VLAN.	1. Configure the basic parameters.This includes configuring VLAN, IP interfaces, and defining gateways per VLAN.
2. Configure the load balancing parameters for the ISP WAN links. a. Configure the ISP routers as real servers. b. Add it to a group. c. Define the metric and health. d. Enable SLB.	2. Configure the load balancing parameters for the ISP WAN links. a. Configure the ISP routers as real servers. b. Optionally assign weight to real servers. c. Add it to a group. d. Define the metric and health. e. Enable SLB.
3. Configure the WAN link ports. a. Configure a proxy IP address.	3. Configure the WAN link ports. a. Enable client processing. b. Enable transparent load balancing. c. Enable DAM.
4. Configure the outbound client ports. a. Configure the redirection filter and enable it for link load balancing. b. Apply the filter to the client ports.	4. Configure the inbound server ports. a. Create a group with the real servers. b. Enable server processing. c. Enable link load balancing. d. Enable filter processing. A real server is configured for every SLB group. 5. Configure virtual server IP addresses and services for each ISP. For each ISP link, configure a virtual server IP address per domain. 6. Configure Alteon to behave like a Domain Name Server. This involves defining the domain record name and mapping the virtual server and real server addresses (ISP router) for each WAN link.

Note: For details about any of the menu commands described in the following examples, refer to the *Alteon Application Switch Operating System Command Reference*.

WAN Link Load Balancing Examples

The following examples are described in this section:

- Example 1: Simple WAN Link Load Balancing, page 639
- Example 2: WAN Link Load Balancing with Server Load Balancing, page 646

Example 1: Simple WAN Link Load Balancing

In this example, many of the load balancing options are left to their default values. See Server Load Balancing, page 165 for details on other options.

Figure 104 - Simple WAN Link Load Balancing Example, page 639 illustrates a simple topology with two WAN links. Two ISPs, a server, and a client are directly connected to Alteon. Alteon load balances traffic between the two WAN links for both inbound and outbound traffic.

The server hosting **www.radware.com** is directly connected to a port on Alteon. To illustrate outbound traffic, a client is directly connected to another port on Alteon.

Figure 104: Simple WAN Link Load Balancing Example

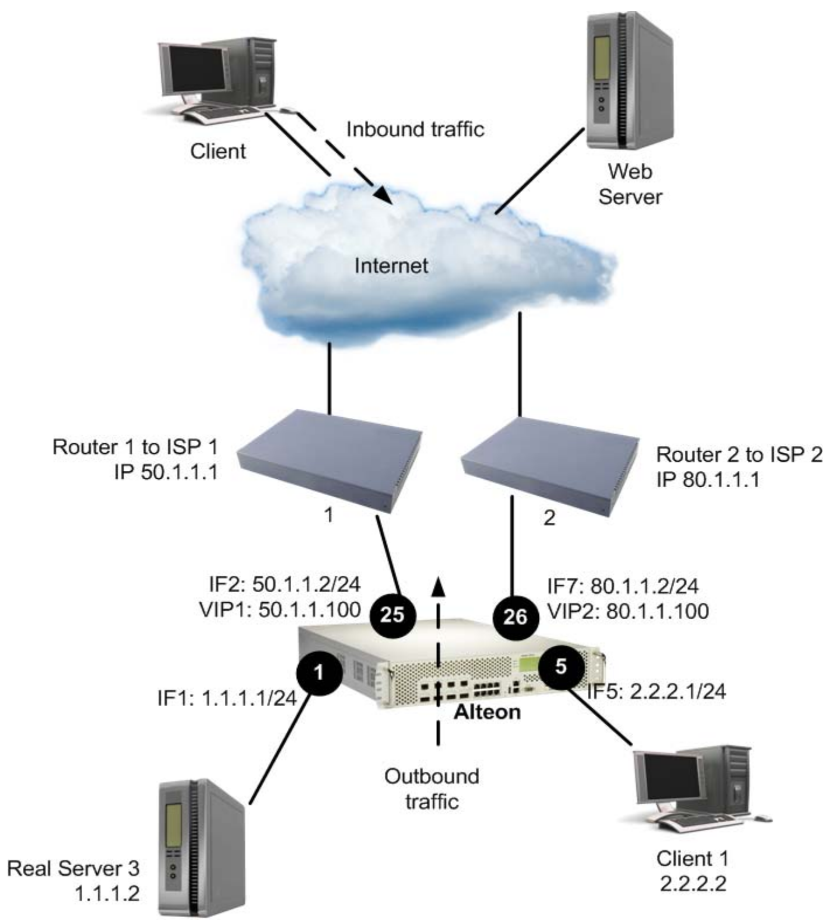


Table 56 - Configuring Simple WAN Load Balancing, page 640 provides an overview of configuring simple WAN load balancing. All definitions for this example refer to Figure 104 - Simple WAN Link Load Balancing Example, page 639.

Table 56: Configuring Simple WAN Load Balancing

Step 1—Configure Basic Parameters

This step includes configuring VLANs, IP interfaces, and defining gateways per VLAN. Gateways per VLAN is recommended if you have not configured other routing protocols. For each ISP, configure a default gateway for each VLAN.

1. Assign an IP address to each of the ISP links. The WAN links in any given real server group must have an IP route to Alteon that performs the load balancing functions. For this example, the two ISP links are the following IP addresses on different IP subnets:

Table 57: ISP links: Real Server IP Addresses

WAN links	IP address
ISP 1	50.1.1.1
ISP 2	80.1.1.1

2. Configure VLANs. The real server IP addresses (WAN links and real server 3) and the respective IP interfaces must be on different VLANs. The **pvid** command sets the default VLAN number which is used to forward frames which are not VLAN tagged. The default number is 1.

```
>> # /cfg/port 25/pvid          (Sets the default VLAN number)
>> # /cfg/port 26/pvid 7        (Sets the default VLAN number)
>> # /cfg/port 1/pvid 1         (Sets the default VLAN number)
>> # /cfg/port 5/pvid 5         (Sets the default VLAN number)
>> # /cfg/vlan 2/ena            (Enable VLAN 2)
>> # /cfg/vlan 2/def 25         (Add port 25 to VLAN 2)
>> # /cfg/vlan 7/ena            (Enable VLAN 7)
>> # /cfg/vlan 7/def 26         (Add port 26 to VLAN 7)
>> # /cfg/vlan 1/ena            (Enable VLAN 2)
>> # /cfg/vlan 1/def 1          (Add port 1 to VLAN 1)
>> # /cfg/vlan 5/ena            (Enable VLAN 5)
>> # /cfg/vlan 5/def 5          (Add port 5 to VLAN 5)
>> # /cfg/l2/stg 1/off          (Disable STG)
```

`>> # /cfg/port 25/pvid`	(Clear STG)
`>> # /cfg/l2/stg 1/port 1257`	(Add ports 1, 2, 5, and 7 to STG 1)

3. Configure the IP interfaces on Alteon. Alteon must have an IP route to all of the real servers that receive switching services. For load balancing the traffic, Alteon uses this path to determine the level of TCP/IP reach of the WAN links.

`>> Main # /cfg/l3/if 2`	(Define interface 2 for ISP 1)
`>> IP Interface 2 # ena`	(Enable interface 2)
`>> IP Interface 2# addr 50.1.1.2`	(Define the IP address for interface 2)
`>> IP Interface 2# mask 255.255.255.0`	(Define the mask for interface 2)
`>> IP Interface 2# broad 50.1.1.255`	(Define the broadcast for interface 2)
`>> IP Interface 2 # vlan 2`	(Specify the VLAN for interface 2)
`>> Main # /cfg/l3/if 7`	(Define interface 7 for ISP 2)
`>> IP Interface 7# ena`	(Enable interface 7)
`>> IP Interface 7# addr 80.1.1.2`	(Define the IP address for interface 7)
`>> IP Interface 7# mask 255.255.255.0`	(Define the mask for interface 7)
`>> IP Interface 7# broad 80.1.1.255`	(Define the broadcast for interface 7)
`>> IP Interface 7# vlan 7`	(Specify the VLAN for interface 7)
`>> Main # /cfg/l3/if 1`	(Define interface 1 for Real server 3)
`>> IP Interface 1# ena`	(Enable interface 1)
`>> IP Interface 1# addr 1.1.1.1`	(Define the IP address for interface 1)
`>> IP Interface 1# mask 255.255.255.0`	(Define the mask for interface 1)
`>> IP Interface 1# broad 1.1.1.255`	(Define the broadcast for interface 1)
`>> IP Interface 1# vlan 1`	(Specify the VLAN for interface 1)
`>> Main # /cfg/l3/if 5`	(Define interface 5 for internal client)
`>> IP Interface 5# ena`	(Enable interface 5)
`>> IP Interface 5# addr 2.2.2.1`	(Define the IP address for interface 5)
`>> IP Interface 5# mask 255.255.255.0`	(Define the mask for interface 5)
`>> IP Interface 5# broad 2.2.2.255`	(Define the broadcast for interface 5)
`>> IP Interface 5# vlan 5`	(Specify the VLAN for interface 5)

Step 2—Configure the Load Balancing Parameters for ISP Routers

Configure the ISP routers with load balancing parameters: real servers, group, metric, and health.

1. Configure IP addresses for WAN link routers.

   Proxy is disabled on the real servers, so that the original destination IP address is preserved.

`>> # /cfg/slb/real 1/rip 50.1.1.1`	(Define IP address for WAN link 1)
`>> Real server 1# ena`	(Enable real server 1)
`>> Real server 1# proxy dis`	(Disable proxy)
`>> # /cfg/slb/real 2/rip 80.1.1.1`	(Define IP address for WAN link 2)
`>> Real server 2# ena`	(Enable real server 2)

`>> Real server 2 # adv`	(Select the advance menu)
`>> Real server 2 Advanced# proxy dis`	(Disable proxy)

2. Create a group to load balance the WAN link routers.

`>> # /cfg/slb/group 100`	(Define a group)
`>> Real Server Group 100# add 1`	(Add real server 1)
`>> Real Server Group 100# add 2`	(Add real server 2)

3. Assign the response metric for the WAN link group.

```
>> Real Server Group 100# metric response
```

   Any of the server load balancing metrics may be used, but response or bandwidth metric is recommended.

4. Configure health check for the WAN link group.

```
>> Real Server Group 100# health icmp
```

5. Enable SLB.

```
>> # /cfg/slb/on
```

Step 3a (Outbound Traffic)—Configure the WAN Link Ports

Configure proxy IP addresses on ports 25 and 26 for WAN link load balancing.

`>> # /cfg/slb/pip/type port`	(Set base type of proxy IP address)
`>> # /cfg/slb/pip`	
`>> Proxy IP Address# add 50.1.1.3 25`	(Set proxy IP address for port 25)
`>> Proxy IP Address# add 80.1.1.7 26`	(Set proxy IP address for port 26)

Each proxy IP address must be unique on your network.

Step 3b (Inbound Traffic)—Configure the WAN Link Ports

1. Enable client processing for ports 25 and 26. This enables inbound traffic to access the virtual server IP address.

```
>> # /cfg/slb/port 25/client ena
>> # /cfg/slb/port 26/client ena
```

2. Enable transparent load balancing for ports 25 and 26. Enable transparent load balancing to ensure the returning traffic from all servers to go back to the same ISP router.

```
>> # /cfg/slb/port 25/rts ena
>> # /cfg/slb/port 26/rts ena
```

3. Enable WAN link load balancing.

`>> # /cfg/slb/linklb`	(Select the link load balancing menu)
`>> # /cfg/slb/linklb/group 100`	(Specify the ISP group of real servers)
`>> # /cfg/slb/linklb/ena`	(Enable link load balancing)

4. Enable Direct Access Mode (DAM). Typically, you have two or more virtual server IP addresses representing the same real service. On the return path, DAM ensures that the real server IP address is mapped to the correct virtual IP address.

```
>> # /cfg/slb/adv/direct ena
```

For information about DAM, refer to Direct Access Mode, page 200.

Step 4a (Outbound Traffic)—Configure the Client Ports

Configure the redirection filter and enable the filter for link load balancing. This is required to translate (NAT) the client IP address to the proxy IP address.

1. Define the WAN link load balancing redirection filter.

```
>> # /cfg/slb/filt 100
>> Filter 100# ena
>> Filter 100# action redir
>> Filter 100# group 100                    (Select the ISP group of real servers)
```

2. Enable WAN link load balancing for the redirection filter.

```
>> Filter 100# adv/redir
>> Filter 100 Redirection Advanced# linklb ena
```

3. Add the link load balancing filter 100 to the outbound client port.

`>> # /cfg/slb/port 5`	(Select port 5)
`>> SLB Port 5# add 100`	(Add filter 100 to port 5)
`>> SLB Port 5# filt ena`	(Enable the filter)

4. If you are configuring link load balancing for outbound traffic only, then go to Step 7—Apply and Save Your Changes, page 645. The remaining steps in this procedure are used for load balancing of inbound traffic only.

Step 4b (Inbound Traffic)—Configure Server Ports

For each real server connected to Alteon, assign a real server number, specify its IP address, and enable the real server. Define a real server group and add the real server to the group.

1. Configure real server and create a group.

`>> # /cfg/slb/real3/rip 1.1.1.2`	(Define IP address for real server 3)
`>> Real server 3# ena`	(Enable real server 3)
`>> # /cfg/slb/group 3`	(Define a group)
`>> Real server Group 3# add 3`	(Add real server 3)

2. Enable server processing.

```
>> # /cfg/slb/port 1/server ena
```

3. Enable filtering on server port 1.

   Filtering is enabled on port 1, because you want Alteon to look up the session table for the transparent load balancing entry.

`>> # /cfg/slb/port 1`	(Select port 1)
`>> SLB Port 1# filt ena`	(Enable the filter)

Step 5—Configure the Virtual Server IP Address and the Services for Each ISP

All client requests are addressed to a virtual server IP address defined on Alteon. Clients acquire the virtual server IP address through normal DNS resolution. In this example, HTTP and FTP are configured as the services running on this virtual server, and this service is associated with the real server group.

Other TCP/IP services can be configured in a similar fashion. For a list of other well-known services and ports, see Table 20 - Well-Known Application Ports, page 175. To configure multiple services, see Multiple Services per Real Server, page 177.

Define a virtual server IP address for each ISP.

Step 5a—Configure the Virtual Server IP Address and the Services for ISP 1

Define a virtual server and add the services and real server group for ISP 1.

1. Configure a virtual server for ISP 1.

`>> # /cfg/slb/virt 1`	(Select the virtual server)
`>> Virtual 1 Server 1# vip 50.1.1.100`	(Set IP address from the ISP 1 subnet)
`>> Virtual 1 Server 1# ena`	(Enable virtual server)

2. Add HTTP and FTP services for the virtual server.

`>> # /cfg/slb/virt 1`	(Select the virtual server)
`>> Virtual 1 Server 1# service 80`	(Add the HTTP service)
`>> Virtual 1 Server 1 HTTP Service# group 3`	(Add real server group)
`>> Virtual 1 Server 1 HTTP Service#..`	(Go to the virtual server menu)
`>> Virtual 1 Server 1# service ftp`	(Add the FTP service)
`>> Virtual 1 Server 1 ftp Service# group 3`	(Add real server group)

Step 5b—Configure the Virtual Server IP Address and the Services for ISP 2

Define a virtual server and add the services and real server group for ISP 2.

1. Configure a virtual server for ISP 2.

`>> # /cfg/slb/virt 2`	(Select the virtual server)
`>> Virtual Server 2# vip 80.1.1.100`	(Set IP address from the ISP 1 subnet)
`>> Virtual Server 2# ena`	(Enable virtual server)

2. Add HTTP and FTP services for the virtual server.

`>> # /cfg/slb/virt 2`	(Select the virtual server)
`>> Virtual Server 2# service 80`	(Add the HTTP service)
`>> Virtual Server 2 HTTP Service# ena`	(Enable the service)
`>> Virtual Server 2 HTTP Service# group 3`	(Add real server group)
`>> Virtual Server 2 HTTP Service#..`	(Go to the virtual server menu)
`>> Virtual Server 2# service ftp`	(Add the FTP service)
`>> Virtual Server 2 ftp Service# ena`	(Enable the service)
`>> Virtual Server 2 ftp Service# group 3`	(Add real server group)

Step 6—Configure Alteon as a Domain Name Server

Define the domain record name and map the virtual server and real server (ISP router) for each WAN link.

1. Configure the domain record to behave as a Domain Name Server.

`>> # /cfg/slb/linklb/drecorcd 1`	(Select the domain record menu)
`>> Domain record 1# domain radware.com`	(Define the domain name)
`>> Domain Record 1# ena`	(Enable the domain)

2. Configure an entry for each ISP and specify the virtual server and real server (ISP router).

   You must map the domain record, **radware.com**, to each ISP. Each ISP has two parameters: a virtual IP address and a real server IP address. The virtual IP address is used to respond to the DNS query for the **radware.com** domain. The real server IP address is used to measure the ISP load and ISP health. These commands map the two parameters to the ISP link.

`>> Domain record 1# entry 1/ena`	(Define entry for ISP 1)
`>> Virt Real Mapping virt 1`	(Select virtual server 1 for ISP 1)
`>> Virt Real Mapping# real 1`	(Select real server for ISP 1)
`>> Domain record 1# entry 2/ena`	(Define entry for ISP 2)
`>> Virt Real Mapping# virt 2`	(Select virtual server 2 for ISP 2)
`>> Virt Real Mapping# real 2`	(Select real server for ISP 2)

Step 7—Apply and Save Your Changes

You must **apply** your changes in order for them to take effect, and you must **save** changes if you want them to remain in effect after reboot.

1. Apply and verify the configuration.

```
>> Layer 4# apply
>> Layer 4# cur
```

   Examine the resulting information. If any settings are incorrect, make the appropriate changes.

2. Save your new configuration changes.

```
>> Layer 4# save
```

3. Check the load balancing information.

```
>> Layer 4# /info/slb/dump
```

4. Check that all load balancing parameters are working as expected. If necessary, make any appropriate configuration changes and then check the information again.

Example 2: WAN Link Load Balancing with Server Load Balancing

In this example, Alteon is configured for standard server load balancing. Alteon is configured to load balance the WAN links for both outbound and inbound traffic and perform server load balancing for inbound traffic.

The configuration is similar to Example 1: Simple WAN Link Load Balancing, page 639, except that the virtual server IP addresses are configured as real server IP addresses and are added to a group.

Figure 105: WAN Link Load Balancing with Server Load Balancing

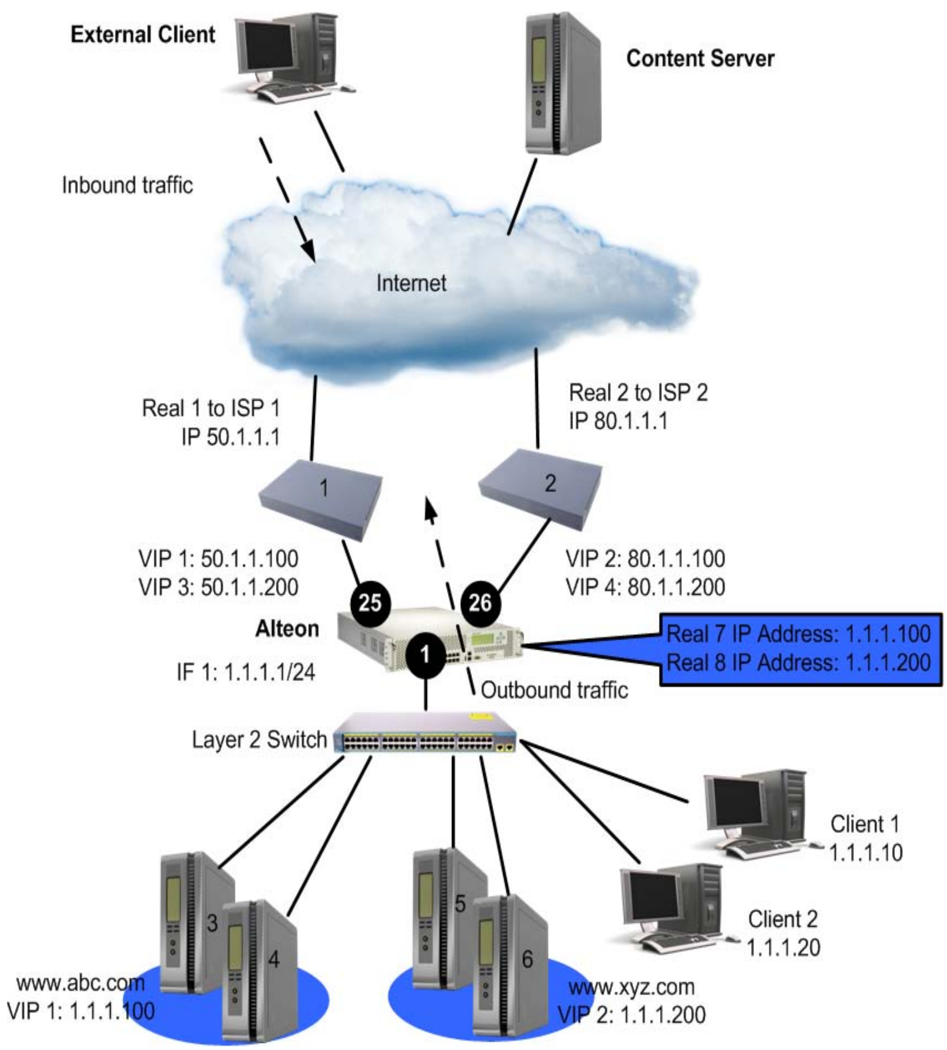


Table 58 - Configuring WAN Link Load Balancing with SLB, page 647 provides an overview of configuring simple WAN load balancing with SLB. All definitions for this example refer to Figure 105 - WAN Link Load Balancing with Server Load Balancing, page 646.

Table 58: Configuring WAN Link Load Balancing with SLB

For outbound traffic	For inbound traffic
Step 1—Configure Basic Parameters, page 647	
Step 2—Configure the Load Balancing Parameters for ISP Routers, page 648	
Step 3a (Outbound Traffic)—Configure the WAN Link Ports, page 649	Step 3b (Inbound Traffic)—Configure the WAN Link Ports, page 649
Step 4a (Outbound Traffic)—Configure the Internal Network Port, page 650	Step 4b (Inbound Traffic)—Configure the Internal Network, page 650
	Step 5—Configure the Virtual Server IP Address and the Services for Each ISP, page 652
	Step 6—Configure Alteon as a Domain Name Server, page 653
Step 7—Apply and Save Your Changes, page 654	

Step 1—Configure Basic Parameters

This step includes configuring VLAN, interfaces, and defining gateways per VLAN. Gateways per VLAN is recommended if you have not configured other routing protocols. Configure a default gateway per VLAN for each ISP.

1. Assign an IP address to each of the ISP links. The WAN links in any given real server group must have an IP route to Alteon that performs the load balancing functions. For this example, the two ISP links are the following IP addresses on different IP subnets:

Table 59: ISP links: Real Server IP Addresses

WAN links	IP address
ISP 1	80.1.1.1
ISP 2	30.1.1.1

2. Configure the IP interfaces on Alteon. Alteon must have an IP route to all of the real servers that receive switching services. For load balancing the traffic, Alteon uses this path to determine the level of TCP/IP reach of the WAN links.
3. On Alteon, configure VLANs.

```
>> # /cfg/port 25/pvid 2          (Sets the default VLAN number)
>> # /cfg/port 26/pvid 7          (Sets the default VLAN number)
>> # /cfg/port 1/pvid 1           (Sets the default VLAN number)
>> # /cfg/port 5/pvid 5           (Sets the default VLAN number)
>> # /cfg/vlan 2/ena              (Enable VLAN 2)
>> # /cfg/vlan 2/def 25           (Add port 25 to VLAN 2)
>> # /cfg/vlan 7/ena              (Enable VLAN 7)
>> # /cfg/vlan 7/def 26           (Add port 26 to VLAN 7)
>> # /cfg/vlan 1/ena              (Enable VLAN 1)
```

`>> # /cfg/vlan 1/def 1`	(Add port 1 to VLAN 1)
`>> # /cfg/vlan 5/ena`	(Enable VLAN 5)
`>> # /cfg/vlan 5/def 5`	(Add port 5 to VLAN 5)
`>> # /cfg/stp 1/off`	(Disable STP)
`>> # /cfg/stp 1/clear`	(Clear STP)
`>> # /cfg/stp 1/add 1 25 26 5`	(Add ports 1, 25, 26, 5 to STP 1)

4. Configure the IP interfaces on Alteon.

`>> # /cfg/if 1`	(Define interface 1)
`>> IP Interface 1# ena`	(Enable interface 1)
`>> IP Interface 1# addr 1.1.1.1`	(Define the IP address for interface 1)
`>> IP Interface 1# mask 255.255.255.0`	(Define the mask for interface 1)
`>> IP Interface 1# broad 1.1.1.255`	(Define the broadcast for interface 1)
`>> IP Interface 1# vlan 1`	(Specify the VLAN for interface 1)
`>> # /cfg/if 2`	(Define interface 2)
`>> IP Interface 2# ena`	(Enable interface 2)
`>> IP Interface 2# addr 50.1.1.2`	(Define the IP address for interface 2)
`>> IP Interface 2# mask 255.255.255.0`	(Define the mask for interface 2)
`>> IP Interface 2# broad 50.1.1.255`	(Define the broadcast for interface 2)
`>> IP Interface 2# vlan 2`	(Specify the VLAN for interface 2)
`>> # /cfg/if 7`	(Define interface 7)
`>> IP Interface 7# ena`	(Enable interface 7)
`>> IP Interface 7# addr 80.1.1.2`	(Define the IP address for interface 7)
`>> IP Interface 7# mask 255.255.255.0`	(Define the mask for interface 7)
`>> IP Interface 7# broad 80.1.1.255`	(Define the broadcast for interface 7)
`>> IP Interface 7# vlan 7`	(Specify the VLAN for interface 7)

Step 2—Configure the Load Balancing Parameters for ISP Routers

On Alteon, configure the ISP routers as if they were real servers, with SLB parameters: real servers, group, metric, and health.

1. Configure IP addresses for WAN link routers.

`>> # /cfg/slb/real 1/rip 50.1.1.1`	(Define IP address for WAN link 1)
`>> Real server 1# ena`	(Enable real server 1)
`>> Real server 1# proxy dis`	(Disable proxy)
`>> # /cfg/slb/real 2/rip 80.1.1.1`	(Define IP address for WAN link 2)
`>> Real server 2# ena`	(Enable real server 2)
`>> Real server 2 # adv`	(Select the advance menu)
`>> Real server 2 Advanced# proxy dis`	(Disable proxy)

Proxy is disabled on the real servers, because link load balancing and full NAT cache redirection cannot coexist.

2. Create a group to load balance the WAN link routers.

`>> # /cfg/slb/group 100`	(Define a group)
`>> Real Server Group 100# add 1`	(Add real server 1)
`>> Real Server Group 100# add 2`	(Add real server 2)

3. Assign the response metric for the WAN link group.

```
>> Real Server Group 100# metric response
```

Any of the server load balancing metrics may be used, but response or bandwidth metric is recommended.

4. Configure health check for the WAN link group.

```
>> Real Server Group 100# health icmp
```

5. Enable SLB.

```
>> # /cfg/slb/on
```

Step 3a (Outbound Traffic)—Configure the WAN Link Ports

Configure proxy IP addresses on ports 25 and 26.

`>> # /cfg/slb/pip/type port`	(Set base type of proxy IP address)
`>> # /cfg/slb/pip`	
`>> Proxy IP Address# add 50.1.1.2 25`	(Set proxy IP address for port 25)
`>> Proxy IP Address# add 80.1.1.7 26`	(Set proxy IP address for port 26)

Each proxy IP address must be unique on your network.

Step 3b (Inbound Traffic)—Configure the WAN Link Ports

1. Enable client processing at ports 25 and 26.

```
>> # /cfg/slb/port 25/client ena
>> # /cfg/slb/port 26/client ena
```

This enables inbound traffic to access the virtual server IP address.

2. Enable transparent load balancing for ports 25 and 26. Enable transparent load balancing to ensure the returning traffic from all servers to go back to the same ISP router.

```
>> # /cfg/slb/port 25/rts ena
>> # /cfg/slb/port 26/rts ena
```

3. Enable WAN link load balancing.

`>> # /cfg/slb/linklb`	(Select the link load balancing menu)
`>> # /cfg/slb/linklb/group 100`	(Specify the ISP group of real servers)
`>> # /cfg/slb/linklb/ena`	(Enable link load balancing)

4. Enable Direct Access Mode (DAM). Typically, you have two or more virtual server IP addresses representing the same real service. On the return path, DAM ensures that the real server IP address is mapped to the correct virtual IP address.

```
>> # /cfg/slb/adv/direct ena
```

For information about DAM, refer to Direct Access Mode, page 200.

Step 4a (Outbound Traffic)—Configure the Internal Network Port

Configure the redirection filter and enable the filter for link load balancing. This is required to translate (NAT) the client IP address to the proxy IP address.

1. Define the WAN link load balancing redirection filter.

```
>> # /cfg/slb/filt 100
.. Filter 100# ena
>> action redir
>> Filter 100# group                          (Select the ISP group of real servers)
```

2. Enable WAN link load balancing for the redirection filter.

```
>> Filter 100# adv
>> Filter 100# /c/slb/filt 100/adv/redir/linklb ena
```

3. Add the link load balancing filter 100 to the outbound client port.

```
>> # /cfg/slb/port 1                          (Select port 1)
>> SLB Port 1# add 100                        (Add filter 100 to port 1)
>> SLB Port 1# filt ena                       (Enable the filter)
```

4. If you are configuring link load balancing for outbound traffic only, then go to Step 7—Apply and Save Your Changes, page 645. The remaining steps in this procedure are for load balancing inbound traffic only.

Step 4b (Inbound Traffic)—Configure the Internal Network

Configure the virtual server IP addresses on Alteon as real server IP addresses. In this example, you will configure two real server IP addresses for each of the two virtual server IP addresses. Then, define a real server group and add the real servers to the group.

1. Configure real server and create a group.

   The real server IP address must be the virtual server IP address of the SLB servers that are hosting **abc.com**.

```
>> # /cfg/slb/real 7/rip 1.1.1.100            (Define IP address for www.abc.com)
>> Real server 7# ena                         (Enable real server 3)
>> # /cfg/slb/group 3                         (Define a group)
>> Real server Group 3# add 3                 (Add real server 7)
```

2. Configure real server and create a group.

   The real server IP address must be the virtual server IP address of the SLB servers that are hosting **xyz.com**.

`>> # /cfg/slb/real 8/rip 1.1.1.200`	(Define IP address for xyz.com)
`>> Real server 8# ena`	(Enable real server 8)
`>> # /cfg/slb/group 4`	(Define a group)
`>> Real server Group 4# add 4`	(Add real server 8)

3. Enable filter on server port 1.

   Filter is enabled on port 1, because you want Alteon to look up the session table for the transparent load balancing entry.

`>> # /cfg/slb/port 1`	(Select port 1)
`>> SLB Port 1# filt ena`	(Enable the filter)

4. Enable server processing on port 1.

```
>> # /cfg/slb/port 1/server ena
```

5. Configure an allow filter for health checks to occur.

   If you have enabled link load balancing filter and server processing on the same port, then an allow filter must be configured for health checks. The allow filter is activated before the link load balancing filter, so that the health check traffic does not get redirected to the WAN links.

`>> # /cfg/slb/filt 50`	
`>> Filter 50# sip 1.1.1.0`	(From server subnet)
`>> Filter 50# smask 255.255.255.0`	
`>> Filter 50# dip 1.1.1.1`	(To IF 1 on Alteon)
`>> Filter 50# action allow`	
`>> Filter 50# ena`	

   For more information on health checking, see Health Checks for Real Servers, page 176.

6. Add the allow filter 50 to port 1.

`>> # /cfg/slb/port 1`	(Select port 1)
`>> SLB Port 1# 50`	(Add filter 50 to port 1)
`>> SLB Port 1# filt ena`	(Enable the filter)

Note: If you are using two Alteons for redundancy, then must add allow filters for VRRP before the redirection filter. For more information on VRRP, see High Availability, page 507.

Step 5—Configure the Virtual Server IP Address and the Services for Each ISP

All client requests are addressed to a virtual server IP address on a virtual server defined on Alteon. Clients acquire the virtual server IP address through normal DNS resolution. In this example, HTTP and FTP are configured as the services running on this virtual server, and this service is associated with the real server group.

Other TCP/IP services can be configured in a similar fashion. For a list of other well-known services and ports, see Table 20 - Well-Known Application Ports, page 175. To configure multiple services, see Configuring Multiple Service Ports, page 197.

Note: Define a virtual server IP address for each ISP.

Step 5a—Configure the Virtual Server IP Address and the Services for ISP 1

Define a virtual server and add the services and real server group for ISP 1.

1. Configure a virtual server for ISP 1.

`>> # /cfg/slb/virt 1`	(Select the virtual server)
`>> Virtual Server 1# vip 50.1.1.100`	(Set IP address from the ISP 1 subnet)
`>> Virtual Server 1# ena`	(Enable virtual server)

2. Add HTTP and FTP services for the virtual server.

`>> # /cfg/slb/virt 1`	(Select the virtual server)
`>> Virtual Server 1# service 80`	(Add the HTTP service)
`>> Virtual Server 1 HTTP Service# ena`	(Enable the service)
`>> Virtual Server 1 HTTP Service# group 3`	(Add real server group)
`>> Virtual Server 1 HTTP Service#...`	(Go to the virtual server menu)
`>> Virtual Server 1# service ftp`	(Add the FTP service)
`>> Virtual Server 1 ftp Service# ena`	(Enable the service)
`>> Virtual Server 1 ftp Service# group 3`	(Add real server group)

Step 5b—Configure the VIrtual Server IP Address and the Services for ISP 2

Define a virtual server and add the services and real server group for ISP 2.

1. Configure a virtual server for ISP 2.

`>> # /cfg/slb/virt 2`	(Select the virtual server)
`>> Virtual 1 Server 2# vip 80.1.1.1`	(Set IP address from the ISP 1 subnet)
`>> Virtual Server 1 Server 2# ena`	(Enable virtual server)

2. Add HTTP and FTP services for the virtual server.

`>> # /cfg/slb/virt 2`	(Select the virtual server)
`>> Virtual 1 Server 2# service 80`	(Add the HTTP service)
`>> Virtual 1 Server 2 HTTP Service# group 3`	(Add real server group)
`>> Virtual 1 Server 2 HTTP Service#...`	(Go to the virtual server menu)
`>> Virtual 1 Server 2# service ftp`	(Add the FTP service)
`>> Virtual 1 Server 2 ftp Service# group 3`	(Add real server group)

Note: Repeat Step 5a—Configure the Virtual Server IP Address and the Services for ISP 1, page 652 and Step 5b—Configure the VIrtual Server IP Address and the Services for ISP 2, page 652 for virtual server 3 and 4, and add group 4 for each of the services. This allows inbound traffic to access SLB servers hosting the XYZ.com.

Step 6—Configure Alteon as a Domain Name Server

This steps involves configuring the domain record name and mapping the virtual server and real server (ISP router) for each WAN link.

```
drecord 1: abc.com
       entry 1 : VIP 1 and Real 1 (for ISP 1)
       entry 2 : VIP 2 and Real 2 (for ISP 2)

drecord 2: xyz.com
       entry 1 : VIP 3 and Real 1 (for ISP 1)
       entry 2 : VIP 4 and Real 2 (for ISP 2)
```

You must map the domain record, **radware.com** to each ISP. Each ISP has two parameters: a virtual IP address and a real server IP address. The virtual IP address is used to respond to the DNS query for the **radware.com** domain. The real server IP address is used to measure the ISP load and ISP health. These commands map the two parameters to the ISP link.

1. Configure the domain record for **abc.com**.

`>> # /cfg/slb/linklb/drecord 1`	(Select the domain record menu)
`>> Domain Record 1# ena`	(Enable the domain)
`>> Domain record 1# domain abc.com`	(Define the domain name)

2. Configure an entry for each ISP and specify the virtual and real server (ISP router).

`>> Domain record 1# entry 1/ena`	(Define entry for ISP 1)
`>> Virt Real Mapping virt 1`	(Select virtual server 1 for ISP 1)
`>> Virt Real Mapping# real 1`	(Select real server for ISP 1)
`>> Domain record 1# entry 2/ena`	(Define entry for ISP 2)
`>> Virt Real Mapping# virt 2`	(Select virtual server 2 for ISP 2)
`>> Virt Real Mapping# real 2`	(Select real server for ISP 2)

3. Configure the domain record for **xyz.com**.

`>> # /cfg/slb/linklb/drecord 2`	(Select the domain record menu)
`>> Domain Record 2# ena`	(Enable the domain)
`>> Domain record 2# domain xyz.com`	(Define the domain name)

4. Configure an entry for each ISP and specify the virtual and real server (ISP router).

`>> Domain record 2# entry 1/ena`	(Define entry for ISP 1)
`>> Virt Real Mapping# virt 3`	(Select virtual server 3 for ISP 1)
`>> Virt Real Mapping# real 1`	(Select real server for ISP 1)
`>> Domain record 1# entry 2/ena`	(Define entry for ISP 2)
`>> Virt Real Mapping# virt 4`	(Select virtual server 4 for ISP 2)
`>> Virt Real Mapping# real 1`	(Select real server for ISP 2)

Step 7—Apply and Save Your Changes

You must **apply** your changes in order for them to take effect, and you must **save** changes if you want them to remain in effect after reboot.

1. Apply and verify the configuration.

```
>> Layer 4# apply
>> Layer 4# cur
```

   Examine the resulting information. If any settings are incorrect, make the appropriate changes.

2. Save your new configuration changes.

```
>> Layer 4# save
```

3. Check the load balancing information.

```
>> Layer 4# /info/slb/dump
```

4. Check that all load balancing parameters are working as expected. If necessary, make any appropriate configuration changes and then check the information again.

Health Checking and Multi-homing

When using health checking with WAN link load balancing, sometimes disruption of service on one link may not be immediately apparent. This is because of how health checking interacts with a load balanced WAN environment.

Consider an Alteon that is multi-homed to two service providers. Alteon has WAN link load balancing configured for incoming and outgoing traffic. If the link to the first service provider is removed, the health check for this link does not fail even though the corresponding interface is down. This is because the health check packet is still being sent and received through the connection to the second service provider. This is a by-product of the tendency of any routing protocol to re-route a packet to an active link.

To overcome this problem, two filters can be used to on the two load-balanced ports to suppress the ICMP echo reply which makes the health check fail if the link fails.

Example

This example applies filter 10 to the link to the first service provider:

```
>> /c/slb/filt 10
           ena
           action deny
           sip 80.1.1.1
           smask 255.255.255.255
           dip 50.1.1.2
           dmask 255.255.255.255
           proto icmp
           vlan any
/c/slb/filt 10/adv
           icmp echorep
```

After the filter is applied to the first link, the filter on the second link is applied. The following commands would apply filter 20 to the link to the second service provider:

```
/c/slb/filt 20
        ena
        action deny
        sip 50.1.1.1
        smask 255.255.255.255
        dip 80.1.1.2
        dmask 255.255.255.255
        proto icmp
        vlan any
/c/slb/filt 20/adv
        icmp echorep
```

Note: In addition to the application of the filters, Radware also recommends using a static route.

Chapter 23 – Firewall Load Balancing

Firewall Load Balancing (FWLB) with Alteon allows multiple active firewalls to operate in parallel. Parallel operation enables users to maximize firewall productivity, scale firewall performance without forklift upgrades, and eliminate the firewall as a single point-of-failure.

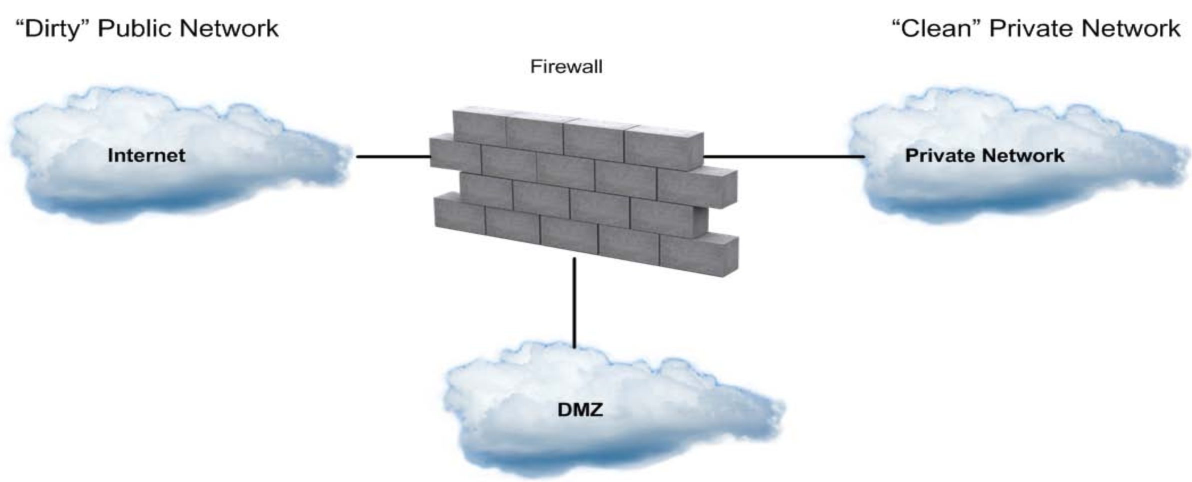
This chapter discusses the following topics:

- Firewall Overview, page 657—An overview of firewalls and the various FWLB solutions supported by Alteon.
- Basic FWLB, page 658—Explanation and example configuration for FWLB in simple networks, using two parallel firewalls and two Alteons. The basic FWLB method combines redirection filters and static routing for FWLB.
- Four-Subnet FWLB, page 668—Explanation and example configuration for FWLB in a large-scale, high-availability network with redundant firewalls and Alteons. This method combines redirection filters, static routing, and Virtual Router Redundancy Protocol (VRRP).
- Advanced FWLB Concepts, page 683
  - — Free-Metric FWLB, page 683—Using other load-balancing metrics (besides hash) by enabling the transparent load balancing (rtsrcmac) option.
  - — Adding a Demilitarized Zone (DMZ), page 686—Adding a DMZ for servers that attach to Alteon between the Internet and the firewalls.
  - — Firewall Health Checks, page 687—Methods for fine-tuning the health checks performed for FWLB.

Firewall Overview

Firewall devices have become indispensable for protecting network resources from unauthorized access. Without FWLB, firewalls can become critical bottlenecks or single points-of-failure for your network. As an example, consider the network in Figure 106 - Firewall Configuration with FWLB, page 657:

Figure 106: Firewall Configuration with FWLB



One network interface card on the firewall is connected to the public side of the network, often to an Internet router. This is known as the *dirty,* or untrusted, side of the firewall. Another network interface card on the firewall is connected to the side of the network with the resources that must be protected. This is known as the *clean,* or trusted, side of the firewall.

In the example in Figure 106 - Firewall Configuration with FWLB, page 657, all traffic passing between the dirty, clean, and *demilitarized zone* (DMZ) networks must traverse the firewall, which examines each individual packet. The firewall is configured with a detailed set of rules that determine which types of traffic are allowed and which types are denied. Heavy traffic can turn the firewall into a serious bottleneck. The firewall is also a single point-of-failure device. If it goes out of service, external clients can no longer reach your services and internal clients can no longer reach the Internet.

Sometimes a DMZ is attached to the firewall or between the Internet and the firewall. Typically, a DMZ contains its own servers that provide dirty-side clients with access to services, making it unnecessary for dirty-side traffic to use clean-side resources.

FWLB provides a variety of options that enhance firewall performance and resolve typical firewall problems. Alteon supports the following FWLB methods:

- **Basic FWLB for simple networks**—This method uses a combination of static routes and redirection filters and is usually employed in smaller networks.

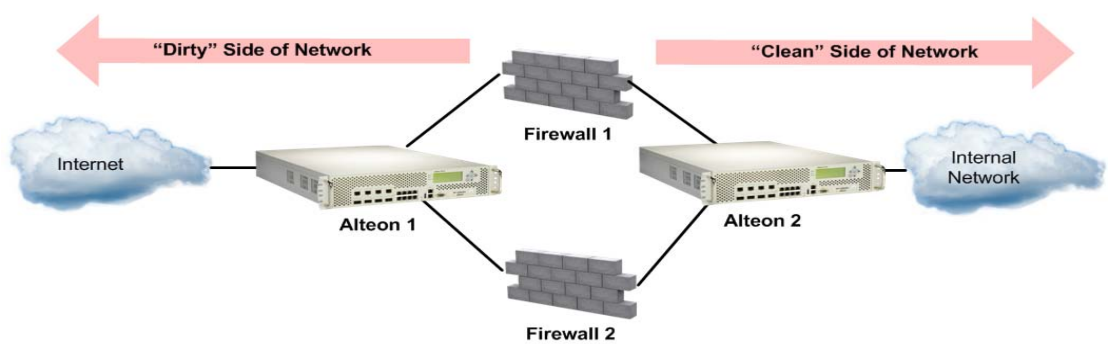
  An Alteon filter on the dirty-side splits incoming traffic into streams headed for different firewalls. To ensure persistence of session traffic through the same firewall, distribution is based on a mathematical *hash* of the IP source and destination addresses. For more information, see Basic FWLB, page 658.
- **Four-Subnet FWLB for larger networks**—Although similar to basic FWLB, the four-subnet method is more often deployed in larger networks that require high-availability solutions. This method adds Virtual Router Redundancy Protocol (VRRP) to the configuration.

  Just as with the basic method, four-subnet FWLB uses the hash metric to distribute firewall traffic and maintain persistence. For more information, see Four-Subnet FWLB, page 668.

Basic FWLB

The basic FWLB method uses a combination of static routes and redirection filters to allow multiple active firewalls to operate in parallel. Figure 107 - Basic FWLB Topology, page 658 illustrates a basic FWLB topology:

Figure 107: Basic FWLB Topology



The firewalls being load balanced are in the middle of the network, separating the dirty side from the clean side. This configuration requires a minimum of two Alteons: one on the dirty side of the firewalls and one on the clean side.

A redirection filter on the dirty-side Alteon splits incoming client traffic into multiple streams. Each stream is routed through a different firewall. The same process is used for outbound server responses. A redirection filter on the clean-side Alteon splits the traffic, and static routes forward each stream through a different firewall and then back to the client.

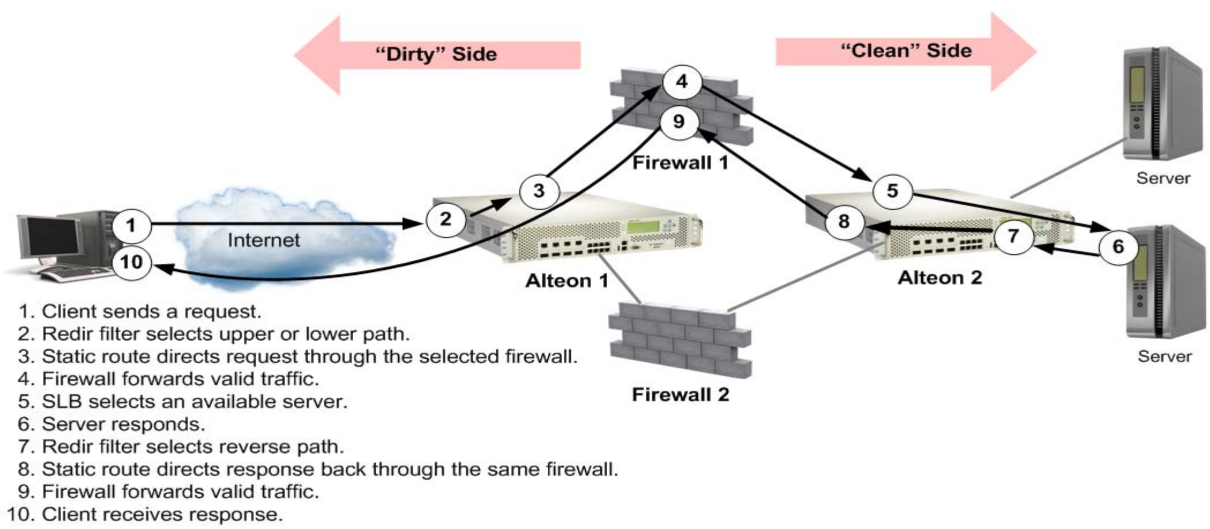
Although other metrics can be used in some configurations (see Free-Metric FWLB, page 683), the distribution of traffic within each stream is normally based on a mathematical hash of the source IP address and destination IP addresses. This ensures that each client request and its related responses will use the same firewall (a feature known as *persistence*) and that the traffic is equally distributed. Persistence is required for the firewall as it maintains state and processes traffic in both directions for a connection.

Although basic FWLB techniques can support more firewalls as well as multiple devices on the clean and dirty sides for redundancy, the configuration complexity increases dramatically. The four-subnet FWLB solution is usually preferred in larger scale, high-availability topologies (see Four-Subnet FWLB, page 668).

Basic FWLB Implementation

As shown in Figure 108 - Basic FWLB Process, page 659, traffic is load balanced among the available firewalls:

Figure 108: Basic FWLB Process



1. The client requests data.

   The external clients are configured to connect to services at the publicly advertised IP address assigned to a virtual server on the clean-side Alteon.

2. A redirection filter balances incoming requests among different IP addresses.

   When the client request arrives at the dirty-side Alteon, a filter redirects it to a real server group that consists of a number of different IP addresses. This redirection filter splits the traffic into balanced streams: one for each IP address in the real server group. For FWLB, each IP address in the real server group represents an IP Interface (IF) on a different subnet on the clean-side Alteon.

3. Requests are routed to the firewalls.

On the dirty-side Alteon, one static route is needed for each traffic stream. For instance, the first static route leads to an IP interface on the clean-side Alteon using the first firewall as the next hop. A second static route leads to a second clean-side IP interface using the second firewall as the next hop, and so on. By combining the redirection filter and static routes, traffic is load balanced among all active firewalls.

All traffic between specific IP source/destination address pairs flows through the same firewall, ensuring that sessions established by the firewalls persist for their duration.

Note: More than one stream can be routed though a particular firewall. You can weight the load to favor one firewall by increasing the number of static routes that traverse it.

4. The firewalls determine if they should allow the packets and, if so, forward them to a virtual server on the clean-side Alteon.

   Client requests are forwarded or discarded according to rules configured for each firewall.

   Note: Rule sets must be consistent across all firewalls.

5. The clean-side Alteon performs normal SLB functions.

   Packets forwarded from the firewalls are sent to the original destination address, that is, the virtual server on the clean-side Alteon. There, they are load balanced to the real servers using standard SLB configuration.

6. The real server responds to the client request.
7. Redirection filters on the clean-side Alteon balance responses among different IP addresses.

   Redirection filters are needed on all ports on the clean-side Alteon that attach to real servers or internal clients on the clean-side of the network. Filters on these ports redirect the Internet-bound traffic to a real server group that consists of a number of different IP addresses. Each IP address represents an IP interface on a different subnet on the dirty-side Alteon.

8. Outbound traffic is routed to the firewalls.

   Static routes are configured on the clean-side Alteon. One static route is needed for each stream that was configured on the dirty-side Alteon. For instance, the first static route is configured to lead to the first dirty-side IP interface using the first firewall as the next hop. The second static route leads to the second dirty-side IP interface using the second firewall as the next hop, and so on.

   Since Alteon intelligently maintains state information, all traffic between specific IP source or destination addresses flows through the same firewall, maintaining session persistence.

   Note: If Network Address Translation (NAT) software is used on the firewalls, FWLB session persistence requires transparent load balancing to be enabled (see Free-Metric FWLB, page 683).

9. The firewall determines if it should allow the packet and, if so, forwards it to the dirty-side Alteon.

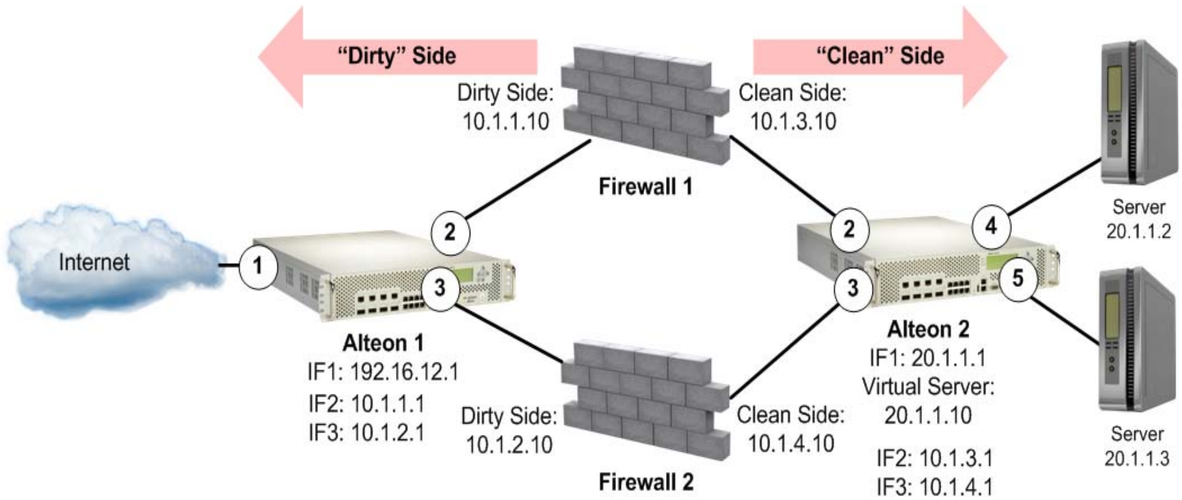
   Each firewall forwards or discards the server responses according to the rules that are configured for it. Forwarded packets are sent to the dirty-side Alteon and out to the Internet.

10. The client receives the server response.

Configuring Basic FWLB

This procedures in the example refer to Figure 109 - Basic FWLB Configuration Example, page 661. While two or four Alteons can be used, this example uses a simple network topology with only two Alteons, one on each side of the firewalls.

Figure 109: Basic FWLB Configuration Example



To configure the dirty-side Alteon

1. Configure VLANs.

   Note: Alternately, if you are using hubs between Alteons and firewalls and you do not want to configure VLANs, you must enable the Spanning Tree Protocol (STP) to prevent broadcast loops.

2. Define the dirty-side IP interface.

   In addition to one IP interface for general Alteon management, there must be one dirty-side IP interface for each firewall path being load balanced. Each must be on a different subnet.

`>> # /cfg/l3/if 1`	(Select IP Interface [IF] 1)
`>> IP Interface 1# addr 192.16.12.1`	(Set address for Alteon management)
`>> IP Interface 1# mask 255.255.255.0`	(Set subnet mask for IF 1)
`>> IP Interface 1# ena`	(Enable IF 1)
`>> IP Interface 1# /cfg/l3/if 2`	(Select IF 2)
`>> IP Interface 2# addr 10.1.1.1`	(Set the IP address for IF 2)
`>> IP Interface 2# mask 255.255.255.0`	(Set subnet mask for IF 2)
`>> IP Interface 2# ena`	(Enable IF 2)
`>> IP Interface 2# /cfg/l3/if 3`	(Select IF 3)

`>> IP Interface 3# addr 10.1.2.1`	(Set the IF 3)
`>> IP Interface 3# mask 255.255.255.0`	(Set subnet mask for IF 3)
`>> IP Interface 3#ena`	(Enable IF3)

3. Configure the clean-side IP interface as if they are real servers on the dirty side.

   Later in this procedure, you will configure one clean-side IP interface on a different subnet for each firewall path being load balanced. On the dirty-side Alteon, create two real servers using the IP address of each clean-side IP interface used for FWLB.

   Note: The real server index number must be the same on both sides of the firewall. For example, if Real Server 1 is the dirty-side IP interface for Firewall 1, then configure Real Server 1 on the clean side with the dirty-side IP interface. Configuring the same real server number on both sides of the firewall ensures that the traffic travels through the same firewall.

`>> IP Interface 3# /cfg/slb/real 1`	(Select Real Server 1)
`>> Real server 1# rip 10.1.3.1`	(Assign clean-side IF 2 address)
`>> Real server 1# ena`	(Enable Real Server 1)
`>> Real server 1# /cfg/slb/real 2`	(Select Real Server 2)
`>> Real server 2# rip 10.1.4.1`	(Assign clean-side IF 3 address)
`>> Real server 2# ena`	(Enable Real Server 2)

   Real servers in the server groups must be ordered the same on both clean side and dirty side Alteon. For example, if the Real Server 1 IF connects to Firewall 1 for the clean side server group, then the Real Server 1 IF on the dirty side should be connected to Firewall 1. Selecting the same real server ensures that the traffic travels through the same firewall.

   Note: Each of the four interfaces used for FWLB (two on each Alteon) in this example must be configured for a different IP subnet.

4. Place the IP interface real servers into a real server group.

`>> IP Interface 2# /cfg/slb/group 1`	(Select Real Server Group 1)
`>> Real server group 1# add 1`	(Add Real Server 1 to Group 1)
`>> Real server group 1# add 2`	(Add Real Server 2 to Group 1)

5. Set the health check type for the real server group to ICMP.

`>> Real server group 1# health icmp`	(Select ICMP as health check type)

6. Set the load-balancing metric for the real server group to **hash**.

```
>> Real server group 1# metric hash
```

Using the hash metric, all traffic between specific IP source/destination address pairs flows through the same firewall. This ensures that sessions established by the firewalls are maintained for their duration.

Note: Other load-balancing metrics such as leastconns, roundrobin, minmiss, response, and bandwidth can be used when enabling the transparent load balancing option. For more information, see Free-Metric FWLB, page 683.

7. Enable SLB.

```
>> Real server group 1# /cfg/slb/on
```

8. Create a filter to allow local subnet traffic on the dirty side of the firewalls to reach the firewall interfaces.

`>> Layer 4# /cfg/slb/filt 10`	(Select Filter 10)
`>> Filter 10# sip any`	(From any source IP address)
`>> Filter 10# dip 192.16.12.0`	(Specify destination IP address)
`>> Filter 10# dmask 255.255.255.0`	(Specify destination mask)
`>> Filter 10# action allow`	(Allow frames with this DIP address)
`>> Filter 10#ena`	(Enable the filter)

9. Create the FWLB redirection filter.

   This filter redirects inbound traffic, load-balancing it among the defined real servers in the group. In this network, the real servers represent IP interfaces on the clean-side Alteon.

`>> Filter 10# /cfg/slb/filt 15`	(Select Filter 15)
`>> Filter 15# sip any`	(From any source IP address)
`>> Filter 15# dip any`	(To any destination IP address)
`>> Filter 15# proto any`	(For any protocol)
`>> Filter 15# action redir`	(Perform redirection)
`>> Filter 15# group 1`	(To Real Server Group 1)
`>> Filter 15# ena`	(Enable this filter)

10. Enable FWLB.

```
>> Filter 15# /adv/redir/fwlb ena
```

11. Add filters to the ingress port.

```
>> SLB Port 5# /cfg/l3/route/ip4
>> IP Static Route# add 10.1.3.1 255.255.255.255 10.1.1.10
>> IP Static Route# add 10.1.4.1 255.255.255.255 10.1.2.10
```

Note: When adding an IPv4 static route, if you are using FWLB and you define two IP interfaces on the same subnet, where one IP interface has a subnet of the host which is also included in the subnet of the second interface, you must specify the interface.

12. Define static routes to the clean-side IP interfaces, using the firewalls as gateways.

    One static route is required for each firewall path being load-balanced. In this case, two paths are required: one that leads to clean-side IF 2 (10.1.3.1) through the first firewall (10.1.1.10) as its gateway, and one that leads to clean-side IF 3 (10.1.4.1) through the second firewall (10.1.2.10) as its gateway.

13. Apply and save the configuration changes

```
>> # apply
>> # save
```

To configure the clean-side Alteon

1. Define the clean-side IP interfaces. Create one clean-side IP interface on a different subnet for each firewall being load balanced.

Note: An extra IP interface (IF 1) prevents server-to-server traffic from being redirected.

```
>> # /cfg/l3/if 1                            (Select IP Interface 1)
>> IP Interface 1# addr 20.1.1.1             (Set IP address for Interface 1)
>> IP Interface 1# mask 255.255.255.0        (Set subnet mask for Interface 1)
>> IP Interface 1# ena                       (Enable IP Interface 1)
>> IP Interface 1# /cfg/l3/if 2              (Select IP Interface 2)
>> IP Interface 2# addr 10.1.3.1             (Set the IP address for Interface 2)
>> IP Interface 2# mask 255.255.255.0        (Set subnet mask for Interface 2)
>> IP Interface 2# ena                       (Enable IP Interface 2)
>> IP Interface 2# /cfg/l3/if 3              (Select IP Interface 3)
>> IP Interface 3# addr 10.1.4.1             (Set the IP address for Interface 3)
>> IP Interface 3# mask 255.255.255.0        (Set subnet mask for Interface 3)
>> IP Interface 3# ena                       (Enable IP Interface 3)
```

2. Configure the dirty-side IP interfaces as if they were real servers on the clean side.

You should already have configured a dirty-side IP interface on a different subnet for each firewall path being load balanced. Create two real servers on the clean-side Alteon using the IP address of each dirty-side IP interface.

Note: The real server index number must be the same on both sides of the firewall. For example, if Real Server 1 is the dirty-side IP interface for Firewall 1, then configure Real Server 1 on the clean side with the dirty-side IP interface. Configuring the same real server number on both sides of the firewall ensures that the traffic travels through the same firewall.

`>> IP Interface 3# /cfg/slb/real 1`	(Select Real Server 1)
`>> Real server 1# rip 10.1.1.1`	(Assign dirty-side IF 1 address)
`>> Real server 1# ena`	(Enable Real Server 1)
`>> Real server 1# /cfg/slb/real 2`	(Select Real Server 2)
`>> Real server 2# rip 10.1.2.1`	(Assign dirty-side IF 2 address)
`>> Real server 2# ena`	(Enable Real Server 2)

Note: Each of the four IP interfaces (two on each Alteon) in this example must be configured for a different IP subnet.

3. Place the real servers into a real server group.

`>> IP Interface 2# /cfg/slb/group 1`	(Select Real Server Group 1)
`>> Real server group 1# add 1`	(Add Real Server 1 to Group 1)
`>> Real server group 1# add 2`	(Add Real Server 2 to Group 1)

4. Set the health check type for the real server group to ICMP.

```
>> Real server group 1# health icmp
```

5. Set the load-balancing metric for the real server group to **hash**.

```
>> Real server group 1# metric hash
```

Note: The clean-side Alteon must use the same metric as defined on the dirty side.

6. Enable SLB.

```
>> Real server group 1# /cfg/slb/on
```

7. Configure ports 2 and 3, which are connected to the clean-side of the firewalls, for client processing.

`>> Real server group 1# /cfg/slb/port 2/client ena`	(Enable client processing on Port 2)
`>> SLB port 2# /cfg/slb/port`	(Enable client processing on Port 3)
`>> SLB port 3# apply`	(Apply the configuration)
`>> SLB port 3# save`	(Save the configuration)

8. Configure the virtual server that will load balance the real servers.

`>> SLB port 3# /cfg/slb/virt 100`	(Configure Virtual Server 100)
`>> Virtual Server 100# vip 20.1.1.10`	(Assign Virtual Server 100 an IP address)
`>> Virtual Server 100# ena`	(Enable the virtual server)

9. Configure the real servers to which traffic will be load balanced. These are the real servers on the network.

`>> Real server group 1# /cfg/slb/real 3`	(Select Real Server 3)
`>> Real server 2 # rip 20.1.12`	(Assign Real Server 2 an IP address)
`>> Real server 2 # ena`	(Enable Real Server 2)
`>> Real server 2 # /cfg/slb/real 4`	(Select Real Server 4)
`>> Real server 3# ena 20.1.1.3`	(Assign Real Server 3 an IP address)

10. Place the real servers into a real server group.

`>> Real server group 3# /cfg/slb/group 200`	(Select Real Server Group 1)
`>> Real server group 200# add 3`	(Select Real Server 2 to Group 200)
`>> Real server group 200# add 4`	(Select Real Server 3 to Group 200)

11. Configure ports 4 and 5, which are connected to the real servers, for server processing.

```
>> Real server group 200# /cfg/slb/port 4/server ena
>> SLB port 4# /cfg/slb/port 5/server ena
```

12. Enable SLB.

```
>> Real server group 1# /cfg/slb/on
```

13. Create a filter to prevent server-to-server traffic from being redirected.

`>> Layer 4# /cfg/slb/filt 10`	(Select Filter 10)
`>> Filter 10# sip any`	(From any source IP address)
`>> Filter 10# dip 20.1.1.0`	(To base IP address for IF 5)
`>> Filter 10# dmask 255.255.255.0`	(For the range of addresses)
`>> Filter 10# proto any`	(For any protocol)

`>> Filter 10# action allow`	(Allow traffic)
`>> Filter 10# ena`	(Enable the filter)

14. Create the redirection filter. This filter redirects outbound traffic, load balancing it among the defined real servers in the group. In this case, the real servers represent IP interfaces on the dirty-side Alteon.

`>> Filter 10# /cfg/slb/filt 15`	(Select Filter 15)
`>> Filter 15# sip any`	(From any source IP address)
`>> Filter 15# dip any`	(To any destination IP address)
`>> Filter 15# proto any`	(For any protocol)
`>> Filter 15# action redir`	(Perform redirection)
`>> Filter 15# group 1`	(To real server Group 1)
`>> Filter 15# ena`	(Enable the filter)

15. Add the filters to the ingress ports for the outbound packets.

    Redirection filters are needed on all the ingress ports on the clean-side Alteon. Ingress ports are any that attach to real servers or internal clients on the clean-side of the network. In this case, two real servers are attached to the clean-side Alteon on ports 4 and 5.

`>> Filter 15# /cfg/slb/port 4`	(Select Ingress Port 4)
`>> SLB Port 4# add 10`	(Add the filter to the ingress port)
`>> SLB Port 4# add 15`	(Add the filter to the ingress port)
`>> SLB Port 4$ filt ena`	(Enable filtering on the port)
`>> SLB Port 4# /cfg/slb/port 5`	(Select Ingress Port 5)
`>> SLB Port 5# add 10`	(Add the filter to the ingress port)
`>> SLB Port 5# add 15`	(Add the filter to the ingress port)
`>> SLB Port 5# filt ena`	(Enable filtering on the port)

16. Define static routes to the dirty-side IP interfaces, using the firewalls as gateways.

    One static route is required for each firewall path being load balanced. In this case, two paths are required: one that leads to dirty-side IF 2 (10.1.1.1) through the first firewall (10.1.3.10) as its gateway and one that leads to dirty-side IF 3 (10.1.2.1) through the second firewall (10.1.4.10) as its gateway.

    Note: Configuring static routes for FWLB does not require IP forwarding to be turned on.

```
>> SLB Port 5# /cfg/l3/route/ip4
>> IP Static Route# add 10.1.1.1 255.255.255.255 10.1.3.10
>> IP Static Route# add 10.1.2.1 255.255.255.255 10.1.4.10
```

Note: When adding an IPv4 static route, if you are using FWLB and you define two IP interfaces on the same subnet, where one IP interface has a subnet of the host which is also included in the subnet of the second interface, you must specify the interface.

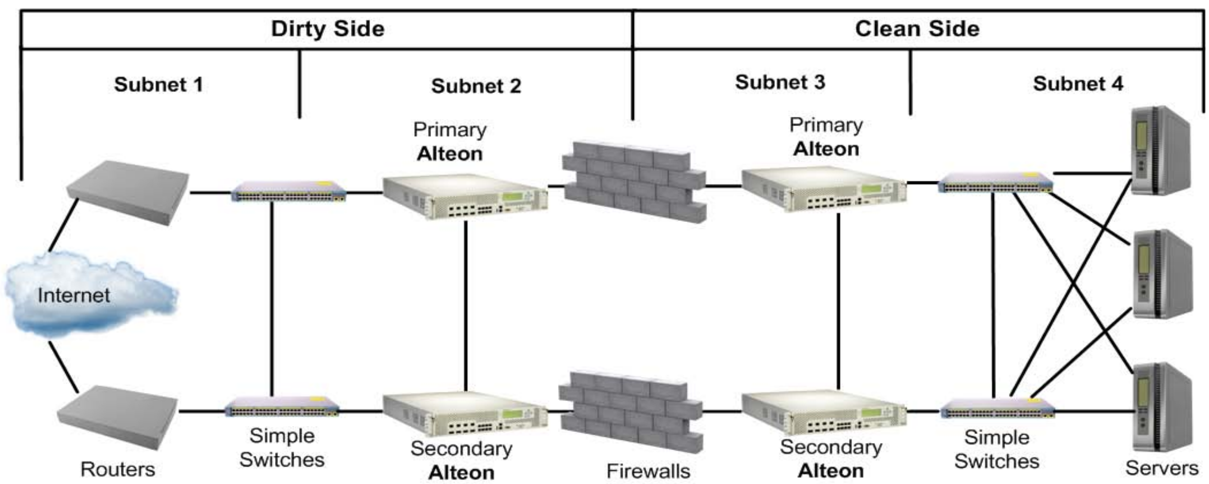
17. Apply and save the configuration changes.

Four-Subnet FWLB

The four-subnet FWLB method is often deployed in large networks that require high availability solutions. This method uses filtering, static routing, and Virtual Router Redundancy Protocol (VRRP) to provide a parallel firewall operation between redundant Alteons.

Figure 110 - Four-Subnet FWLB Network Topology, page 668 illustrates one possible network topology using the four-subnet method:

Figure 110: Four-Subnet FWLB Network Topology



This network is classified as a high availability network because no single component or link failure can cause network resources to become unavailable. Simple switches and vertical block interswitch connections are used to provide multiple paths for network failover. However, the interswitch links may be trunked together with multiple ports for additional protection from failure.

Note: Other topologies that use internal hubs, or diagonal cross-connections between Alteons and simple switches are also possible. While such topologies may resolve networking issues in special circumstances, they can make configuration more complex and can cause restrictions when using advanced features such as active-active VRRP, free-metric FWLB, or content-intelligent switching.

In the example topology in Figure 110 - Four-Subnet FWLB Network Topology, page 668, the network is divided into four sections:

- Subnet 1 includes all equipment between the exterior routers and dirty-side Alteons.
- Subnet 2 includes the dirty-side Alteons with their interswitch link, and dirty-side firewall interfaces.
- Subnet 3 includes the clean-side firewall interfaces, and clean-side Alteons with their interswitch link.
- Subnet 4 includes all equipment between the clean-side Alteons and their servers.

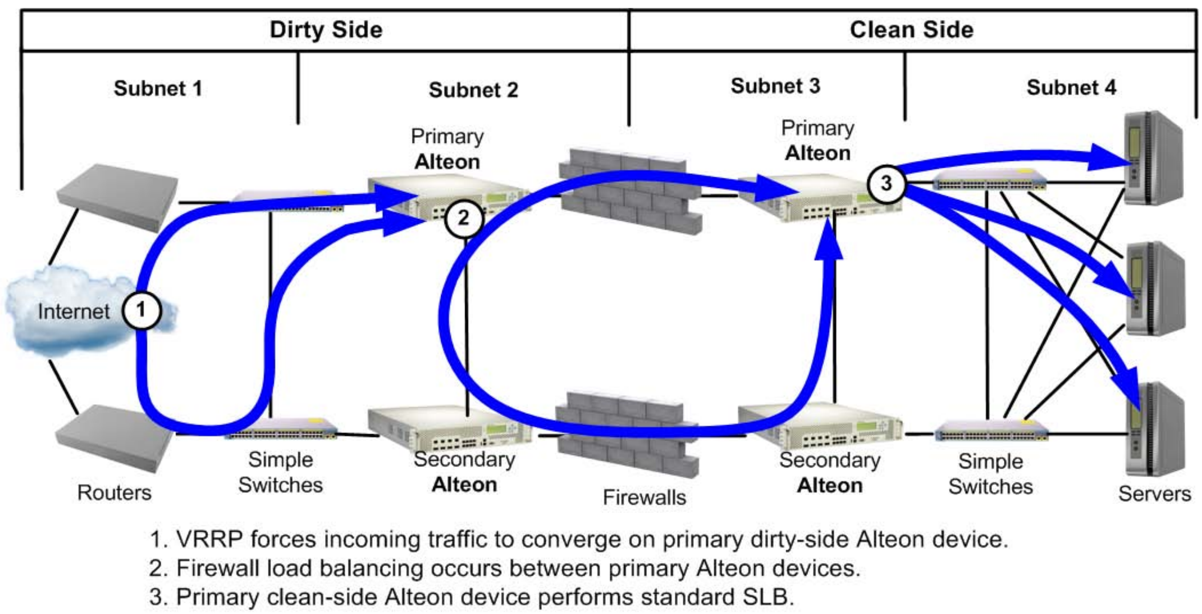
In this network, external traffic arrives through both routers. Since VRRP is enabled, one of the dirty-side Alteons acts as the primary and receives all traffic. The dirty-side primary Alteon performs FWLB similar to basic FWLB—a redirection filter splits traffic into multiple streams which are routed through the available firewalls to the primary clean-side Alteon.

Just as with the basic method, four-subnet FWLB uses the hash metric to distribute firewall traffic and maintain persistence, though other load-balancing metrics can be used by configuring an additional transparent load balancing option (see Free-Metric FWLB, page 683).

Four-Subnet FWLB Implementation

In the example in Figure 111 - Example Four-Subnet FWLB Implementation, page 669, traffic between the redundant Alteons is load balanced among the available firewalls:

Figure 111: Example Four-Subnet FWLB Implementation



1. Incoming traffic converges on the primary dirty-side Alteon.

   External traffic arrives through redundant routers. A set of interconnected switches ensures that both routers have a path to each dirty-side Alteon.

   VRRP is configured on each dirty-side Alteon so that one acts as the primary routing switch. If the primary fails, the secondary takes over.

2. FWLB is performed between primary Alteons.

   Just as with basic FWLB, filters on the ingress ports of the dirty-side Alteon redirect traffic to a real server group composed of multiple IP addresses. This configuration splits incoming traffic into multiple streams. Each stream is then routed toward the primary clean-side Alteon through a different firewall.

   Although other load-balancing metrics can be used in some configurations (see Free-Metric FWLB, page 683), the distribution of traffic within each stream is normally based on a mathematical hash of the IP source and destination addresses. Hashing ensures that each request and its related responses use the same firewall (a feature known as persistence), and that the streams are statistically equal in traffic load.

3. The primary clean-side Alteon forwards the traffic to its destination.

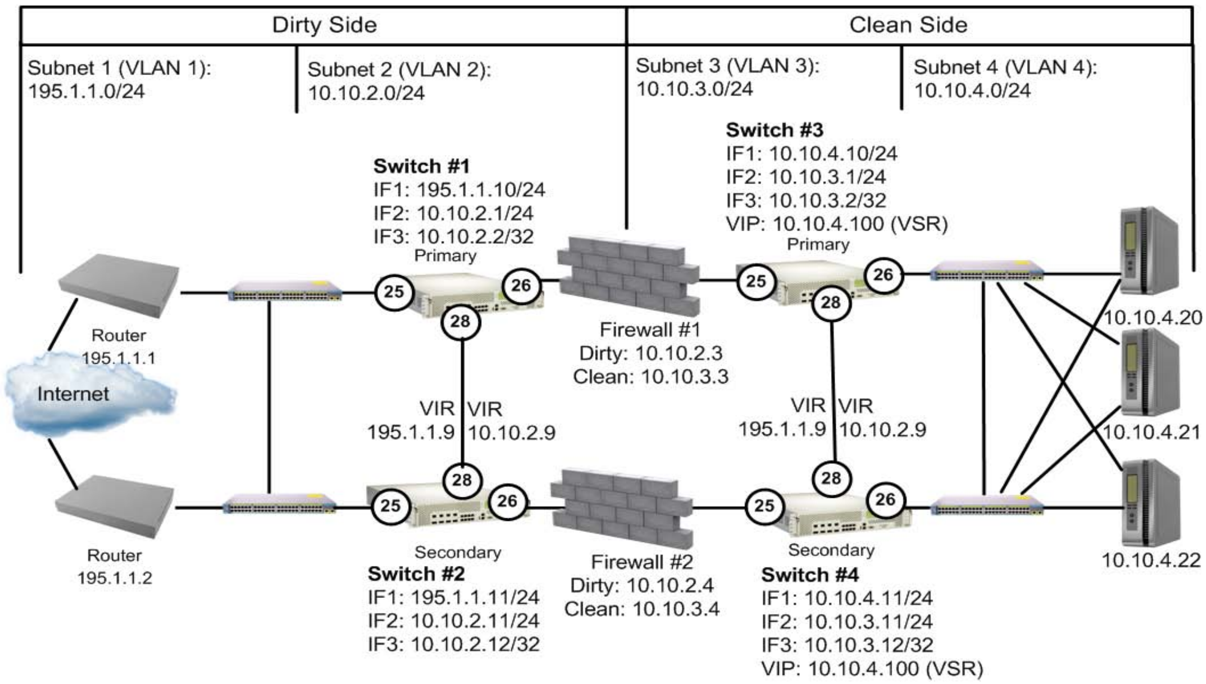
   After traffic arrives at the primary clean-side Alteon, it is forwarded to its destination. In this example, Alteon uses regular SLB settings to select a real server on the internal network for each incoming request.

   The same process is used for outbound server responses–a filter on the clean-side Alteon splits the traffic, and static routes forward each response stream back through the same firewall that forwarded the original request.

Configuring Four-Subnet FWLB

Figure 112 - Example Four-Subnet FWLB Configuration, page 670 illustrates an example network for four-subnet FWLB. While other complex topologies are possible, this example assumes a high availability network using block (rather than diagonal) interconnections between Alteons.

Figure 112: Example Four-Subnet FWLB Configuration



Note: The port designations of both dirty-side Alteons are identical, as are the port designations of both clean-side Alteons. This simplifies configuration by allowing you to synchronize the configuration of each primary Alteon with the secondary.

Four-subnet FWLB configuration includes the following procedures:

- Configure routers and firewalls and test them for proper operation, as explained in Configure the Routers, page 671 and Configure the Firewalls, page 671.
- Configure VLANs, IP interfaces, and static routes on all Alteons and test them, as explained in:
  - Configure the Primary Dirty-Side Alteon, page 672—Configure FWLB groups and redirection filters on the primary dirty-side Alteon.
  - Configure the Secondary Dirty-Side Alteon, page 673—Configure and synchronize VRRP on the primary dirty-side Alteon.
  - Configure the Primary Clean-Side Alteon, page 675—Configure FWLB and SLB groups, and add FWLB redirection filters on the primary clean-side Alteon.
  - Configure the Secondary Clean-Side Alteon, page 676—Configure VRRP on the primary clean-side Alteon and synchronize the secondary.
  - Verify Proper Connectivity, page 677
- Configure secondary Alteons with VRRP support settings, as explained in:
  - Configure VRRP on the Secondary Dirty-Side Alteon, page 677
  - Configure VRRP on the Secondary Clean-Side Alteon, page 678

— Complete Primary Dirty-Side Alteon Configuration, page 678
— Complete Primary Clean-Side Alteon Configuration, page 680

Configure the Routers

The routers must be configured with a static route to the destination services being accessed by the external clients.

In this example, the external clients are configured to connect to services at a publicly advertised IP address on this network. Since the real servers are load balanced behind a virtual server on the clean-side Alteon using normal SLB settings, the routers require a static route to the virtual server IP address. The next hop for this static route is the Alteon Virtual Interface Router (VIR), which is in the same subnet as the routers:

```
Route Added: 10.10.4.100 (to clean-side virtual server) via 195.1.1.9 (Subnet 1
VIR)
```

Configure the Firewalls

Before you configure Alteons, the firewalls must be properly configured. For incoming traffic, each firewall must be configured with a static route to the clean-side virtual server, using the VIR in its clean-side subnet as the next hop. For outbound traffic, each firewall must use the VIR in its dirty-side subnet as the default gateway.

As shown in Table 60, in this example the firewalls are configured with the following IP addresses:

Table 60: Four-Subnet Firewall IP Address Configuration

Firewall	IP Addresses
Firewall 1	
Dirty-side IP interface Clean-side IP interface Default Gateway Route added	10.10.2.3 10.10.3.3 10.10.2.9 (Subnet 2 VIR) 10.10.4.100 (virtual server) via 10.10.3.9 (Subnet 3 VIR)
Firewall 2	
Dirty-side IP interface Clean-side IP interface Default gateway Route added	10.10.2.4 10.10.3.4 10.10.2.9 (dirty-side VIR) 10.10.4.100 (virtual server) via 10.10.3.9 (Subnet 3 VIR)

The firewalls must also be configured with rules that determine which types of traffic will be forwarded through the firewall and which will be dropped. All firewalls participating in FWLB must be configured with the same set of rules.

Note: It is important to test the behavior of the firewalls prior to adding FWLB.

Configure the Primary Dirty-Side Alteon

The following is an example configuration for a primary dirty-side Alteon.

To configure the primary dirty-side Alteon

1. Configure VLANs on the primary dirty-side Alteon. Two VLANs are required. VLAN 1 includes port 25 for the Internet connection. VLAN 2 includes port 26 for the firewall connection, and port 28 for the interswitch connection.

```
>> /cfg/l2/vlan 2
>> add 26
>> add 28
>> ena
```

Note: Port 25 is part of VLAN 1 by default and does not require manual configuration.

2. Configure IP interfaces on the primary dirty-side Alteon.

   Three IP interfaces (IFs) are used. IF 1 is on placed on Subnet 1. IF 2 is used for routing traffic through the top firewall. IF 3 is used for routing traffic through the lower firewall. To avoid confusion, IF 2 and IF 3 are used in the same way on all Alteons.

```
>> /cfg/l3/if 1
>> mask 255.255.255.0
>> addr 195.1.1.10
>> ena
>> /cfg/l3/if 2
>> mask 255.255.255.0
>> addr 10.10.2.1
>> vlan 2
>> ena
>> /cfg/l3/if 3
>> mask 255.255.255.255
>> addr 10.10.2.2
>> vlan 2
>> ena
```

Note: By configuring the IP interface mask prior to the IP address, the broadcast address is calculated. Also, only the first IP interface in a given subnet is given the full subnet range mask. Subsequent IP interfaces (such as IF 3) are given individual masks.

3. Turn Spanning Tree Protocol (STP) off for the primary dirty-side Alteon.

```
>> /cfg/l2/stg #/off
```

4. Configure static routes on the primary dirty-side Alteon.

   Four static routes are required:

   — To primary clean-side IF 2 via Firewall 1 using dirty-side IF 2

— To primary clean-side IF 3 via Firewall 2 using dirty-side IF 3
— To secondary clean-side IF 2 via Firewall 1 using dirty-side IF 2
— To secondary clean-side IF 3 via Firewall 2 using dirty-side IF 3

Note: IF 2 is used on all Alteons whenever routing through the top firewall, and IF 3 is used on all Alteons whenever routing through the lower firewall.

The static route **add** command uses the following format:

```
add <destination address> <dest. mask> <gateway address> <source interface>
```

This example requires the following static route configuration:

```
>> /cfg/l3/route/ip4|ip6
>> # add 10.10.3.1 255.255.255.255 10.10.2.3 2
>> # add 10.10.3.2 255.255.255.255 10.10.2.4 3
>> # add 10.10.3.11 255.255.255.255 10.10.2.3 2
>> # add 10.10.3.12 255.255.255.255 10.10.2.4 3
```

Note: When defining static routes for FWLB, it is important to specify the source IP interface numbers.

5. When dynamic routing protocols are not used, configure a gateway to the external routers.

```
>> /cfg/l3/gw 1/addr 195.1.1.1
>> /cfg/l3/gw 2/addr 195.1.1.2
```

6. Apply and save the configuration, and reboot Alteon.

```
>> # apply
>> # save
>> # /boot/reset
```

Configure the Secondary Dirty-Side Alteon

The following is an example configuration for a secondary dirty-side Alteon.

To configure the secondary dirty-side Alteon

Except for the IP interfaces, this configuration is identical to the configuration of the primary dirty-side Alteon.

1. Configure VLANs on the secondary dirty-side Alteon.

```
>> /cfg/l2/vlan 2
>> add 26
>> add 28
>> ena
```

2. Configure IP interfaces on the secondary dirty-side Alteon.

```
>> /cfg/l3/if 1
>> mask 255.255.255.0
>> addr 195.1.1.11
>> ena
>> /cfg/l3/if 2
>> mask 255.255.255.0
>> addr 10.10.2.11
>> vlan 2
>> ena
>> /cfg/l3/if 3
>> mask 255.255.255.255
>> addr 10.10.2.12
>> vlan 2
>> ena
```

3. Turn STP off for the secondary dirty-side Alteon.

```
>> /cfg/l2/stg #/off
```

4. Configure static routes on the secondary dirty-side Alteon.

```
>> /cfg/l3/frwd/route
>> # add 10.10.3.1 255.255.255.255 10.10.2.3 2
>> # add 10.10.3.2 255.255.255.255 10.10.2.4 3
>> # add 10.10.3.11 255.255.255.255 10.10.2.3 2
>> # add 10.10.3.12 255.255.255.255 10.10.2.4 3
```

5. When dynamic routing protocols are not used, configure a gateway to the external routers on the secondary dirty-side Alteon.

```
>> /cfg/l3/gw 1/addr 195.1.1.1
>> /cfg/l3/gw 2/addr 195.1.1.2
```

6. Apply and save the configuration, and reboot Alteon.

```
>> # apply
>> # save
>> # /boot/reset
```

Configure the Primary Clean-Side Alteon

The following is an example configuration for a primary clean-side Alteon.

To configure the primary clean-side Alteon

1. Configure VLANs on the primary clean-side Alteon.

   Two VLANs are required. VLAN 3 includes the firewall port and interswitch connection port. VLAN 4 includes the port that attaches to the real servers.

```
>> /cfg/l2/vlan 2
>> add 25
>> add 28
>> ena
>> /cfg/l2/vlan 4
>> add 26
>> ena
```

2. Configure IP interfaces on the primary clean-side Alteon.

```
>> /cfg/l3/if 1
>> mask 255.255.255.0
>> addr 10.10.4.10
>> vlan 4
>> ena
>> /cfg/l3/if 2
>> mask 255.255.255.0
>> addr 10.10.3.1
>> vlan 3
>> ena
>> /cfg/l3/if 3
>> mask 255.255.255.255
>> addr 10.10.3.2
>> vlan 3
>> ena
```

3. Turn STP off for the primary clean-side Alteon.

```
>> /cfg/l2/stg #/off
```

   Spanning Tree Protocol is disabled because VLANs prevent broadcast loops.

4. Configure static routes on the primary clean-side Alteon.

   Four static routes are needed:

   — To primary dirty-side IF 2 via Firewall 1 using clean-side IF 2

   — To primary dirty-side IF 3 via Firewall 2 using clean-side IF 3

   — To secondary dirty-side IF 2 via Firewall 1 using clean-side IF 2

   — To secondary dirty-side IF 3 via Firewall 2 using clean-side IF 3

   The static route **add** command uses the following format:

```
add <destination address> <dest. mask> <gateway address> <source interface>
```

This example requires the following static route configuration:

```
>> /cfg/l3/frwd/route
>> # add 10.10.2.1 255.255.255.255 10.10.3.3 2
>> # add 10.10.2.2 255.255.255.255 10.10.3.4 3
>> # add 10.10.2.11 255.255.255.255 10.10.3.3 2
>> # add 10.10.2.12 255.255.255.255 10.10.3.4 3
```

5. Apply and save the configuration, and reboot Alteon.

```
>> # apply
>> # save
>> # /boot/reset
```

Configure the Secondary Clean-Side Alteon

The following is an example configuration for a secondary clean-side Alteon.

To configure the secondary clean-side Alteon

1. Configure VLANs on the secondary clean-side Alteon.

```
>> /cfg/l2/vlan 3
>> add 25
>> add 28
>> ena
>> /cfg/l2/vlan 4
>> add 26
>> ena
```

2. Configure IP interfaces on the secondary clean-side Alteon.

```
>> /cfg/l3/if 1
>> mask 255.255.255.0
>> addr 10.10.4.11
>> vlan 4
>> ena
>> /cfg/l3/if 2
>> mask 255.255.255.0
>> addr 10.10.3.11
>> vlan 3
>> ena
>> /cfg/l3/if 3
>> mask 255.255.255.255
>> addr 10.10.3.12
>> vlan 3
>> ena
```

3. Turn STP off for the secondary clean-side Alteon.

```
>> /cfg/l2/stg #/off
```

Spanning Tree Protocol is disabled because VLANs prevent broadcast loops.

4. Configure static routes on the secondary clean-side Alteon.

```
>> /cfg/l3/frwd/route
>> # add 10.10.2.1 255.255.255.255 10.10.3.3 2
>> # add 10.10.2.2 255.255.255.255 10.10.3.4 3
>> # add 10.10.2.11 255.255.255.255 10.10.3.3 2
>> # add 10.10.2.12 255.255.255.255 10.10.3.4 3
```

5. Apply and save the configuration, and reboot Alteon.

```
>> # apply
>> # save
>> # /boot/reset
```

Verify Proper Connectivity

To verify proper configuration at this point in the process, use the ping option to test network connectivity. At each Alteon, you should receive a valid response when pinging the destination addresses established in the static routes.

For example, on the secondary clean-side Alteon, the following commands should receive a valid response:

```
>> # ping 10.10.2.1
Response; 10.10.2.1: #1 OK, RTT 1 msec.
>> # ping 10.10.2.2
Response; 10.10.2.2: #1 OK, RTT 1 msec.
>> # ping 10.10.2.11
Response; 10.10.2.11: #1 OK, RTT 1 msec.
>> # ping 10.10.2.12
Response; 10.10.2.12: #1 OK, RTT 1 msec.
```

Configure VRRP on the Secondary Dirty-Side Alteon

The secondary dirty-side Alteon must be configured with the primary as its peer. Once this is done, the secondary Alteon receives the remainder of its configuration from the primary when synchronized in a later step.

In this example, the secondary Alteon is configured to use primary dirty-side Interface 1 as its peer.

```
>> # /cfg/l3/vrrp/on
>> # /cfg/slb
>> # on
>> # sync/peer 1
>> # addr 195.1.1.10
>> # ena
>> # apply
>> # save
```

Configure VRRP on the Secondary Clean-Side Alteon

In this example, the secondary Alteon uses primary clean-side Interface 1 as its peer.

```
>> # /cfg/l3/vrrp/on
>> # /cfg/slb
>> # on
>> # sync/peer 1
>> # addr 10.10.4.10
>> # ena
>> # apply
>> # save
```

Complete Primary Dirty-Side Alteon Configuration

The following is an example configuration for a primary dirty-side Alteon.

To complete the primary dirty-side Alteon configuration

1. Create an FWLB real server group on the primary dirty-side Alteon.

   A real server group is used as the target for the FWLB redirection filter. Each IP address that is assigned to the group represents a path through a different firewall. In this case, since two firewalls are used, two addresses are added to the group.

   Earlier, it was stated that this example uses IF 2 on all Alteons whenever routing through the top firewall, and IF 3 on all Alteons whenever routing through the lower firewall. Therefore, the first address represents the primary clean-side IF 2, and the second represents the primary clean-side IF 3.

```
>> # /cfg/slb
>> # on
>> # real 1
>> # rip 10.10.3.1
>> # ena
>> # /cfg/slb/real 2
>> # rip 10.10.3.2
>> # ena
>> # /cfg/slb/group 1
>> # add 1
>> # add 2
>> # metric hash
```

   Using the hash metric, all traffic between specific IP source/destination address pairs flows through the same firewall, ensuring that sessions established by the firewalls are maintained for their duration (persistence).

   Note: Other load balancing metrics, such as leastconns, roundrobin, minmiss, response, and bandwidth, can be used when enabling the transparent load balancing option. For more information, see Free-Metric FWLB, page 683.

2. Create the FWLB filters.

   Three filters are required on the port attaching to the routers:

   — Filter 10 prevents local traffic from being redirected.

— Filter 20 prevents VRRP traffic (and other multicast traffic on the reserved 224.0.0.0/24 network) from being redirected.

— Filter 2048 redirects the remaining traffic to the firewall group.

```
>> # /cfg/slb/filt 10
>> # dip 195.1.1.0
>> # dmask 255.255.255.0
>> # ena
>> # /cfg/slb/filt 20
>> # dip 224.0.0.0
>> # dmask 255.255.255.0
>> # ena
>> # /cfg/slb/filt 2048
>> # action redir
>> # group 1
>> # ena
>> # /cfg/slb/port 1
>> # filt ena
>> # add 10
>> # add 20
>> # add 2048
```

3. Configure VRRP on the primary dirty-side Alteon. VRRP in this example requires two virtual routers: one for the subnet attached to the routers and one for the subnet attached to the firewalls.

```
>> # /cfg/l3/vrrp 2
>> # on
>> # vr 1
>> # vrid 1                          (Configure Virtual Router 1)
>> # addr 195.1.1.9                  (For the subnet attached to the
                                     routers)
>> # if 1
>> # prio 101
>> # share dis
>> # ena
>> # track
>> # ifs ena
>> # ports ena
>> # /cfg/l3/vrrp/vr 2
>> # vrid 2                          (Configure Virtual Router 2)
>> # addr 10.10.2.0                  (For the subnet attached to the
                                     firewall)
>> #if 2
>> # prio 101
>> # share dis
>> # ena
>> # track
>> # ifs ena
>> # ports ena
```

4. Configure the VRRP peer on the primary dirty-side Alteon.

```
>> # /cfg/slb/sync
>> # prios d
>> # peer 1
>> # ena
>> # addr 195.1.1.11
```

5. Apply and save your configuration changes.

```
>> # apply
>> # save
```

6. Synchronize primary and secondary dirty-side Alteons for the VRRP configuration.

```
>> # /oper/slb/sync
```

Complete Primary Clean-Side Alteon Configuration

The following is an example configuration for a primary clean-side Alteon.

To complete the primary clean-side Alteon configuration

1. Create an FWLB real server group on the primary clean-side Alteon.

   A real server group is used as the target for the FWLB redirection filter. Each IP address assigned to the group represents a return path through a different firewall. In this case, since two firewalls are used, two addresses are added to the group. The two addresses are the interfaces of the dirty-side Alteon, and are configured as if they are real servers.

   Note: IF 2 is used on all Alteons whenever routing through the top firewall, and IF 3 is used on all Alteons whenever routing through the lower firewall.

```
>> # /cfg/slb
>> # on
>> # real 1
>> # rip 10.10.2.1                                  (IF2 of the primary dirty-side
                                                    Alteon)
>> # ena
>> # /cfg/slb/real 2
>> # rip 10.10.2.2                                  (IF2 of the primary dirty-side
                                                    Alteon)
>> # ena
>> # /cfg/slb/group 1
>> # add 1
>> # add 2
>> # metric hash
```

Note: The clean-side Alteon must use the same metric as defined on the dirty side. For information on using metrics other than hash, see Free-Metric FWLB, page 683.

2. Create an SLB real server group on the primary clean-side Alteon to which traffic will be load balanced.

   The external clients are configured to connect to HTTP services at a publicly advertised IP address. The servers on this network are load balanced by a virtual server on the clean-side Alteon. SLB options are configured as follows:

`>> # /cfg/slb`	(Select the *SLB* menu)
`>> # real 20`	(Select Real Server 20)
`>> # rip 10.10.4.20`	(Set IP address of Real Server 20)
`>> # ena`	(Enable)
`>> # /cfg/slb.real 21`	(Select Real Server 21)
`>> # rip 10.10.4.21`	(Set IP address of Real Server 21)
`>> # ena`	(Enable)
`>> # /cfg/slb/real 22`	(Select Real Server 22)
`>> # rip 10.10.4.22`	(Set IP address of Real Server 22)
`>> # ena`	(Enable)
`>> # /cfg/slb/group 2`	(Select Real Server group 2)
`>> # add 20`	(Add the Real Servers to the group)
`>> # add 21`	
`>> # add 22`	
`>> # metric leastconns`	(Select least connections as the load-balancing metric)
`>> # /cfg/slb/virt 1`	(Select the *Virtual Server 1* menu)
`>> # vip 10.10.4.100`	(Set the virtual server IP address)
`>> # service http`	(Select HTTP for load balancing)
`>> # group 2`	(Add Real Server Group 2)
`>> # ena`	(Enable the virtual server)
`>> # /cfg/slb/port/26/server ena`	(Enable server processing on the port connected to the real servers)
`>> # /cfg/slb/port/25/client ena`	(Enable client processing on the port connected to the firewall)
`>> # /cfg/slb/port/28/client ena`	(Enable client processing on the interswitch connection)

Note: The virtual server IP address configured in this step will also be configured as a Virtual Server Router (VSR) when VRRP is configured in a later step.

3. Create the FWLB filters on the primary clean-side Alteon.

   Three filters are required on the port attaching to the real servers:

   — Filter 10 prevents local traffic from being redirected.
   — Filter 20 prevents VRRP traffic from being redirected.
   — Filter 2048 redirects the remaining traffic to the firewall group.

```
>> # /cfg/slb/filt 10
>> # dip 10.10.4.0
>> # dmask 255.255.255.0
>> # ena
>> # /cfg/slb/filt 20
>> # dip 224.0.0.0
>> # dmask 255.255.255.0
>> # ena
>> # /cfg/slb/filt 2048
>> # action redir
>> # group 1
>> # ena
>> # /cfg/slb/port 4
>> # filt ena
>> # add 10
>> # add 20
>> # add 2048
```

4. Configure VRRP on the primary clean-side Alteon.

   VRRP in this example requires two virtual routers to be configured: one for the subnet attached to the real servers and one for the subnet attached to the firewalls.

```
>> # /cfg/l3/vrrp
>> # on
>> # vr 1
>> # vrid 3
>> # addr 10.10.4.9
>> # if 1
>> # prio 100
>> # share dis
>> # ena
>> # track
>> # ifs ena
>> # ports ena
>> # /cfg/l3/vrrp/vr 2
>> # vrid 4
>> # addr 10.10.3.9
>> # if 2
>> # prio 101
>> # share dis
>> # ena
>> # track
>> # ifs ena
>> # ports ena
```

A third virtual router is required for the virtual server used for optional SLB.

```
>> # /cfg/l3/vrrp/vr 3
>> # vrid 5
>> # addr 10.10.4.100
>> # prio 102
>> # share dis
>> # ena
>> # track
>> # ifs ena
>> # ports ena
```

5. Configure the peer on the primary clean-side Alteon.

```
>> # /cfg/slb/sync
>> # prios d
>> # peer 1
>> # ena
>> # addr 10.10.4.11
```

6. Apply and save your configuration changes.

```
>> # apply
>> # save
```

7. Synchronize primary and secondary dirty-side Alteons for the VRRP configuration.

```
>> # /oper/slb/sync
```

Advanced FWLB Concepts

This section includes the following topics:

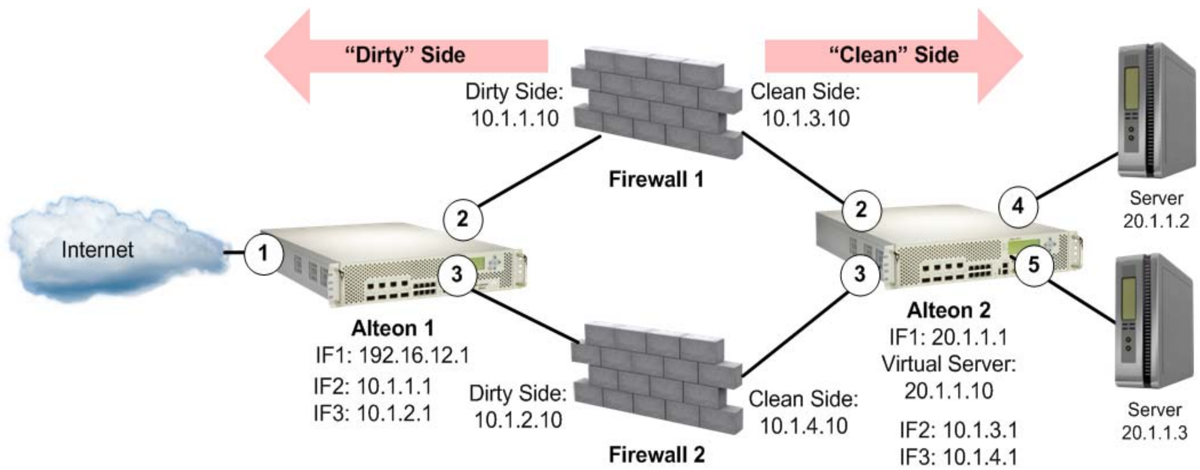
Free-Metric FWLB

Free-metric FWLB lets you use load-balancing metrics other than hash, such as leastconns, roundrobin, minmiss, response, and bandwidth, for more versatility. The free-metric method uses the transparent load balancing option, which can be used with basic FWLB or four-subnet FWLB networks.

Free-Metric with Basic FWLB

This example uses the basic FWLB network as illustrated in Figure 113 - Basic FWLB Network, page 684:

Figure 113: Basic FWLB Network



To configure free-metric FWLB in a basic network

1. On the clean-side Alteon, enable RTS on the ports attached to the firewalls (Ports 2 and 3). Enable filter and server processing on ports 2 and 3 so that the responses from the real server are looked-up in the session table.

```
>> # /cfg/slb/port 2/rts enable
>> # /cfg/slb/port 3/rts enable
```

2. On the clean-side Alteon, remove the redirection filter from the ports attached to the real servers (Ports 4 and 5), but ensure that filter processing is enabled.

   The redirection filter is removed so that the return packet traverses through the same firewall. If the firewalls synchronize their states, then it is not required to remove the redirection filter. Filter processing is enabled to make use of the RTS-created sessions.

```
>> # /cfg/slb/port 4/rem 2048
>> # filt ena
>> # /cfg/slb/port 5/rem 2048
>> # filt ena
```

   Use the hash metric if the session is from an FTP or RTSP servers.

3. On the dirty-side Alteon, set the FWLB metric.

```
>> # /cfg/slb/group 1
>> # metric <metric type>
```

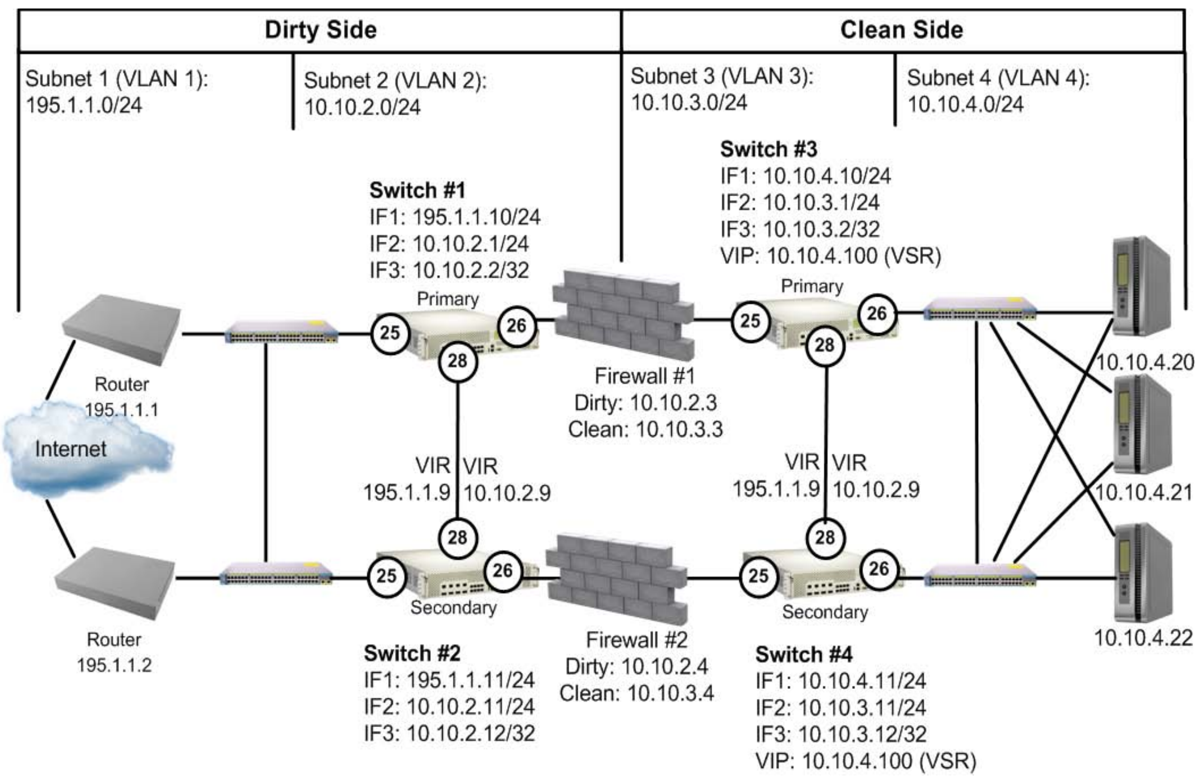
Any of the following load-balancing metrics can be used: hash, leastconns, roundrobin, minmiss, response, or bandwidth. See Metrics for Real Server Groups, page 180 for details on using each metric.

Note: Some metrics allow other options (such as weights) to be configured.

Free-Metric with Four-Subnet FWLB

This example uses the four-subnet FWLB network as illustrated in Figure 114 - Four-Subnet Network, page 685:

Figure 114: Four-Subnet Network



To use free-metric FWLB in a four-subnet FWLB network

1. On the clean-side Alteons, enable RTS on the ports attached to the firewalls (Port 3) and on the interswitch port (port 9).

   Enable filter and server processing on Ports 3 and 9, so that the responses from the real server are looked up in the session table on *both* clean-side Alteons:

```
>> # /cfg/slb/port 26/rts enable
>> # /cfg/slb/port 28/rts enable
```

2. On the clean-side Alteons, remove the redirection filter from the ports attached to the real servers (Ports 4), and ensure filter processing is enabled.

To view the original redirection filters that were configured for the four-subnet example, see step 3. Do this on *both* clean-side Alteons:

```
>> # /cfg/slb/port 26/rts enable
>> # filt ena
```

3. On the dirty-side Alteons, set the FWLB metric, on *both* dirty-side Alteons:

```
>> # /cfg/slb/group 1
>> # metric <metric type>
```

Any of the following load-balancing metrics can be used: hash, leastconns, roundrobin, minmiss, response, or bandwidth. See Metrics for Real Server Groups, page 180 for details on using each metric.

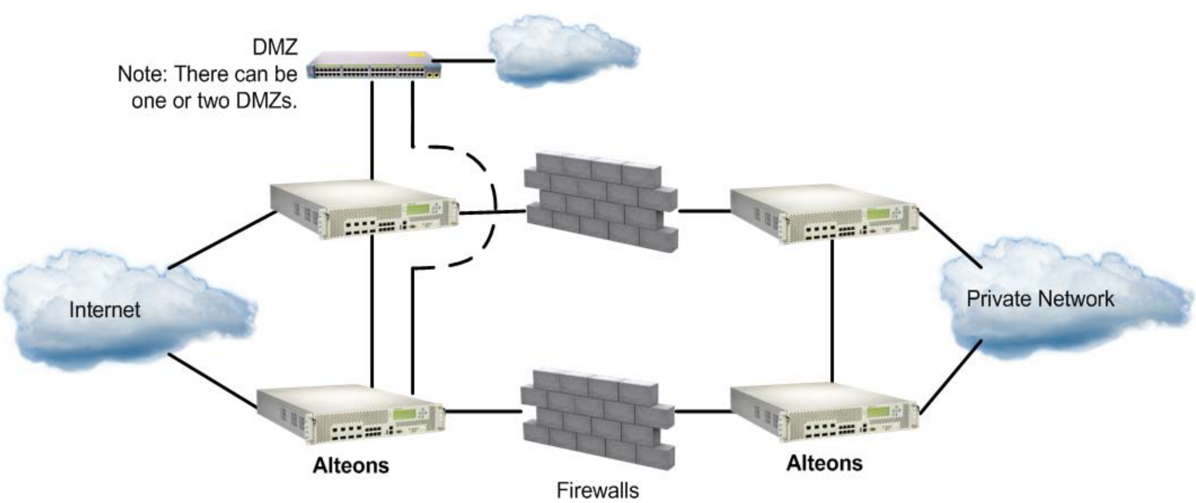
Note: Some metrics allow other options (such as weights) to be configured.

Adding a Demilitarized Zone (DMZ)

Implementing a DMZ in conjunction with FWLB enables Alteon to perform traffic filtering, off-loading this task from the firewall. A DMZ is created by configuring FWLB with another real server group and a redirection filter towards the DMZ subnets.

The DMZ servers can be connected to Alteon on the dirty side of the firewall. A typical firewall load-balancing configuration with a DMZ is shown in Figure 115 - FWLB with a Demilitarized Done (DMZ), page 686:

Figure 115: FWLB with a Demilitarized Done (DMZ)



The DMZ servers can be attached to Alteon directly or through an intermediate hub or Alteon. Alteon is then configured with filters to permit or deny access to the DMZ servers. In this way, two levels of security are implemented: one that restricts access to the DMZ through the Alteon filters and another that restricts access to the clean network through the stateful inspection performed by the firewalls.

To add the filters required for the DMZ (to each Alteon)

1. On the dirty-side Alteon, create the filter to allow HTTP traffic to reach the DMZ Web servers.
   In this example, the DMZ Web servers use IP addresses 205.178.29.0/24.

```
>> # /cfg/slb/filt 80                     (Select Filter 80)
>> Filter 80# sip any                     (From any source IP address)
>> Filter 80# dip 205.178.29.0            (To the DMZ base destination)
>> Filter 80# dmask 255.255.255.0         (For the range of DMZ addresses)
>> Filter 80# proto tcp                   (For TCP protocol traffic)
>> Filter 80# sport any                   (From any source port)
>> Filter 80# dport http                  (To an HTTP destination port)
>> Filter 80# action allow                (Allow the traffic)
>> Filter 80# ena                         (Enable the filter)
```

2. Create another filter to deny all other traffic to the DMZ Web servers.

```
>> Filter 80# /cfg/slb/filt 89            (Select Filter 89)
>> Filter 89# sip any                     (From any source IP address)
>> Filter 89# dip 205.178.29.0            (To the DMZ base destination)
>> Filter 89# dmask 255.255.255.0         (For the range of DMZ addresses)
>> Filter 89# proto any                   (For TCP protocol traffic)
>> Filter 89# action deny                 (Allow the traffic)
>> Filter 89# ena                         (Enable the filter)
```

Note: The deny filter has a higher filter number than the allow filter. This is necessary so that the allow filter has the higher order of precedence.

3. Add the filters to the traffic ingress ports.

```
>> Filter 89# /cfg/slb/port 1             (Select the ingress port)
>> SLB Port 1# add 80                     (Add the allow filter)
>> SLB Port 1# add 89                     (Add the deny filter)
```

4. Apply and save the configuration changes.

```
>> SLB Port 1# apply
>> SLB Port 1# save
```

Firewall Health Checks

Basic FWLB health checking is automatic. No special configuration is necessary unless you want to tune the health checking parameters. For details, see Health Checking, page 481.

Firewall Service Monitoring

To maintain high availability, Alteon monitors firewall health status and send packets only to healthy firewalls. There are two methods of firewall service monitoring: ICMP and HTTP. Each Alteon monitors the health of the firewalls on a regular basis by pinging the IP interfaces configured on its partner Alteon on the other side of the firewall.

If an Alteon IP interface fails to respond to a user-specified number of pings, it (and, by implication, the associated firewall) is placed in a **Server Failed** state. When this happens, the partner Alteon stops routing traffic to that IP interface, and instead distributes it across the remaining healthy Alteon IP interfaces and firewalls.

When an Alteon IP interface is in the **Server Failed** state, its partner Alteon continues to send pings to it at user-configurable intervals. After a specified number of successful pings, the IP interface (and its associated firewall) is brought back into service.

For example, to configure Alteon to allow one-second intervals between health checks or pings, two failed health checks to remove the firewall, and four successful health checks to restore the firewall to the real server group, use the following command:

```
>> /cfg/slb/real <real server number> /inter 1/retry 2/restr 4
```

Physical Link Monitoring

Alteon also monitors the physical link status of ports connected to firewalls. If the physical link to a firewall goes down, that firewall is placed immediately in the **Server Failed** state. When Alteon detects that a failed physical link to a firewall has been restored, it brings the firewall back into service.

Using HTTP Health Checks

For those firewalls that do not permit ICMP pings to pass through, Alteon can be configured to perform HTTP health checks.

To use HTTP health checks

1. Set the health check type to HTTP instead of ICMP.

```
>> # /cfg/slb/group 1/health http
```

2. Enable HTTP access to Alteon.

```
>> # /cfg/sys/access/http ena
```

3. Configure a "dummy" redirect filter as the last filter (after the **redirect all** filter) to force the HTTP health checks to activate.

`>> # /cfg/slb/filt 2048`	(Select Filter 2048)
`>> Filter 2048# proto tcp`	(For TCP protocol traffic)
`>> Filter 2048# action redir`	(Redirect the traffic)
`>> Filter 2048# group 1`	(Set real server group for redirection)
`>> Filter 2048# rport http`	(Set real server port for redirection)
`>> Filter 2048# ena`	(Enable the filter)

Note: Enure that the number of each real filter is lower than the number of the "dummy" redirect filter.

4. Apply filter to the port directed to the firewall.

`>> # /cfg/slb/port #/add 2048`	(Add the dummy filter)

In addition to HTTP, Alteon lets you configure up to five (5) different TCP services to listen for health checks. For example, you can configure FTP and SMTP ports to perform health checks. For a list of other well-known application ports, see Table 20 - Well-Known Application Ports, page 175.

Chapter 24 – Virtual Private Network Load Balancing

The Virtual Private Network (VPN) load balancing feature allows Alteon to simultaneously load balance up to 255 VPN devices. Alteon records from which VPN server a session was initiated and ensures that traffic returns back to the same VPN server from which the session started.

The following topics are addressed in this chapter:

- Overview, page 691—Describes a VPN network and how VPN load balancing works in Alteon.
- VPN Load Balancing Configuration, page 693—Provides step-by-step instructions to configure VPN load balancing on a four-subnet network with four Alteons and two VPN devices.

Overview

A VPN is a connection that has the appearance and advantages of a dedicated link, but it occurs over a shared network. Using a technique called *tunneling*, data packets are transmitted across a routed network, such as the Internet, in a private tunnel that simulates a point-to-point connection. This approach enables network traffic from many sources to travel via separate tunnels across the infrastructure. It also enables traffic from many sources to be differentiated, so that it can be directed to specific destinations and receive specific levels of service.

VPNs provide the security features of a firewall, network address translation, data encryption, and authentication and authorization. Since most of the data sent between VPN initiators and terminators is encrypted, network devices cannot use information inside the packet to make intelligent routing decisions.

How VPN Load Balancing Works

VPN load balancing requires that all ingress traffic passing through a particular VPN must traverse the same VPN as it egresses back to the client. Traffic ingressing from the Internet is usually addressed to the VPNs, with the real destination encrypted inside the datagram. Traffic egressing the VPNs into the intranet contains the real destination in the clear.

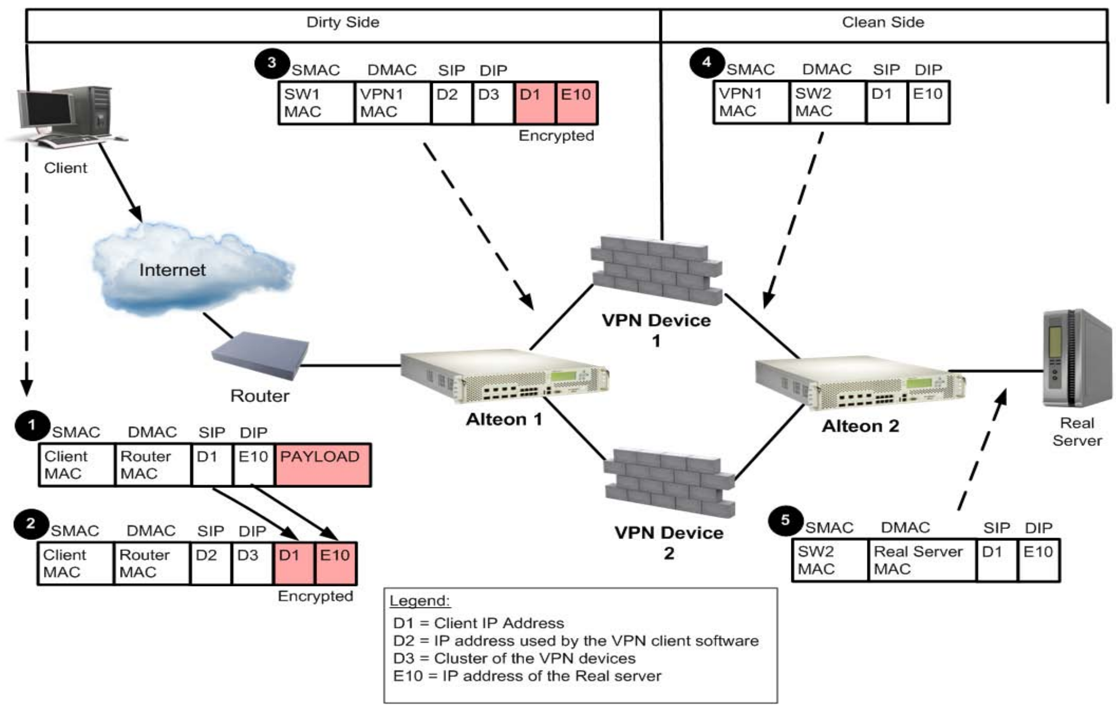
In many VPN/firewall configurations, it may not be possible to use the hash algorithm on the source and destination address, because the address may be encrypted inside the datagram. Also, the source and destination IP addresses of the packet may change as the packet traverses from the dirty-side Alteons to clean-side Alteons, and back.

To support VPN load balancing, Alteon records the state on frames entering Alteon to and from the VPNs. This session table ensures that the same VPN server handles all the traffic between an inside host and an outside client for a particular session.

Note: VPN load balancing is supported for connecting from remote sites to the network behind the VPN cluster IP address. A connection initiated from clients internal to the VPN gateways is not supported.

Basic frame flow, from the dirty side of the network to the clean side, is illustrated in Figure 116 - Basic Frame Flow, page 692. An external client is accessing an internal server. The VPN devices do not perform Network Address Translation (NAT).

Figure 116: Basic Frame Flow



The basic steps for this example configuration are as follows:

1. The client prepares to send traffic to the destination real server (with IP address E10).
2. The VPN client software encrypts the packet and sends it to the cluster IP address (D3) of the VPN devices.
3. Alteon 1 makes an entry in the session table and forwards the packet to VPN Device 1.

Note: Radware recommends using the hash load-balancing metric to select the VPN device.

4. VPN Device 1 strips the IP header and decrypts the encrypted IP header.
5. Alteon 2 forwards the packet to the real server.

If an entry is found, the frame is forwarded normally. If an entry is not found, Alteon determines which VPN device processed the frame by performing a lookup with the source MAC address of the frame. If the MAC address matches a MAC address of a VPN device, Alteon adds an entry to the session table so that reverse traffic is redirected to the same VPN device.

VPN Load-Balancing Persistence

VPN load-balancing persistence ensures that VPN sessions that exist in a load-balanced environment retain their persistence with the load-balanced server.

Since both the ISAKMP and IPSec protocols are used in a VPN environment, load balancing such an environment involves maintaining persistence for two protocols. For each user VPN login, the security associations must be established and key exchanges performed using the ISAKMP protocol before the IPSec protocols can be sent securely. Alteon redirects the ISAKMP request to a load-balanced VPN server and creates a session. Subsequent ISAKMP requests are sent to this session. When the associated IPSec session arrives, Alteon looks for the associated ISAKMP session using session lookup so that it can be load balanced to the same server. If the ISAKMP session is not found, the IPSec session is bound to a VPN server according to the previously configured load-balancing metrics.

VPN Load Balancing Configuration

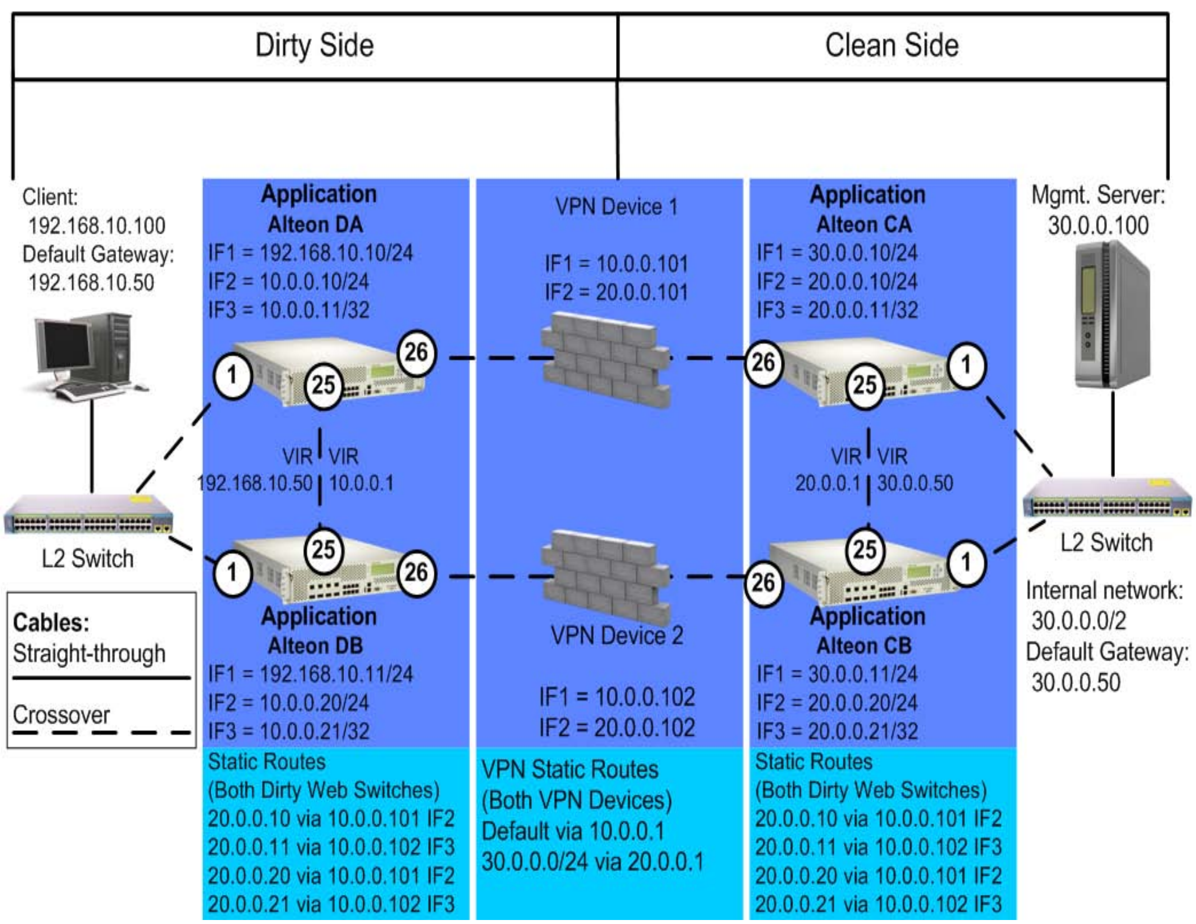
Before you start configuring Alteon for VPN load balancing, do the following:

- Configure Alteon with firewall load balancing (FWLB).
- Configure a filter to enable the transparent load balancing (Return to Source MAC address) option. This adds an opposite entry in the session table so that the return traffic matches its source MAC address.

`>> # /cfg/slb/filt 20/adv`	(Select the Advanced menu for Filter 20)
`>> Filter 20 Advanced# rtsrcmac ena`	(Enable Return to Source MAC Address)

Figure 117 - Example VPN Load-Balancing Configuration, page 693 illustrates VPN load balancing with two VPN devices and four Alteons in a four-subnet scenario:

Figure 117: Example VPN Load-Balancing Configuration



To configure the clean-side Alteon CA

1. Turn off BOOTP.

```
>> # /cfg/sys/bootp dis
```

2. Define and enable VLAN 2 for ports 25, and 26.

```
>> # /cfg/l2/vlan 2/ena/def 25 26
```

3. Turn off the Spanning Tree Protocol (STP)

```
>> # /cfg/l2/stg/off
```

4. Define the clean-side IP interfaces. Create one clean-side IP interface on a different subnet for each VPN device being load balanced.

`>> #/cfg/l3/if 1 ena`	(Select IP Interface 1 and enable)
`>> IP Interface 1# mask 255.255.0`	(Set subnet mask for Interface 1)
`>> IP Interface 1# addr 30.9.0.10`	(Set IP address for Interface 1)
`>> IP Interface 1 # vlan 1`	(For VLAN 1)
`>> IP Interface 1 #/ cfg/13/if 2/ena`	(Select IP Interface 2 and enable)
`>> IP Interface 2 # mask 255.255.255.0`	(Set subnet mask for Interface 2)
`>> IP Interface 2 # addr 20.0.0.10`	(Set IP address for Interface 2)
`>> IP Interface 2 # vlan 2`	(For VLAN 2)
`>> IP Interface 2 # /cgf/13/if 3/ena`	(Select IP Interface 3 and enable)
`>> IP Interface 3# mask 255.255.255.`	(Set subnet mask for Interface 3)
`>> IP Interface 3# addr 20.0.0.11`	(Set IP address for Interface 3)
`>> IP Interface 3# vlan 2`	(For VLAN 2)

5. Configure routes for each of the IP interfaces you configured in step 4 using the VPN devices as gateways. One static route for redirection is required for each VPN device being load balanced.

`>>#/cfg/l3/route`	
`>> IP Static Route# add 10.0.0.10`	(Static route destination IP address)
`>> IP Static Route# 255.255.255.255`	(Destination subnet mask)
`>> IP Static Route# 20.0.0101`	(Enter gateway IP address)
`>> IP Static Route# 2`	(For Interface 2)
`>> IP Static Route# add 10.0.0.11`	(Enter destination IP address)
`>> IP Static Route# 255.255.255.255`	(Destination subnet mask)
`>> IP Static Route# 20.0.0102`	(Enter gateway IP address)
`>> IP Static Route# 3`	(For Interface 3)

`>> IP Static Route# add 10.0.0.20`	(Enter destination IP address)
`>> IP Static Route# 255.255.255.255`	(Destination subnet mask)
`>> IP Static Route# 20.0.0.101`	(Enter gateway IP address)
`>> IP Static Route# 2`	(For Interface 2)
`>> IP Static Route# add 10.0.0.21`	(Static route destination IP address)
`>> IP Static Route# 255.155.255255`	(Destination subnet mask)
`>> IP Static Route# 20.0.0.102`	(Enter gateway IP address)
`>> IP Static Route# 3`	(For Interface 3)

6. Configure VRRP for Virtual Routers 1 and 2.

`>> # /cfg/l3/vrrp/on`	(Enable VRRP)
`>> Virtual Router Redundancy Protocol# vr 1`	(Select *Virtual Router 1* menu)
`>> VRRP Virtual Router 1# ena`	(Enable the virtual router)
`>> VRRP Virtual Router 1# vrid 1`	(Assign Virtual Router ID 1)
`>> VRRP Virtual Router 1# if 1`	(To Interface Number 1)
`>> VRRP Virtual Router 1# prio 101`	(Set the renter priority)
`>> VRRP Virtual Router 1# addr 30.0.0.50`	(Set IP address of virtual router)
`>> VRRP Virtual Router 1# share dis`	(Disable sharing)
`>> VRRP Virtual Router 1# track`	(Select *Virtual Router Tracking* menu)
`>> VRRP VR 1 Priority Tracking# vrs ena`	(Enable tracking of virtual routers)
`>> VRRP VR 1 Priority Tracking# apply`	(Apply the configuration)
`>> VRRP VR 1 Priority Tracking# save`	(Save the configuration)
`>> VRRP VR 1 Priority Tracking# /cfg/l3/vrrp/vr 2`	Select *Virtual Router 2* menu)
`>> VRRP Virtual Router 2# ena`	(Enable the virtual router)
`>> VRRP Virtual Router 2# vrid 2`	(Assign virtual router ID 2)
`>> VRRP Virtual Router 2# if 2`	(To Interface Number 2)
`>> VRRP Virtual Router 2# prio 101`	(Set the renter priority)
`>> VRRP Virtual Router 2# addr 20.0.0.1`	(Set IP address of virtual router)
`>> VRRP Virtual Router 2# share dis`	(Disable sharing)
`>> VRRP Virtual Router 2# track`	(Select the *Virtual Router Tracking* menu)
`>> VRRP VR 2 Priority Tracking# ports ena`	(Track VLAN ports)
`>> VRRP VR 2 Priority Tracking# apply`	(Apply the configuration)
`>> VRRP VR 2 Priority Tracking# save`	(Save the configuration)

7. Enable SLB on the clean Alteon CA.

```
>> # /cfg/slb/on
```

8. Configure real servers for health checking VPN devices.

`>> # /cfg/slb/real 1/ena`	(Enable SLB for Real Server 1)
`>> Real server 1 # rip 10.0.0.10`	(Assign IP address for Real Server 1)
`>> Real server 1 # /cfg/slb/real 2/ena`	(Enable SLB for Real Server 2)
`>> Real server 2 # rip 10.0.0.11`	(Assign IP address for Real Server 2)
`>> Real server 2 # /cfg/slb/real 3/ena`	(Enable SLB for Real Server 3)
`>> Real server 3 # rip 10.0.0.20`	(Assign IP address for Real Server 3)
`>> Real server 3 # /cfg/slb/real 4/ena`	(Enable SLB for Real Server 4)
`>> Real server 4 # rip 10.0.0.21`	(Assign IP address for Real Server 4)

9. Configure Real Server group 1, and add Real Servers 1, 2, 3, and 4 to the group.

`>> # /cfg/slb/group 1`	(Configure Real Server Group 1)
`>> Real server group 1# metric hash`	(Select SLB hash metric for Group 1)
`>> Real server group 1 # add 1`	(Add real servers 1 through 4 to Group 1)
`>> Real server 1# add 2/add3/add4`	

10. Configure a filter to enable the transparent load balancing (Return to Source MAC address) option. This adds an opposite entry in the session table so that the return traffic matches its source MAC address.

`>> # /cfg/slb/filt 20/adv`	(Select the Advanced menu for Filter 20)
`>> Filter 20 Advanced# rtsrcmac ena`	(Enable Return to Source MAC Address)

11. Enable filter processing on the server ports so that the responses from the real server are looked up in the VPN session table.

```
>> # /cfg/slb/port 1/filt ena
```

12. When dynamic routing protocols are not used, configure a gateway to the external router.

```
>> # /cfg/l3/gw 1/addr 192.168.10.50
```

13. Apply and save the configuration, and reboot Alteon.

```
>> # apply
>> # save
>> # /boot/reset
```

To configure the clean-side Alteon CB

1. Turn off BOOTP.

```
>> # /cfg/sys/bootp dis
```

2. Define and enable VLAN 2 for ports 25 and 26.

```
>> # /cfg/l2/vlan 2/ena/def 25 26
```

3. Turn off the Spanning Tree Protocol (STP).

```
>> # /cfg/l2/stg #/off
```

4. Define the clean-side IP interfaces. Create one clean-side IP interface on a different subnet for each VPN device being load balanced.

```
>> # /cfg/l3/if 1/ena/mask 255.255.255.0/addr 30.0.0.11
>> # /cfg/l3/if 2/ena/mask 255.255.255.0/addr 20.0.0.20/vl 2
>> # /cfg/l3/if 3/ena/mask 255.255.255.255/addr 20.0.0.21/vl 2
```

5. Configure routes for each of the IP interfaces you configured in step 4, using the VPN devices as gateways. One static route is required for each VPN device being load balanced.

```
>> #/cfg/l3/route>> # add 10.0.0.10 255.255.255.255 20.0.0.101 2
>> # add 10.0.0.11 255.255.255.255 20.0.0.102 3
>> # add 10.0.0.20 255.255.255.255 20.0.0.101 2
>> # add 10.0.0.21 255.255.255.255 20.0.0.102 3
```

6. Configure Virtual Router Redundancy Protocol (VRRP) for Virtual Routers 1 and 2.

```
>> # /cfg/l3/vrrp/on
>> Virtual Router Redundancy Protocol# vr
>> VRRP Virtual Router 1# ena
>> VRRP Virtual Router 1# vrid
>> VRRP Virtual Router 1# if
>> VRRP Virtual Router 1# addr 30.0.0.50
>> VRRP Virtual Router 1# share dis
>> VRRP Virtual Router 1 # track/vrs ena
>> VRRP Virtual Router 1 Priority Tracking# /cfg/l3/vrrp/vr 2
>> VRRP Virtual Router 2# ena
>> VRRP Virtual Router 2 # vrid 2
>> VRRP Virtual Router 2 # if 2
>> VRRP Virtual Router 2 # addr 20.0.0.1
>> VRRP Virtual Router 2 # share dis
>> VRRP Virtual Router 2 # track/ports ena
```

7. Enable SLB.

```
VRRP Virtual Router 2 Priority Tracking# /cfg/slb on
```

8. Configure real servers for health checking VPN devices.

```
>> Layer 4# /cfg/slb/real 1/ena/rip 10.0.010
>> Real server 1# /cfg/slb/real 2/ena/rip 10.0.0.11
>> Real server 2# /cfg/slb/real 3/ena/rip 10.0.0.20
>> Real server 3# /cfg/slb/real 4/ena/rip 10.0.0.21
```

9. Enable the real server group.

```
>> Real server 4 # /cfg/slb/group
>> Real server group 1# metric hash
>> Real server group 1# add 1/add 2/add 3/ add 4
```

10. Configure a filter to enable the transparent load balancing (Return to Source MAC address) option. This adds an opposite entry in the session table so that the return traffic matches its source MAC address.

`>> # /cfg/slb/filt 20/adv`	(Select the Advanced menu for Filter 20)
`>> Filter 20 Advanced# rtsrcmac ena`	(Enable Return to Source MAC Address)

11. Enable filter processing on the server ports so that the response from the real server will be looked up in VPN session table.

```
>> SLB port 25# /cfg/slb/port 1 /filt ena
```

12. Apply and save the configuration, and reboot Alteon.

```
>> SLB port 25# apply
>> SLB port 25# save
>> SLB port 25# /boot/reset
```

To configure the dirty-side Alteon DA

1. Turn off BOOTP.

```
>> # /cfg/sys/bootp dis
```

2. Define and enable VLAN 2 for ports 25 and 26.

```
>> # /cfg/l2/vlan 2/ena/def 25 26
```

3. Turn off the Spanning Tree Protocol (STP).

```
>> # /cfg/l2/stg/off
```

4. Configure IP interfaces 1, 2, and 3.

```
>> # /cfg/l3/if 1/ena/mask 255.255.255.0/addr 192.168.10.10
>> # /cfg/l3/if 2/ena/mask 255.255.255.0/addr 10.0.0.10/vl 2
>> # /cfg/l3/if 3/ena/mask 255.255.255.255/addr 10.0.0.11/vl 2
```

5. Define static routes for each of the IP interfaces you configured in step 4 using the VPN devices as gateways. One static route is required for each VPN device being load balanced.

```
>> # /cfg/l3/route
>> # add 20.0.0.10 255.255.255.255 10.0.0.101 2
>> # add 20.0.0.11 255.255.255.255 10.0.0.102 3
>> # add 20.0.0.20 255.255.255.255 10.0.0.101 2
>> # add 20.0.0.21 255.255.255.255 10.0.0.102 3
```

6. Configure VRRP for Virtual Routers 1 and 2.

```
>> # /cfg/l3/vrrp/on
>> Virtual Router Redundancy Protocol# /cfg/l3/vrrp/vr 1
>> VRRP Virtual Router 1# ena
>> VRRP Virtual Router 1# vrid 1
>> VRRP Virtual Router 1# if 1
>> VRRP Virtual Router 1# prio 101
>> VRRP Virtual Router 1# addr 192.168.10.50
>> VRRP Virtual Router 1# share dis
>> VRRP Virtual Router 1# track
>> VRRP Virtual Router 1 Priority Tracking# vrs ena
>> VRRP Virtual Router 1 Priority Tracking# ports ena
>> VRRP Virtual Router 1 Priority Tracking# /cfg/l3/vrrp/vr 2
>> VRRP Virtual Router 2# ena
>> VRRP Virtual Router 2# vrid 2
>> VRRP Virtual Router 2# if 2
>> VRRP Virtual Router 2# prio 101
>> VRRP Virtual Router 2# addr 10.0.0.1
>> VRRP Virtual Router 2# share dis
>> VRRP Virtual Router 2# track
>> VRRP Virtual Router 2 Priority Tracking# vrs ena
>> VRRP Virtual Router 2 Priority Tracking# ports>> # ena
```

7. Enable SLB.

```
>> VRRP Virtual Router 1 Priority Tracking# /cfg/slb/ on
```

8. Configure real servers for health-checking VPN devices.

```
>> Layer 4# real 1/ena/rip 20.0.0.10
>> Real server 1# /cfg/slb/real 2/ena/rip 20.0.0.11
>> Real server 2# /cfg/slb/real 3/ena/rip 20.0.0.20
>> Real server 3# /cfg/slb/real 4/ena/rip 20.0.0.21
```

9. Enable the real server group.

```
>> Real server 1# /cfg/slb/group 1
>> Real server group 1# metric hash
>> Real server group 1# add 1/add 2/add 3/add 4
```

10. Configure the filters to allow local subnet traffic on the dirty side of the VPN device to reach the VPN device interfaces.

```
>> # /cfg/slb/filt 100
>> # ena
>> # sip any
>> # dip 192.168.10.0/dmask 255.255.255.0
>> # action allow
>> # /cfg/slb/filt 110
>> # ena
>> # sip any
>> # dip 224.0.0.0/dmask 255.0.0.0
>> # action allow
```

11. Create the redirection filter and enable FWLB.

    This filter redirects inbound traffic, redirecting it among the defined real servers in the group.

```
>> # /cfg/slb/filt 2048
>> # ena>> # sip any
>> # dip any
>> # action redir
>> # /cfg/slb/filt 2048/adv/redir
>> # fwlb ena
```

12. Create a filter to allow the management firewall (policy server) to reach the VPN firewall.

```
>> # /cfg/slb/filt 120 ena
>> # sip 192.168.10.120
>> # smask 255.255.255.255
>> # dip 10.0.0.0
>> # dmask 255.255.255.0
```

13. Add filters to the ingress port.

```
>> # /cfg/slb/port 1
>> # filt ena
>> # add 100/add 110/add 2048
```

14. Apply and save the configuration, and reboot Alteon.

```
>> # apply
>> # save
>> # /boot/reset
```

To configure the dirty-side Alteon DB

1. Turn off BOOTP.

```
>> # /cfg/sys/bootp dis
```

2. Define and enable VLAN 2 for ports 25 and 26.

```
>> # /cfg/l2/vlan 2/ena/def 25 26
```

3. Turn off Spanning Tree Protocol (STP).

```
>> # /cfg/l2/stg/off
```

4. Configure IP interfaces 1, 2, and 3.

```
>> # /cfg/l3/if 1/ena/mask 255.255.255.0/addr 192.168.10.11
>> # /cfg/l3/if 2/ena/mask 255.255.255.0/addr 10.0.0.20/vl 2
>> # /cfg/l3/if 3/ena/mask 255.255.255.255/addr 10.0.0.21/vl 2
```

5. Configure routes for each of the IP interfaces you configured in step 4.

```
>> # /cfg/l3/route
>> # add 20.0.0.10 255.255.255.255 10.0.0.101 2
>> # add 20.0.0.11 255.255.255.255 10.0.0.102 3
>> # add 20.0.0.20 255.255.255.255 10.0.0.101 2
>> # add 20.0.0.21 255.255.255.255 10.0.0.102 3
```

6. Configure VRRP for Virtual Routers 1 and 2.

```
>> # /cfg/l3/vrrp/on
>> # /cfg/l3/vrrp/vr 1
>> # ena
>> # vrid 1
>> # if 1
>> # addr 192.168.10.50
>> # share dis
>> # track
>> # vrs ena
>> # ports ena
>> # /cfg/l3/vrrp/vr 2
>> # ena
>> # vrid 2
>> # if 2
>> # addr 10.0.0.1
>> # share dis
>> # track
>> # vrs ena
>> # ports ena
```

7. Enable SLB.

```
>> # /cfg/slb/on
```

8. Configure real servers for health checking VPN devices.

```
>> # /cfg/slb/real 1/ena/rip 20.0.0.10
>> # /cfg/slb/real 2/ena/rip 20.0.0.11
>> # /cfg/slb/real 3/ena/rip 20.0.0.20
>> # /cfg/slb/real 4/ena/rip 20.0.0.21
```

9. Enable the real server group, and place real servers 1 through 4 into the real server group.

```
>> # /cfg/slb/group 1
>> # metric hash
>> # add 1/add 2/add 3/add 4
```

10. Configure the filters to allow local subnet traffic on the dirty side of the VPN device to reach the VPN device interfaces.

```
>> # /cfg/slb/filt 100
>> # ena
>> # sip any
>> # dip 192.168.10.0/dmask 255.255.255.0
>> # /cfg/slb/filt 110
>> # ena
>> # sip any
>> # dip 224.0.0.0/dmask 255.0.0.0
```

11. Create the redirection filter and enable FWLB.

    This filter will redirect inbound traffic, among the defined real servers in the group.

```
>> # /cfg/slb/filt 2048
>> # ena
>> # sip any
>> # dip any
>> # proto any
>> # action redir
>> # /cfg/slb/filt 2048/adv/redir
>> # fwlb ena
```

12. Add filters to the ingress port.

```
>> # /cfg/slb/port 1
>> # filt ena
>> # add 100/add 110/add 2048
```

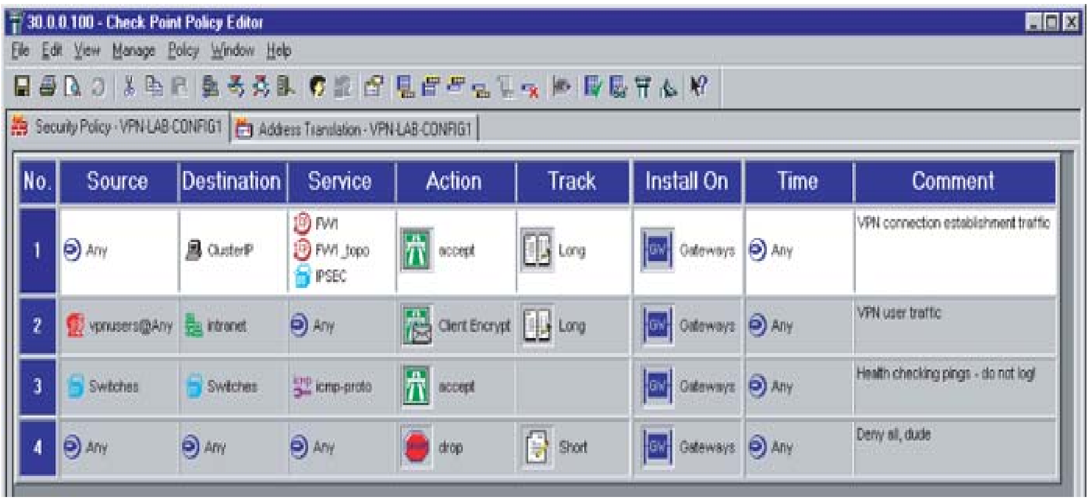
13. Apply and save the configuration and reboot Alteon.

```
>> # apply
>> # save
>> # /boot/reset
```

To test the configurations and general topology

Alteons should be able to perform health checks to each other and all devices should see four real servers.

Figure 118: Checkpoint Rules for both VPN Devices as seen in the Policy Editor



1. Disconnect the cables (cause failures) to change the available servers that are up

```
>> # /info/slb/dump
```

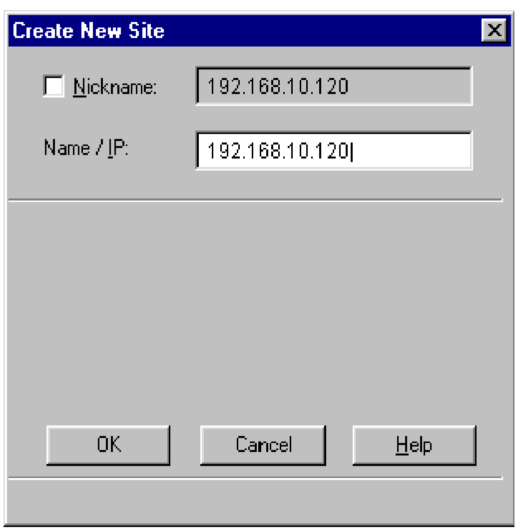
This should change the VRRP preferences. You can view VRRP preferences using the command **/info/l3/vrrp**.

2. Watch for accepted and dropped traffic. In the toolbar, go to **Window > Log Viewer**.

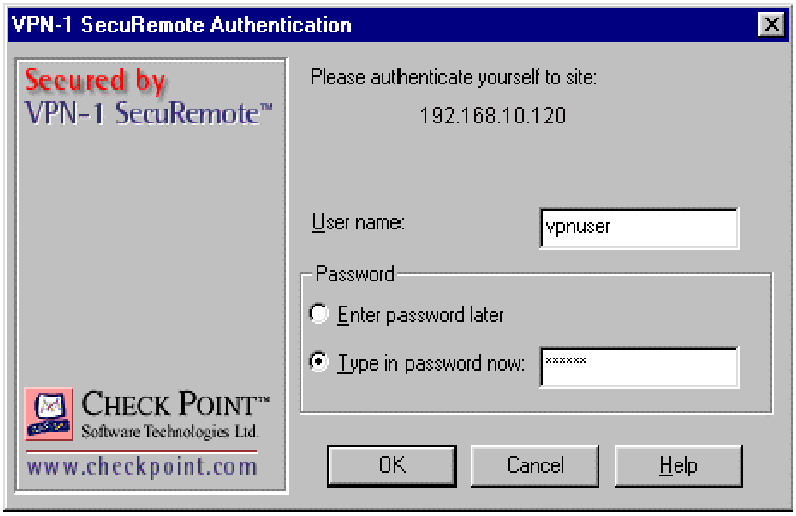
Note: To help simplify the logs, the health checks are *not* logged.

To test the VPN

1. Launch the SecuRemote client on the dirty side of the network.
2. Add a new site.



3. Enter the policy server IP address: 192.168.10.120. You have the option of adding a nickname.
4. Launch a browser (such as Netscape or Internet Explorer) and go to **http://30.0.0.100**.
5. Enter **vpnuser** for user name and **alteon** for the password.



A message displays verifying that you were authenticated.

6. Browse to the Web site.

   If there are other services running on other servers in the internal network, you should reach those services. All traffic traveling over the VPN is decrypted at the VPN device. You can verify which VPN device is being used by looking at the Log Viewer. You should also see the client authentication as well as the decrypted traffic.

7. To verify that the FWLB and hash metric is working correctly on the dirty-side Alteons (that is, hashed on client IP address/destination IP address), do one of the following:
   — Configure your current client with an IP address one higher (or lower) in the last octet, and try to re-establish the VPN connection.
   — Add another PC on the dirty side and connect to it.

Note: When many clients are coming from *behind* a VPN gateway (for example, not using the SecuRemote clients but using a VPN 1 Gateway or other compatible VPN Gateway), you do *not* see load balancing across those clients. Each SecuRemote client is treated differently, but each VPN 1 Gateway is treated as one client each (that is, one Client IP address). VPN Device 1 and VPN Device 2 belong to one cluster IP.

Chapter 25 – Global Server Load Balancing

This chapter provides information for configuring Global Server Load Balancing (GSLB) across multiple geographic sites. This chapter includes the following topics:

Using GSLB

To use GSLB, you must purchase an additional software license and license string. Contact Radware Technical Support to acquire additional software licenses. GSLB configurations running in earlier versions of the Alteon are maintained after upgrading. When you upgrade the software image to the new version, the configuration is migrated.

Once you have obtained the proper password key to enable GSLB, do the following:

1. Connect to the CLI via Telnet or the console port, and log in as the administrator, following the directions in the “Command Line Interface” chapter of the *Alteon Application Switch Operating System Command Reference.*
2. From the CLI, enter the `/oper/swkey` command.

   You are prompted to enter the license string. If it is correct for this MAC address, Alteon accepts the password, permanently records it in non-volatile RAM (NVRAM), and then enables the feature.

Distributed Site Session Protocol (DSSP)

Distributed Site Selection Protocol (DSSP) is a Radware proprietary protocol for supporting Alteon GSLB functionality which resides above TCP. It enables Alteons in various sites to communicate and exchange required GSLB data and statuses. Availability is determined by regular health checks to determine the status of a remote real server. It enables the sending and receiving of remote site updates. DSSP supports server response time and sessions available in the remote site updates.

DSSP Versions

By default, DSSP version 1 is enabled. Alteon supports the following DSSP versions:

- DSSP version 1—The initial release of DSSP.
- DSSP version 2—DSSP version 2 adds support for server response time, CPU use, session availability, and session utilization in the remote site updates.
- DSSP version 3—DSSP version 3 adds support for the availability persistence selection metric.
- DSSP version 4—DSSP version 4 adds support for the client proximity selection metric in remote site updates.
- DSSP version 5—DSSP version 5 adds support for IPv6 remote servers updates.

For details on the DSSP site selection metrics, see GSLB Metrics, page 708.

Support for DSSP Versions

Although all versions of DSSP are supported, if you require interconnection to Alteons running earlier software versions, use the DSSP version that best accommodates those earlier software versions.

If interconnection to Alteons running older software versions is not required, use the most recent version supported by all Alteons.

To change the DSSP version

```
>> /cfg/slb/gslb/version <1-5>.
```

GSLB Overview

GSLB enables balancing server traffic load across multiple physical sites. The Alteon GSLB implementation takes into account an individual site's health, response time, and geographic location to smoothly integrate the resources of the dispersed server sites for complete global performance.

Benefits

GSLB meets the following demands for distributed network services:

- High content availability is achieved through distributed content and distributed decision-making. If one site becomes disabled, the others become aware of it and take up the load.
- There is no latency during client connection set-up. Instant site hand-off decisions can be made by any distributed Alteon.
- The best performing sites receive a majority of traffic over a given period of time but are not overwhelmed.
- Alteons at different sites regularly exchange information through the Distributed Site State Protocol (DSSP), and can trigger exchanges when any site's health status changes. This ensures that each active site has valid state knowledge and statistics. All versions of DSSP are supported.
- GSLB implementation takes geography as well as network topology into account.
- Creative control is given to the network administrator or Webmaster to build and control content by user, location, target application, and more.

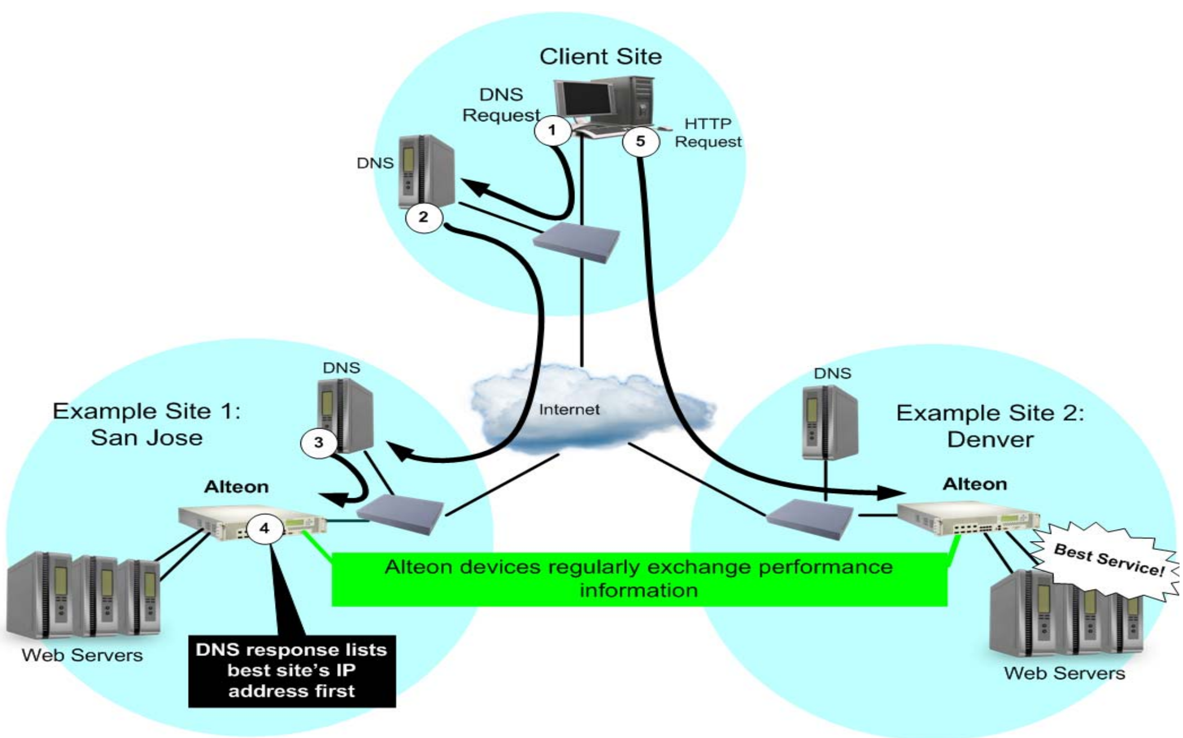
- GSLB is easy to deploy, manage, and scale. Alteon configuration is straightforward. There are no complex system topologies involving routers, protocols, and so on.
- Flexible design options are provided.
- All IP protocols are supported.
- Supports IPv4, IPv6, and mixed IP version environments.

How GSLB Works

A GSLB device performs or initiates a global server selection to direct client traffic to the best server for a given domain during the initial client connection.

GSLB is based on the Domain Name System (DNS) and proximity by source IP address. In the example in Figure 119 - DNS Resolution with GSLB, page 707, a client is using a Web browser to view the Web site for the Example Corporation at “www.example.com”. The Example Corporation has two Web sites: one in San Jose and one in Denver, each with identical content and available services. Both Web sites have an Alteon configured for GSLB, with domain name set to “www.gslb.example.com.” These devices are also configured as the Authoritative Name Servers for “www.example.com.” On the company master DNS server, the configuration is to delegate “www.example.com” to “www.gslb.example.com”.

Figure 119: DNS Resolution with GSLB



The DNS resolution for this GSLB configuration is as follows:

1. The client Web browser requests the "www.example.com" IP address from the local DNS.
2. The client's DNS asks its upstream DNS, which in turn asks the next, and so on, until the address is resolved.

   Eventually, the request reaches an upstream DNS server that has the IP address information available or the request reaches one of the Example, Inc. DNS servers.

3. The Example Inc.'s San Jose DNS tells the local DNS to query the Alteon with GSLB software as the authoritative name server for "www.example.com."
4. The San Jose Alteon responds to the DNS request, listing the IP address with the current best service.

   Each Alteon with GSLB software is capable of responding to the client's name resolution request. Since each Alteon regularly checks and communicates health and performance information with its peers, either Alteon can determine which sites are best able to serve the client's Web access needs. It can respond with a list of IP addresses for the Example Inc.'s distributed sites, which are prioritized by performance, geography, and other criteria.

   In this case, the San Jose Alteon knows that Example Inc. Denver currently provides better service, and lists Example Inc. Denver's virtual server IP address first when responding to the DNS request.
5. The client connects to Example Inc. Denver for the best service.

   The client's Web browser uses the IP address information obtained from the DNS request to open a connection to the best available site. The IP addresses represent virtual servers at any site, which are locally load balanced according to regular SLB configuration.

   If the site serving the client HTTP content suddenly experiences a failure (no healthy real servers) or becomes overloaded with traffic (all real servers reach their maximum connection limit), Alteon issues an HTTP redirect and transparently causes the client to connect to another peer site.

   The end result is that the client gets quick, reliable service with no latency and no special client-side configuration.

GSLB Metrics

This section describes all GSLB metrics as governed by DSSP. All metrics can be prioritized for selection order, and can be weighted on a per site basis. This section includes the following sub-sections:

- Metric Preferences, page 710
- Rules, page 710
- GSLB Availability Persistence, page 710
- GSLB Client Proximity Metric, page 711

For details on the configuration parameters of GSLB metrics, see the **`/cfg/slb/gslb/rule`** menu and the command descriptions in the *Alteon Application Switch Operating System Command Reference*.

The following is a list of all GSLB metrics:

- **Session utilization capacity threshold**—Causes the GSLB-enabled Alteon to not select the server when the session utilization of the server goes above the threshold. The session utilization is the percentage of sessions used over total sessions that results in normalized sessions between servers. When the server is not available, the session utilization is 100%. This is a threshold metric and it overwrites all other metrics. This metric requires remote site updates.
- **CPU utilization capacity threshold—Causes** the GSLB-enabled Alteon to not select the server when the CPU utilization of the site with the server goes above the threshold. CPU utilization is the highest CPU utilization for periods of up to 64 seconds among SPs. This is a threshold metric and overwrites all other metrics.This metric requires remote site updates.
- **Session available capacity threshold**—Does not select the server when the number of available sessions on the server falls below the threshold. When the server is not available, the session available is 0. This is a threshold metric and it overwrites all other metrics. This metric requires remote site updates.

- **Geographical preference**—Causes the GSLB-enabled Alteon to select the server based on the same IANA region of the source IP address and the server IP address. This metric does not require remote site updates.

  You can use the **/info/slb/gslb/geo** command to obtain a list of the IP address ranges that are mapped to each region. The regions are as follows:

  — North America

  — South America

  — Europe

  — Caribbean

  — Pacific Rim

  — Sub-Sahara

  — Japan

- **Network preference**—Selects the server based on the preferred network of the source IP address. This metric does not require remote site updates.
- **Weighted least connections**—Selects the server based on which server has the lowest session utilization. Session utilization is the percentage of sessions used over total sessions, which results in normalized sessions between servers. A server whose session utilization is 100% is considered unavailable. This metric requires remote site updates.
- **Weighted response time**—Selects the server based on the lowest response time in milliseconds from an SLB health check of the servers. This metric requires SLB health checks and remote site updates.
- **Weighted round-robin**—Selects the server based on round-robin of the servers.
- **Weighted random**—Selects the server based on uniform random distribution of the servers.
- **Availability**—Selects the same server while the server is still available. If the same server is not available, this metric selects the next server based on a ranking of the local virtual server and remote real server in a list from the highest (48) to the lowest (1) ranking. Multiple servers can have the same priority. This metric allows servers to be grouped based on priorities, or into primary and secondary groups. This metric requires SLB health checks and remote site updates. This is discussed in further detail in GSLB Availability Persistence, page 710.
- **Quality of service**—Selects the server based on combination of the lowest session utilization and the lowest response time of the SLB health check of the servers. This metric requires SLB health checks and remote site updates.
- **Minmisses**—Selects the same server based on the hash of source IP address and domain name. The hash calculation uses all the servers that are configured for the domain irrespective of the state of the server. When the server selected is not available, minmisses selects the next available server.
- **Hashing**—Selects the same server based on the hash of source IP address and domain name. The hash calculation uses only the servers that are available for the domain. The server selected may be affected when a server becomes available or not available, since the hash calculation uses only the servers that are available.
- **DNS local**—Selects the local virtual server only when the local virtual server is available. This metric applies to DNS-based GSLB only.
- **DNS always**—Selects the local virtual server even though the local virtual server is not available. This metric applies to DNS-based GSLB only.
- **Remote**—Selects the remote real servers only.
- **Persistence**—Selects the server for which the persistency cache contains the client IP address and subnet mask.

Metric Preferences

Setting metric preferences enables the GSLB selection mechanism to use multiple metrics from a metric preference list. GSLB selection starts with the first metric in the list. It then goes to the next metric when no server is selected, or when more than the required servers is selected. The GSLB selection stops when the metric results in at least one and no more than the required servers, or after the last metric in the list is reached. For DNS direct-based GSLB, the DNS response can be configured to return up to 10 required servers. For HTTP redirects based GSLB, the required server is one.

The following metrics can be selected from the metric preference list:

- Geographical preference
- Network preference
- Least connections
- Response time
- Round-robin
- Random
- Availability
- Quality of service
- Minmisses
- Hashing
- DNS local
- DNS always
- Remote
- Persistence

Rules

A rule lets the GSLB selection mechanism use a different GSLB site selection metric preference list, and rules can be set based for the time of day. Each domain has one or more rules. Each rule has a metric preference list. The GSLB selection selects the first rule that matches the domain and starts with the first metric in the metric preference list of the rule. For more information on rules, see Configuring GSLB with Rules, page 730.

GSLB Availability Persistence

The GSLB *availability* metric is used in GSLB rules to select a server exclusively when that server is available. Should that server become unavailable, the next available server in a list is selected to service requests. Availability is determined by a rank assigned to each server ranging from the lowest score of 1 to the highest score of 48. Multiple servers can be scored the same.

Rules that use availability as the primary metric handle failures by selecting the server with the next highest score compared to that of the server that failed, and begins forwarding requests to that server. Should the server that failed become operational again, that server regains precedence and requests are routed to it once more.

GSLB availability persistence allows the administrator to change the behavior of the availability metric to reassign requests to a server that had previously failed because of its higher initial score. With availability persistence enabled, a server that takes over after a failure is assigned the highest possible availability value (48). This ensures that after the server that failed becomes operational again, it cannot regain precedence from the recovery server. Should this new primary server fail, its original availability value is restored and the next server in the list gains the high precedence.

To enable GSLB availability persistence

1. DSSP version 3 must be enabled on all Alteons with GSLB configured, using the following command:

```
/cfg/slb/gslb/version 3
```

2. The availability metric must be the first metric configured in the first GSLB rule. For information on rule creation, see Rules, page 710.
3. Enable availability persistence on the backup Alteon (the Alteon that will take over from the primary Alteon) using the following command:

```
/oper/slb/gslb/avpersis <virtual server number> enable
```

Note: This is an operational command that *does not* survive an Alteon reboot.

4. After the primary server recovers, you can revert to the configured availabilities on the Alteon whose virtual server currently has precedence. This is the Alteon with the virtual server that is advertising an availability of 48:

```
/oper/slb/gslb/avpersis <virtual server number> disable.
```

After both sites are reporting their configured availability, turn the feature back on by enabling availability persistence on Alteon with the backup server:

```
/oper/slb/gslb/avpersis <virtual server number> enable
```

You can use the following command to enable or disable availability persistence on the backup Alteon:

```
/cfg/slb/virt <virtual server number>/avpersis enable/disable.
```

GSLB Client Proximity Metric

The GSLB client proximity metric measures the proximity between each data center and the client. It is limited to HTTP and HTTPS traffic. This section describes the basic commands used with client proximity.

For more information about the client proximity commands, see the *Alteon Application Switch Operating System Command Reference*. For example configurations using GSLB client proximity, see Configuring GSLB with Client Proximity, page 734.

To enable client proximity for HTTP/HTTPS

```
/cfg/slb/virt <x>/service <x>/http/clientprox http|https|none
```

To set client proximity parameters

```
/cfg/slb/gslb/clntprox
```

To view the client proximity statistics

```
/stats/slb/gslb/clntprox
```

GSLB Persistence Metric

When Alteon receives a GSLB client request that includes a rule with the *persistence* metric, it searches the relevant server persistency cache for the client IP address and subnet mask.

If Alteon finds the client IP address and mask, it executes the rule. If Alteon does not find the client IP address and mask, it returns a saved GSLB load-balancing decision from the persistence table and stops the process.

To enable GSLB persistence

```
/cfg/slb/gslb/rule/metric/gmetric/persistence
```

GSLB and DNSSEC

The Domain Name System Security Extensions (DNSSEC) adds authentication security measurements to Alteon to defend the DNS protocol against known DNS threats. DNS digitally signs records for DNS lookup using public-key cryptography. The correct DNSKEY record is authenticated using a chain of trust, starting with a set of verified public keys for the DNS root zone, which is the trusted third party. When DNSSEC is used, each answer to a DNS lookup contains an RRSIG DNS record in addition to the requested record type. The RRSIG record is a digital signature of the DNS resource record set answer. The digital signature can be verified by locating the correct public key found in a DNSKEY record. The DNS record is used in the authentication of DNSKEYs in the lookup procedure using the chain of trust.

To enable the use of replacement keys, a key rollover procedure is used. New keys are rolled out in new DNSKEY records in addition to the existing old keys.

For authentication purposes, Alteon uses two different keys in DNSKEY records, with different DNSKEY records for each. Key Signing Keys (KSKs) are used to sign the Zone Signing Key (ZSKs) and are exported (publicly) to the parent DNS. ZSKs are used to sign the DNS resource records (RRs). Because the ZSKs are controlled and used by one specific DNS zone, they can be switched more easily and more frequently. RFC 4614 recommends changing ZSKs on a monthly basis, enabling them to be shorter in bit length (for example, 1024). The KSK validity period is usually one year, and needs a higher bit length (for example, 2048), making it harder to forge. When a new KSK is created, the delegation signer (DS) record must be transferred to the parent zone, and must be signed and published there.

When working with GSLB and DNSSEC enabled, the configuration of remote sites must be identical for all Alteons participating in the GSLB configuration (**`/cfg/slb/gslb/site x`**).

For GSLB sites to synchronize Alteon peers, the passphrase for Alteon synchronization must be enabled (**`/cfg/slb/sync/passphrs`**). Failing to set the passphrase generates an error message.

For more information on configuring GSLB with DNSSEC, see Configuring GSLB with DNSSEC, page 743.

Configuring Basic GSLB

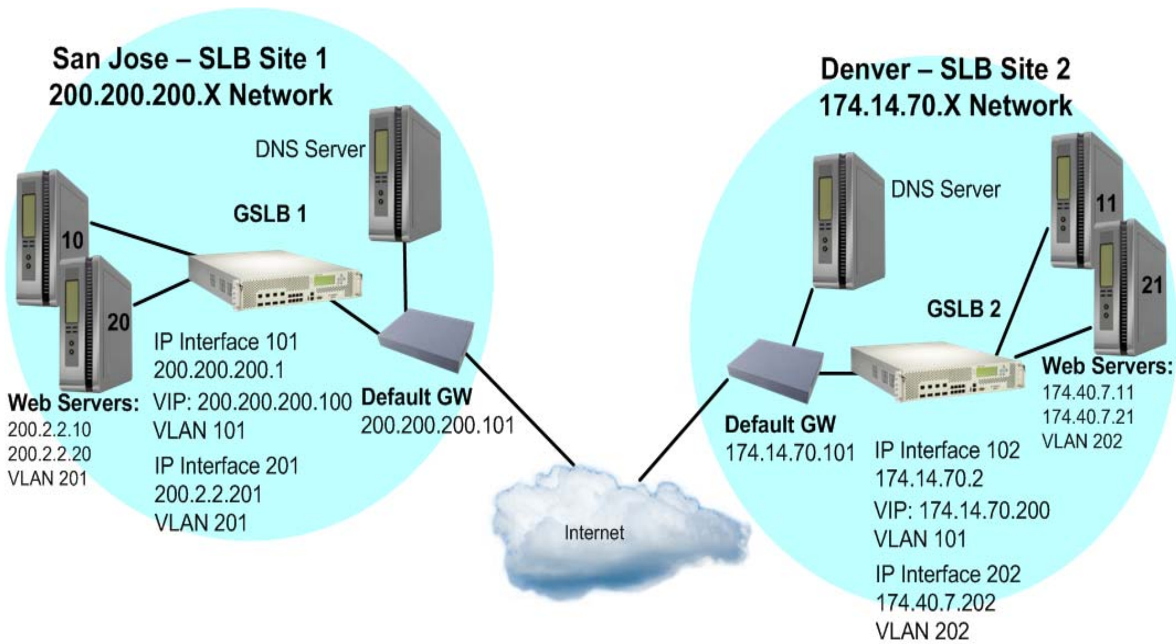
The basic GSLB configuration procedure is an extension of the standard configuration procedure for SLB, as follows:

1. Use the administrator login to connect to the Alteon you want to configure.
2. Activate the GSLB software key. For details, see the *Alteon Application Switch Operating System Command Reference* for details.
3. Configure Alteon at each site with basic attributes:
   - Configure the Alteon IP interface.
   - Configure the default gateways.
4. Configure Alteon at each site to act as the DNS server for each service that is hosted on its virtual servers. Also, configure the master DNS server to recognize Alteon as the authoritative DNS server for the hosted services.
5. Configure Alteon at each site for local SLB:
   - Define each local real server.
   - Group local real servers into real server groups.
   - Define the local virtual server with its IP address, services, and real server groups.
   - Define the port states.
   - Enable SLB.
6. Configure each Alteon so that they recognize their remote peers.
   - Configure a remote real server entry on each Alteon for each remote service. This is the virtual server IP address that is configured on the remote peer.
   - Add the remote real server entry to an appropriate real server group.
   - Enable GSLB.

Example GSLB Topology

The procedures to implement the example GSLB topology illustrated in Figure 120 - GSLB Topology Example, page 714 are described in this example.

Figure 120: GSLB Topology Example



Notes

- In the procedures described in this example, many of the options are left at their default values. For more details about these options, see Implementing Server Load Balancing, page 167.
- For details about any of the processes or menu commands described in this example, see the *Alteon Application Switch Operating System Command Reference*.

To configure the basics at the San Jose Site

1. Optionally on the San Jose Alteon, configure management access and the management gateway address, and then enable the management port.

Command	Description
`>> # /cfg/sys/mmgmt/addr 50.133.88.31`	(Management port IP address)
`>> Management Port# mask 255.255.255.0`	(Management port mask)
`>> Management Port# gw 50.133.88.1`	(Management port gateway address)
`>> Management Port# ena`	(Enable the management port)

2. If you are using the BBI for managing the San Jose Alteon, change its service port.

   By default, GSLB listens on service port 80 for HTTP redirection. By default, the BBI also uses port 80. Both services cannot use the same port. If the BBI is enabled, configure it to use a different port.

   Note: Use the **`/cfg/sys/access/http`** command to enable BBI.

For example, enter the following command to change the BBI port to 8080:

```
>> # /cfg/sys/access/wport 8080
```

3. Configure a VLAN for the Internet traffic.

```
>> # /cfg/l2/vlan 101/name internet              (VLAN 101 for Internet)
>> VLAN 101# add 2/ena                           (Add Port 2 to VLAN 101)

Port 2 is an UNTAGGED port and its current PVID is 1.
Confirm changing PVID from 1 to 101 [y/n]: y
Current ports for VLAN 101:      empty
Pending new ports for VLAN 101:  2
Current status: disabled
New status:     enabled
```

4. Configure another VLAN for local server traffic, and add server ports to this VLAN.

```
>> # /cfg/l2/vlan 201/name servers               (VLAN 201 for local servers)
>> VLAN 201# add 4/ena                           (Add Port 4 to VLAN 201)

Port 4 is an UNTAGGED port and its current PVID is 1.
Confirm changing PVID from 1 to 201 [y/n]: y
Current ports for VLAN 201:      empty
Pending new ports for VLAN 201:  10
Current status: disabled
New status:     enabled

>> VLAN 201# add 3/ena                           (Add Port 3 to VLAN 201)

Port 3 is an UNTAGGED port and its current PVID is 1.
Confirm changing PVID from 1 to 201 [y/n]: y
Current ports for VLAN 201:      empty
Pending new ports for VLAN 201:  3 4
```

5. Define an IP interface to the local real servers.

```
>> # /cfg/l3/if 201                              (Select IP Interface 201)
>> IP Interface 201# addr 200.2.2.201            (Assign IP address for the interface)
>> IP Interface 201# mask 255.255.255.0          (Assign network mask)
>> IP Interface 201# vlan 201                    (Assign interface to VLAN 201)
>> IP Interface 201# ena                         (Enable IP interface 201)
```

6. Define an IP interface to the Internet. The IP interface responds when asked to resolve client DNS requests.

```
>> # /cfg/l3/if 101                              (Select IP Interface 101)
>> IP Interface 101# addr 200.200.200.1          (Assign IP address for the interface)
>> IP Interface 101# mask 255.255.255.0          (Assign network mask)
>> IP Interface 101# vlan 101                    (Assign interface to VLAN 101)
>> IP Interface 101# ena                         (Enable IP Interface 101)
```

7. Define the default gateway. The router at the edge of the site acts as the default gateway to the Internet. The default gateway address should be on the same subnet as the IP interface 101, which points to the Internet. To configure the default gateway for this example, enter the following commands:

`>> IP Interface 101# /cfg/l3/gw 1`	(Select Default Gateway 1)
`>> Default gateway 1# addr 200.200.200.101`	(Assign IP address for the gateway)
`>> Default gateway 1# ena`	(Enable Default Gateway 1)

8. Apply and save the configuration.

```
>> # apply
>> # save
```

9. Configure the master DNS server to recognize the local GSLB Alteon as the authoritative name server for the hosted services.

   Determine the domain name that will be distributed to both sites and the hostname for each distributed service. In this example, the San Jose DNS server is configured to recognize 200.200.200.1 (the IP interface of the San Jose GSLB Alteon) as the authoritative name server for "www.gslb.example.com."

To configure the San Jose Alteon for standard SLB

1. Assign an IP address to each of the real servers in the local San Jose server pool.

   The real servers in any real server group must have an IP route to Alteon that will perform the SLB functions. This is most easily accomplished by placing Alteons and servers on the same IP subnet, although advanced routing techniques can be used as long as they do not violate the topology rules.

   For this example, the host real servers have IP addresses on the same IP subnet.

2. Define each local real server. For each local real server, you must assign a real server number, specify its actual IP address, and enable the real server. For example:

`>> Default gateway 1# /cfg/slb/real 10`	(Configure Real Server 10)
`>> Real server 10# rip 200.2.2.10`	(Assign IP address to Server 10)
`>> Real server 10# ena`	(Enable Real Server 10)
`>> Real server 10# /cfg/slb/real 20`	(Configure Real Server 20)
`>> Real server 20# rip 200.2.2.20`	(Assign IP address to Server 20)
`>> Real server 20# ena`	(Enable Real Server 20)

3. On the San Jose Alteon, define a real server group.

   Combine the real servers into one service group and set the necessary health checking parameters. In this example, HTTP health checking is used to ensure that Web content is being served. If the index.html file is not accessible on a real server during health checks, the real server will be marked as down.

`>> Real server 2# /cfg/slb/group 1`	(Select Real Server Group 1)
`>> Real server group 1# add 10`	(Add Real Server 10 to Group 1)
`>> Real server group 1# add 20`	(Add Real Server 20 to Group 1)

`>> Real server group 1# health http`	(Use HTTP for health checks)
`>> Real server group 1# content index.html`	(Set URL content for health checks)

4. On the San Jose Alteon, define a virtual server.

   All client requests are addressed to a virtual server IP address defined on Alteon. Clients acquire the virtual server IP address through normal DNS resolution. HTTP uses well-known TCP port 80. In this example, HTTP is configured as the only service running on this virtual server IP address and, is associated with the real server group. For example:

`>> Real server group 1# /cfg/slb/virt 1`	(Select Virtual Server 1)
`>> Virtual server 1# vip 200.200.200.100`	(Assign a virtual server IP address)
`>> Virtual Server 1# service 80`	
`>> Virtual server 1 http Service# group 1`	(Associate virtual port to real group)
`>> Virtual server 1 http Service# /cfg/slb/virt 1 ena`	(Enable virtual server)

   This configuration is not limited to HTTP services. For a list of other well-known TCP/IP services and ports, see Well-Known Application Ports, page 175.

5. On the San Jose Alteon, define the type of Layer 4 traffic processing each port must support. The ports are configured as follows:

`>> Virtual server 1# /cfg/slb/port 4`	(Select physical Port 4)
`>> SLB port 4# server ena`	(Enable server processing on Port 4)
`>> SLB port 4# /cfg/slb/port 3`	(Select physical Port 3)
`>> SLB port 3# server ena`	(Enable server processing on Port 3)
`>> SLB port 3# /cfg/slb/port 2`	(Select physical Port 2)
`>> SLB port 2# client ena`	(Enable client processing on Port 2)

6. On the San Jose Alteon, enable SLB.

```
>> SLB port 6# /cfg/slb/on
```

To configure the San Jose Site for GSLB

1. On the San Jose Alteon, turn on GSLB.

```
>> Virtual server 1# /cfg/slb/gslb/on
```

2. Enable DSSP version 2 to send out remote site updates.

   Note: Unless you are in the middle of network migration from an Alteon version prior to 22.0, you should always enable DSSP version 2 or later.

```
>> # /cfg/slb/gslb/version 2
```

3. On the San Jose Alteon, define each remote site.

   When you start configuring at the San Jose site, San Jose is local and Denver is remote. Add and enable the remote Alteon's Internet-facing IP interface address. In this example, there is only one remote site: Denver, with an IP interface address of 74.14.70.2. Use the following commands:

`>> # /cfg/slb/gslb/site 2`	(Select Remote Site 2)
`>> Remote site 2# name site_2`	(Name Remote Site 2)
`>> Remote site 2# prima 174.14.70.2`	(Define remote interface)
`>> Remote site 2# ena`	(Enable Remote Site 1)

   Each additional remote site should be configured in the same manner. You can enable up to 64 remote sites.

4. On the San Jose Alteon, assign each remote distributed service to a local virtual server.

   Configure the local San Jose site to recognize the services offered at the remote Denver site. To do this, configure one real server entry on the San Jose Alteon for each virtual server located at each remote site. Since there is only one remote site (Denver) with only one virtual server, only one more local real server entry is needed at the San Jose site.

   The new real server entry is configured with the remote virtual server IP address (that is, Alteon 2's VIP address, rather than the usual IP address of a local physical server). Do not confuse this value with the IP interface address on the remote Alteon.

   The `remote` parameter is enabled, and the real server entry is added to the real server group under the local virtual server for the intended service. Finally, since the real server health checks are performed across the Internet, the health-checking interval should be increased to 30 or 60 seconds to avoid generating excess traffic. The health check interval should also depend on the number of remote sites. The more remote sites you have, the larger the time interval should be. For example:

`>> # /cfg/slb/real 2`	(Create an entry for Real Server 2)
`>> Real server 2# rip 174.14.70.200`	(Set remote virtual server IP address)
`>> Real server 3# adv/remote enable`	(Define the real server as remote)
`>> Real server 3# inter 30`	(Set a higher health check interval)
`>> Real server 3# ena`	(Enable the real server entry)
`>> Real server 3# /cfg/slb/group 1`	(Select appropriate real server group)
`>> Real server group 1# add 2`	(Add Real Server 2 to the Group 1)

   Note: You should note where each configured value originates, or this step can result in improper configuration.

5. On the San Jose Alteon, define the domain name and hostname for each service hosted on each virtual server.

   In this example, the domain name for Example Inc. is "gslb.example.com," and the hostname for the only service (HTTP) is "www." These values are configured as follows:

`>> Real server group 1# /cfg/slb/virt 1`	(Select Virtual Server 1)
`>> Virtual server 1# dname gslb.example.com`	(Define domain name)
`>> Virtual server 1# service 80/hname www`	(Define HTTP hostname)

To define other services (such as FTP), make additional hostname entries.

6. Apply and verify the configuration.

```
>> Global SLB# apply
>> Global SLB# cur
```

7. Examine the resulting information. If any settings are incorrect, make and apply any appropriate changes, and then check again.

```
>> Global SLB# /cfg/slb/cur
```

8. Save your new configuration changes.

```
>> Layer 4# save
```

To configure the basics at the Denver Site

Following the same procedure described for San Jose (see To configure the basics at the San Jose Site, page 714), configure the Denver site as follows:

1. Optionally, on the Denver Alteon, configure management access and the management gateway address.

`>> # /cfg/sys/mmgmt/addr 49.133.88.31`	(Management port IP address)
`>> Management Port# mask 255.255.255.0`	(Management. port mask)
`>> Management Port# gw 49.133.88.1`	(Management. port gateway address)
`>> Management Port# ena`	(Enable the management port)

2. If you are using the BBI for managing the San Jose Alteon, change its service port.

   By default, GSLB listens on service port 80 for HTTP redirection. By default, the BBI also uses port 80. Both services cannot use the same port. If the BBI is enabled, configure it to use a different port.

   Note: Use the **`/cfg/sys/access/http`** command to enable BBI.

   For example, enter the following command to change the BBI port to 8080:

```
>> # /cfg/sys/wport 8080
```

3. Configure a VLAN for the Internet traffic.

```
>> # /cfg/l2/vlan 102/name internet              (VLAN 102 for Internet)
>> VLAN 102# add 2/ena                           (Add port 2 to VLAN 102 and enable)

Port 2 is an UNTAGGED port and its current PVID is 1.
Confirm changing PVID from 1 to 102 [y/n]: y
Current ports for VLAN 102:      empty
Pending new ports for VLAN 102:  2
Current status: disabled
New status:     enabled
```

4. Configure a VLAN for local server traffic, and add server ports to this VLAN.

```
>> # /cfg/l2/vlan 202/name servers               (VLAN 202 for local servers)
>> VLAN 202# add 11/ena                          (Add Port 11 to VLAN 202)

Port 10 is an UNTAGGED port and its current PVID is 1.
Confirm changing PVID from 1 to 202 [y/n]: y
Current ports for VLAN 202:      empty
Pending new ports for VLAN 202:  11
Current status: disabled
New status:     enabled

>> VLAN 202# add 12/ena                          (Add Port 12 to VLAN 201)

Port 11 is an UNTAGGED port and its current PVID is 1.
Confirm changing PVID from 1 to 202 [y/n]: y
Current ports for VLAN 202:      empty
Pending new ports for VLAN 202:  11 12
```

5. Define an IP interface to the local real servers.

```
>> # /cfg/l3/if 202                              (Select IP Interface 202)
>> IP Interface 202# addr 174.40.7.202           (Assign IP address for the interface)
>> IP Interface 202# mask 255.255.255.0          (Assign network mask)
>> IP Interface 202# vlan 202                    (Assign interface to VLAN 202)
>> IP Interface 202# ena                         (Enable IP Interface 202)
```

6. Define an IP interface to the Internet.

```
>> # /cfg/l3/if 102                              (Select IP Interface 102)
>> IP Interface 102# addr 174.14.70.2            (Assign IP address for the
                                                 interface)
>> IP Interface 102# mask 255.255.255.0          (Assign network mask)
>> IP Interface 102# vlan 102                    (Assign interface to VLAN 102)
>> IP Interface 102# ena                         (Enable IP Interface 102)
```

7. Define the default gateway.

```
>> IP Interface 102# /cfg/l3/gw 1                (Select Default Gateway 1)
>> Default gateway 1# addr 174.14.70.101         (Assign IP address for the
                                                 gateway)
>> Default gateway 1# ena                        (Enable Default Gateway 1)
```

8. Apply and save the configuration.

```
>> # apply
>> # save
```

9. Configure the local DNS server to recognize the local GSLB Alteon as the authoritative name server for the hosted services.

   Determine the domain name that will be distributed to both sites and the hostname for each distributed service. In this example, the Denver DNS server is configured to recognize 174.14.70.2 (the IP interface of the Denver GSLB Alteon, configured with the domain name "www.gslb.example.com") as the authoritative name server for "www.example.com."

To configure the Denver Alteon for Standard SLB

1. Assign an IP address to each of the real servers in the local Denver server pool, as defined in Table 61:

Table 61: Denver Real Server IP Addresses

Real Server	IP address
Server 11	174.14.7.11
Server 21	174.14.7.21

2. On the Denver Alteon, define each local real server.

```
>> Default gateway 1# /cfg/slb/real 11      (Server C is Real Server 1)
>> Real server 11# rip 174.14.7.11          (Assign IP address for Server 11)
>> Real server 11# ena                      (Enable Real Server 11)
>> Real server 11# /cfg/slb/real 21         (Server D is Real Server 21)
>> Real server 21# rip 174.14.7.21          (Assign IP address for Server 21)
>> Real server 21# ena                      (Enable Real Server 21)
```

3. On the Denver Alteon, define a real server group.

```
>> Real server 2# /cfg/slb/group 1          (Select Real Server Group 1)
>> Real server group 1# add 11              (Add Real Server 11 to Group 1)
>> Real server group 1# add 21              (Add Real Server 21 to Group 1)
>> Real server group 1# health http         (Use HTTP for health checks)
>> Real server group 1# content index.html  (Set URL content for health checks)
```

4. On the Denver Alteon, define a virtual server.

```
>> Real server group 1# /cfg/slb/virt 1     (Select Virtual Server 1)
>> Virtual server 1# vip 174.14.70.200      (Assign IP address)
>> Virtual server 1# service http           (Select the HTTP Service menu)
```

`>> Virtual server 1 http service# group 1`	(Associate virtual port to real group)
`>> Virtual server 1 http service# /cfg/slb/virt 1/ena`	(Enable the virtual server)

5. On the Denver Alteon, define the type of Layer 4 processing each port must support, as follows:

`>> # /cfg/slb/port 11`	(Select physical Port 11)
`>> SLB port 11# server ena`	(Enable server processing on Port 11)
`>> SLB port 11# /cfg/slb/port 12`	(Select physical Port 12)
`>> SLB port 12# server ena`	(Enable server processing on Port 12)
`>> SLB port 12# /cfg/slb/port 2`	(Select physical Port 2)
`>> SLB port 2# client ena`	(Enable client processing on Port 2)

6. On the Denver Alteon, enable SLB.

```
>> # /cfg/slb/on
```

To configure the Denver Site for GSLB

Following the same procedure described for San Jose (see To configure the San Jose Site for GSLB, page 717), configure the Denver site as follows:

1. On the Denver Alteon, turn on GSLB.

```
>> Virtual server 1# /cfg/slb/gslb/on
```

2. Enable DSSP version 2 to send out remote site updates.

Note: Unless you are in the middle of network migration from an Alteon version prior to 22.0, you should always enable DSSP version 2.

```
>> # /cfg/slb/gslb/version 2
```

3. On the Denver Alteon, define each remote site.

When you start configuring at the Denver site, Denver is local and San Jose is remote. Add and enable the remote Alteon's IP address interface. In this example, there is only one remote site: San Jose, with an IP interface address of 200.200.200.1. Use the following commands:

`>> # /cfg/slb/gslb/site 1`	(Select Remote Site 1)
`>> Remote site 1# prima 200.200.200.1`	(Define remote IP interface address)
`>> Remote site 1# ena`	(Enable Remote Site 1)

Each additional remote site would be configured in the same manner. You can enable up to 64 remote sites.

4. On the Denver Alteon, assign each remote distributed service to a local virtual server.

In this step, the local Denver site is configured to recognize the services offered at the remote San Jose site. As before, configure one real server entry on the Denver Alteon for each virtual server located at each remote site. Since there is only one remote site (San Jose) with only one virtual server, only one more local real server entry is needed at the Denver site.

The new real server entry is configured with the IP address of the remote virtual server, rather than the usual IP address of a local physical server. Do not confuse this value with the IP interface address on the remote Alteon.

The `remote` parameter is enabled, and the real server entry is added to the real server group under the local virtual server for the intended service. Finally, since the real server health checks are headed across the Internet, the health-checking interval should be increased to 30 or 60 seconds to avoid generating excess traffic. The more remote sites you have, the larger the time interval should be.

For example:

`>> Remote site 1# /cfg/slb/real 1`	(Create an entry for Real Server 1)
`>> Real server 1# rip 200.200.200.100`	(Set remote virtual server IP address)
`>> Real server 1# adv/remote enable`	(Define the real server as remote)
`>> Real server 1# inter 30`	(Set a high health check interval)
`>> Real server 1# ena`	(Enable the real server entry)
`>> Real server 1# /cfg/slb/group 1`	(Select appropriate. real server group)
`>> Real server group 1# add 1`	(Add Real Server 1 to Group 1)

Note: You should note where each configured value originates or this step can result in improper configuration.

5. On the Denver Alteon, define the domain name and hostname for each service hosted on each virtual server.

   These are the same as for the San Jose Alteon: the domain name is "gslb.example.com," and the hostname for the HTTP service is "www." Configure these values as follows:

`>> Real server group 1# /cfg/slb/virt 1`	(Select Virtual Server 1)
`>> Virtual server 1# dname gslb.example.com`	(Define domain name)
`>> Virtual server 1# service 80`	(Select HTTP for virtual server)
`>> Virtual server 1 http# hname www`	(Define HTTP hostname)

6. Apply and verify the configuration.

```
>> Global SLB# apply
>> Global SLB# cur
```

7. Examine the resulting information. If any settings are incorrect, make and apply any appropriate changes, and then check again.

```
>> Global SLB# /cfg/slb/cur
```

8. Save your new configuration changes. Table 62 shows the resulting port usage:

Table 62: Web Host Example: Port Usage

Port	Host	Layer 4 Processing
11	Server 11	Server
12	Server 12	Server
2	Default Gateway Router. This connects Alteon to the Internet where all client requests originate.	Client

Configuring a Standalone GSLB Domain

An Alteon can serve as a *standalone* GSLB device, meaning that it can perform GSLB health checking and site selection to other sites without supporting SLB to local real servers.

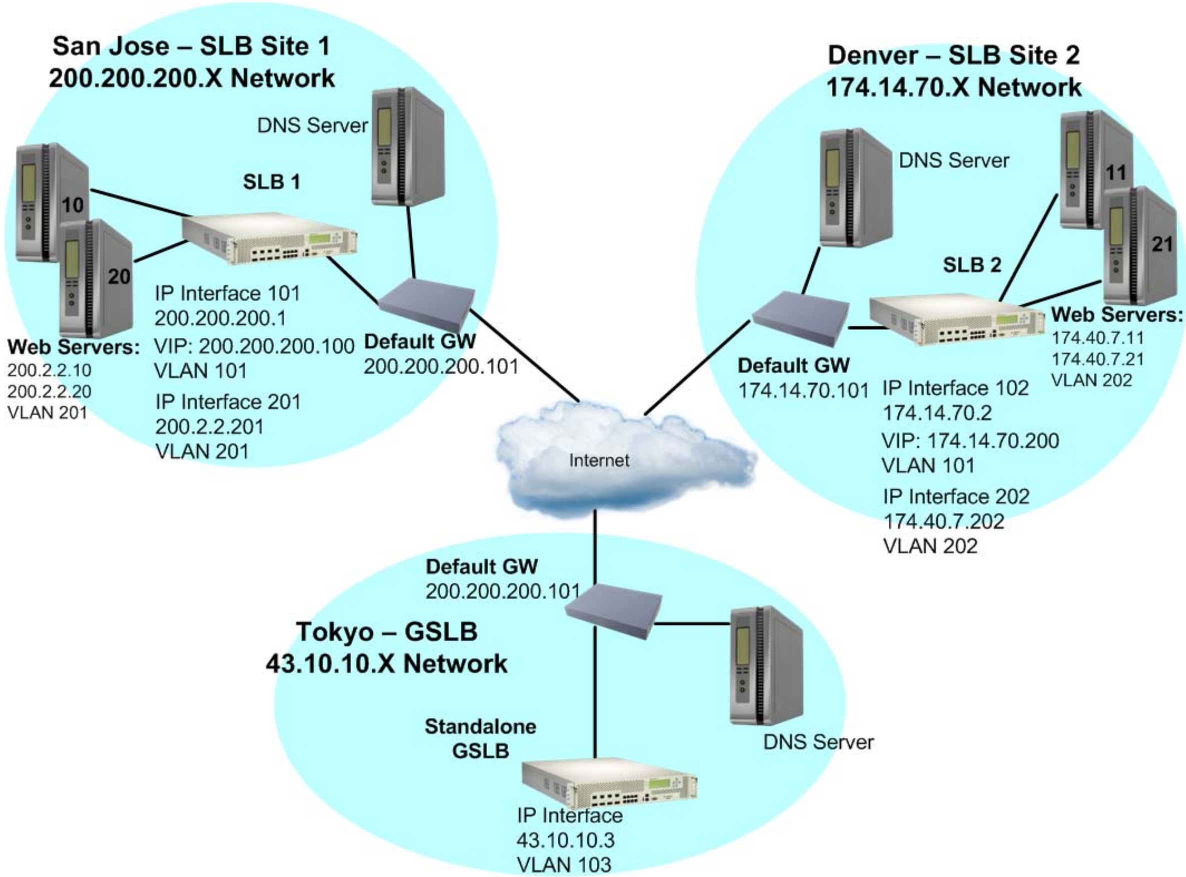
The remote sites can be other sites configured with an Alteon running GSLB, an Alteon running only SLB, or even a site that uses another vendor's load balancers.

An Alteon running GSLB can operate in standalone mode as long as it uses site selection metrics that do not require remote site updates.

Example GSLB Topology with a Standalone GSLB Site

The procedures to implement the example GSLB topology illustrated in Figure 121 - GSLB Topology with a Standalone GSLB Site Example, page 725 are described in this example.

Figure 121: GSLB Topology with a Standalone GSLB Site Example



To configure the basics at the Tokyo site

Following a similar procedure as described in Configuring Basic GSLB, page 713, configure a third site—Tokyo—in standalone mode.

Remember that in standalone mode, Alteon does not require SLB configuration of local real servers.

1. Optionally, on the Tokyo Alteon, configure management access and management gateway address.

`>> # /cfg/sys/mmgmt/addr 43.100.80.20`	(Management port IP address)
`>> Management Port# mask 255.255.255.0`	(Management port mask)
`>> Management Port# gw 43.100.80.1`	(Management port gateway address)
`>> Management Port# ena`	(Enable the management port)

2. Configure a VLAN for the Internet traffic.

```
>> # /cfg/l2/vlan 103/name internet           (VLAN 102 for Internet)
>> VLAN 103# add 3                            (Add Port 3 to VLAN 103)

Port 3 is an UNTAGGED port and its current PVID is 1.
Confirm changing PVID from 1 to 103 [y/n]: y
Current ports for VLAN 103:      empty
Pending new ports for VLAN 103:  3
Current status: disabled
New status:     enabled
```

3. Define an IP interface to the Internet.

```
>> # /cfg/l3/if 103                           (Select IP Interface 103)
>> IP Interface 103# addr 43.10.10.3          (Assign IP address for the
                                              interface)
>> IP Interface 103# mask 255.255.255.0       (Assign network mask)
>> IP Interface 103# ena                      (Enable IP Interface 103)
>> IP Interface 103# vlan 103                 (Assign interface to VLAN 103)
```

4. Define the default gateway.

```
>> IP Interface 103# /cfg/l3/gw 1             (Select Default Gateway 1)
>> Default gateway 1# addr 43.10.10.103       (Assign IP address for the gateway)
>> Default gateway 1# ena                     (Enable Default Gateway 1)
```

5. Apply and save the configuration.

```
>> # apply
>> # save
```

6. Configure the local DNS server to recognize the local GSLB device as the authoritative name server for the hosted services.

   Determine the domain name that will be distributed to both sites and the hostname for each distributed service. In this example, the Tokyo DNS server is configured to recognize 43.10.10.3 (the IP interface of the Tokyo GSLB device) as the authoritative name server for "www.gslb.example.com."

To configure the Tokyo site for GSLB

Following the similar procedure described for San Jose (see To configure the San Jose Site for GSLB, page 717), configure the Tokyo site as follows:

1. On the Tokyo Alteon, turn on SLB and GSLB.

```
>> # /cfg/slb                                 (Select the SLB Menu)
>> SLB# on                                    (Activate SLB for Alteon)
>> # /cfg/slb/gslb                            (Select the GSLB Menu)
>> Global SLB# on                             (Activate GSLB for Alteon)
```

2. On the Tokyo Alteon, assign each remote distributed service to a local virtual server.

   In this step, the local site, Tokyo, is configured to recognize the services offered at the remote San Jose and Denver sites. As before, configure one real server entry on the Tokyo Alteon for each virtual server located at each remote site.

   The new real server entry is configured with the IP address of the remote virtual server, rather than the usual IP address of a local physical server. Do not confuse this value with the IP interface address on the remote Alteon.

`>> # /cfg/slb/real 1`	(Create an entry for San Jose)
`>> Real server 1# ena`	(Enable the real server entry)
`>> Real server 1# name San_Jose`	(Set a name for the real server entry)
`>> Real server 1# rip 200.200.200.100`	(Set remote VIP address of San Jose)
`>> Real server 1# adv/remote enable`	(Define the real server as remote)
`>> # /cfg/slb/real 2`	(Create an entry for Denver)
`>> Real server 2# ena`	(Enable the real server entry)
`>> Real server 2# name Denver`	(Set a name for the real server entry)
`>> Real server 2# rip 74.14.70.200`	(Set remote VIP address for Denver)
`>> Real server 2# adv/remote enable`	(Define the real server as remote)

Note: You should note where each configured value originates, or this step can result in improper configuration.

3. Define a network that will match and accept all incoming traffic for the other sites.

`>> # /cfg/gslb/net 1`	(Create an entry for the new network)
`>> Network 1# ena`	(Enable the new network)
`>> Network 1# sip 0.0.0.0`	(Define a source IP address match)
`>> Network 1# mask 0.0.0.0`	(Define a network mask match)
`>> Network 1# addreal 1`	(Add the San Jose site to the network)
`>> Network 1# addreal 2`	(Add the Denver site to the network)

4. Define a new rule that will make the new network active.

`>> # /cfg/slb/gslb/rule 1/ena`	(Enable the new rule)
`>> Rule 1# dname gslb.example.com`	(Define a domain name)
`>> Rule 1# metric 1/gmetric network`	(Define the metric this rule will use)
`>> Rule 1# metric 1/addnet 1`	(Add network to the rule metric)

5. Apply and verify the configuration.

```
>> Virtual Server 2 http Service# apply
>> Global SLB# cur
```

6. Examine the resulting information. If any settings are incorrect, make and apply any appropriate changes, and then check again.

```
>> Global SLB# /cfg/slb/cur
```

7. Save your new configuration changes.

```
>> Layer 4# save
```

Note: Configuration for the Tokyo site is now complete.

Example Standalone DNS Server Configuration

The following is the DNS server functionality:

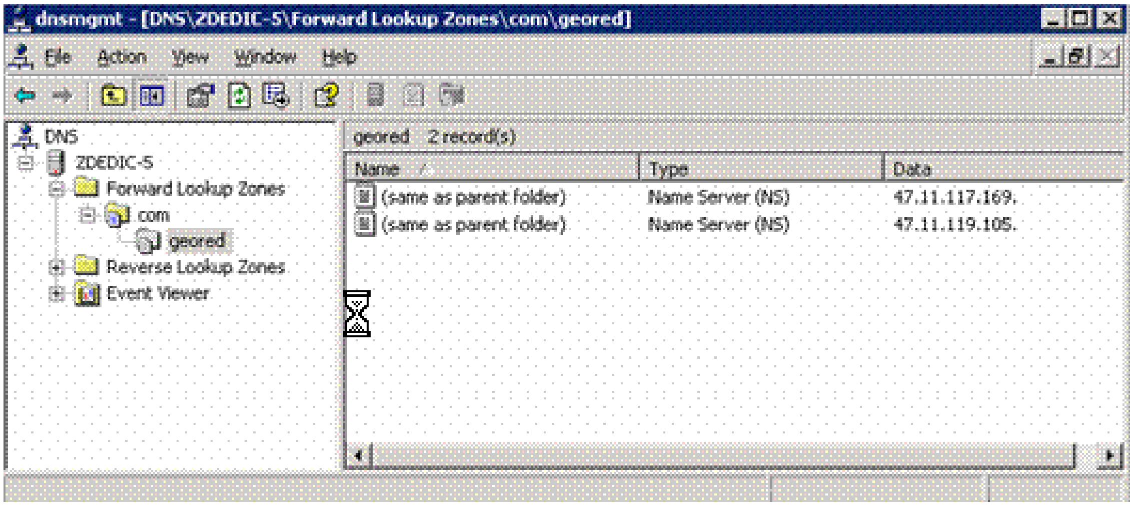
- It is possible to configure two (2) NS records: NS1 with Alteon1 interface IP address and NS2 with Alteon2 interface IP address.
- If Alteon2 is alive, Alteon1 is down (domain name cannot be resolved by using NS1) and DNS server switches to NS2.
- If Alteon1 is alive, Alteon2 is down (domain name cannot be resolved by using NS2) and DNS server switches to NS1.
- If Alteon1 is alive (NS1 is used), Alteon2 was down. After that, Alteon2 is alive and DNS server continues to use NS1.
- If Alteon2 is alive (NS2 is used), Alteon1 was down. After that, Alteon1 is alive and DNS server continues to use NS2
- The round-robin algorithm for DNS server can be disabled.

To configure a Microsoft Windows 2003 DNS Server

The DNS server is configured to resolve domain name (e.g. “geored.com”) into active Alteon virtual IP address which represents active MCS system (Alteon1 VIP1, Alteon1 VIP2, or Alteon2 VIP1, Alteon2 VIP2).

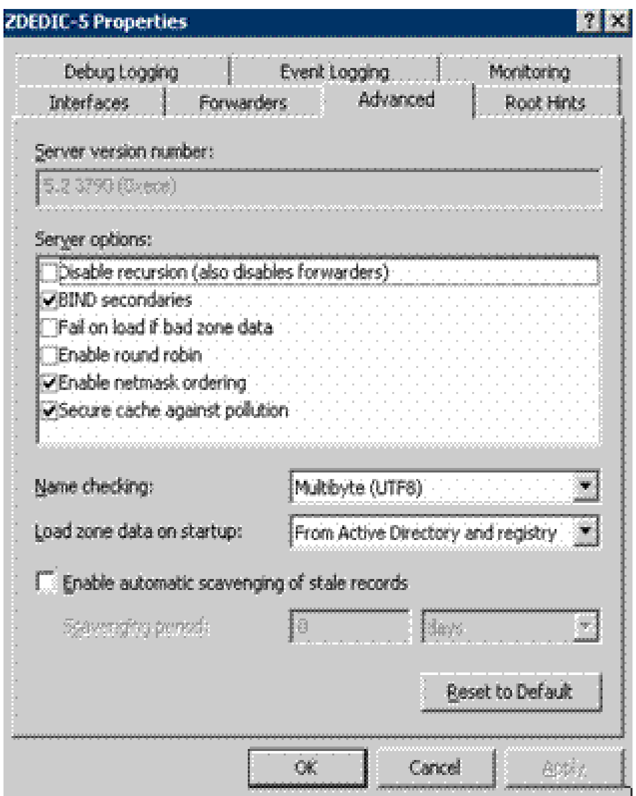
1. Open the DNS console.
2. Create a primary forward lookup zone “com”.
3. Create a delegation in zone “com”: Delegated domain “geored”; FQDN - “geored.com”.
4. Add the first resource record: FQDN—Alteon1 interface IP address (which is set up by using the command `/cfg/l3/if 1`); IP address – Alteon1 interface IP address.
5. Add second resource record: FQDN—Alteon2 interface IP address; IP address—Alteon2 interface IP address. This example configuration is shown in Figure 122 - DNS Console, page 729:

Figure 122: DNS Console



6. Set TTL equal to 10 seconds for records of zone “com”.
7. Disable the round-robin algorithm for the server as shown in Figure 123 - ZDEDIC-5 Properties Window, page 729.

Figure 123: ZDEDIC-5 Properties Window



Note: If the DNS server is down, the clients (PCC, Sigma phone that supports DNS and AudioCodes GW) do not work.

Master/Slave DNS Configuration

The following is the DNS configuration for a GSLB setup where each site contains a master and slave:

1. Add the first resource record, FQDN—Alteon1 (site1 master) interface IP address (set by the command **`/cfg/l3/if 1`**), where IP address is the Alteon1 interface IP address.
2. Add the second resource record, FQDN—Alteon2 (site1 backup) interface IP address.
3. Add the third resource record, FQDN—Alteon3 (site2 master) interface IP address.
4. Add the fourth resource record, FQDN—Alteon4 (site2 backup) interface IP address.

Configuring GSLB with Rules

GSLB rules can be configured on a per-domain basis to allow dynamic site selection based on time of day for a given domain. The criteria for domain rules are as follows:

- Each domain has one or more rules.
- Each rule has a metric preference list.
- The site selection selects the first rule that matches for the domain and starts with the first metric in the metric preference list of the rule.

Up to 128 rules can be configured, with up to eight metrics per rule. The site selection metric sequence in the default Rule 1 is as follows:

1. **Network Preference**—The first metric in Rule 1 is set to **Network Preference**, which selects the server based on the preferred network of the source IP address for a given domain. If preferred networks are not configured, this metric is not used in the default rule. For more information on configuring preferred networks, see Configuring GSLB Network Preference, page 733.
2. **None**—The second metric in Rule 1 is set to **None** in order to let you configure the local or availability metric here. The local server or the server with the highest availability is selected before any subsequent metric is used to select other servers.
3. **Geographical preference**—The third metric in Rule 1 is set to **Geographical Preference** so that the IANA-defined geographical region based on the client source IP is used to redirect the request to the client's region.
4. **Least connections**
5. **Round-robin**—*Round-robin*, or random, should be the last metric defined in a rule because it always returns a value if there is at least one functional site.
6. None
7. None
8. **Dnsalways**—The last metric in Rule 1 should be configured as **dnsalways** so that the GSLB selection mechanism selects at least the local virtual server when all servers are not available. In this case, metric 6 can be configured with DNS always.

Configuring Time-Based Rules

This section explains how to configure time-based rules.

To configure the first time-based rule

Using the base configuration Configuring Basic GSLB, page 713, you can define a new time-based rule for certain networks, as follows:

1. Disable the default Rule 1, in order to ensure that the time-based rule is processed first.

```
>> # /cfg/slb/gslb/rule 1/dis
```

2. Configure the networks to be added to the GSLB rule.

```
>> # /cfg/slb/gslb/net 43/sip 43.0.0.0/mask 255.0.0.0/addvirt 1/ena
>> # /cfg/slb/gslb/net 55/sip 55.0.0.0/mask 255.0.0.0/addreal 10/ena
>> # /cfg/slb/gslb/net 56/sip 56.0.0.0/mask 255.0.0.0/addreal 10/ena
```

3. Configure a new rule.

```
>> # /cfg/slb/gslb/rule 2
```

4. Specify a start and end time for this rule to be applied.

```
>> Rule 2# start 7 00/end 18 00/ena              (From 7AM until 6PM)
>> Rule 2# ena                                   (Enable the rule)
```

5. Specify the GSLB metrics to select a site if a server is not selected at first. Since network metric is the first metric, make sure to add the configured networks to metric 1.

```
>> # /cfg/slb/gslb/rule 2/metric 1/gmetric network
>> # /cfg/slb/gslb/rule 2/metric 1/addnet 43/addnet 55/addnet 56
```

6. Specify the other preferred GSLB metrics.

```
>> # /cfg/slb/gslb/rule 2/metric 2/gmetric geographical
>> # /cfg/slb/gslb/rule 2/metric 3/gmetric roundrobin
```

To configure the second time-based rule

Using the steps in configure the first time-based rule, page 731, configure another rule with the following parameters:

```
>> # /cfg/slb/gslb/net 48/sip 48.0.0.0/mask 240.0.0.0/addreal 2/en
>> # /cfg/slb/gslb/rule 4/start 18 00/end 7 00/ena
>> # /cfg/slb/gslb/rule 4/metric 1/gmetric network/addnet 48
>> # /cfg/slb/gslb/rule 4/metric 2/gmetric geographical
>> # /cfg/slb/gslb/rule 4/metric 3/gmetric random
```

1. Add the rule to the configured virtual server. Using the basic GSLB example, add the following command to the virtual server configuration as shown in step 5 in To configure the San Jose Site for GSLB, page 717:

`>> # /cfg/slb/virt 1/addrule 2/addrule 4`	(Add Rules 2 and 4 to the virtual server/domain)

2. Apply and save the configuration.

```
>> Rule 2 Metric 4# apply
>> Rule 2 Metric 4# save
```

Using the Availability Metric in a Rule

The availability metric selects the next server in a priority list when any capacity thresholds of the previous servers has been reached.

To use the availability metric in a rule

1. Set the availability metric for metric 2 in Rule 1.

```
>> # /cfg/slb/gslb/rule 1/metric 2/gmetric availability
```

2. Set the availability values for the real/virt servers. For example:

`>> Rule 1# /cfg/slb/virt 1/avail 11`	(Set available weight for virtual server)
`>> Rule 1# /cfg/slb/real 10/avail 22`	(Set available weight for real server)
`>> Rule 1# /cfg/slb/real 11/avail 33`	(Set available weight for real server)

3. Apply and save the configuration.

```
>> Rule 1 Metric 4# apply
>> Rule 1 Metric 4# save
```

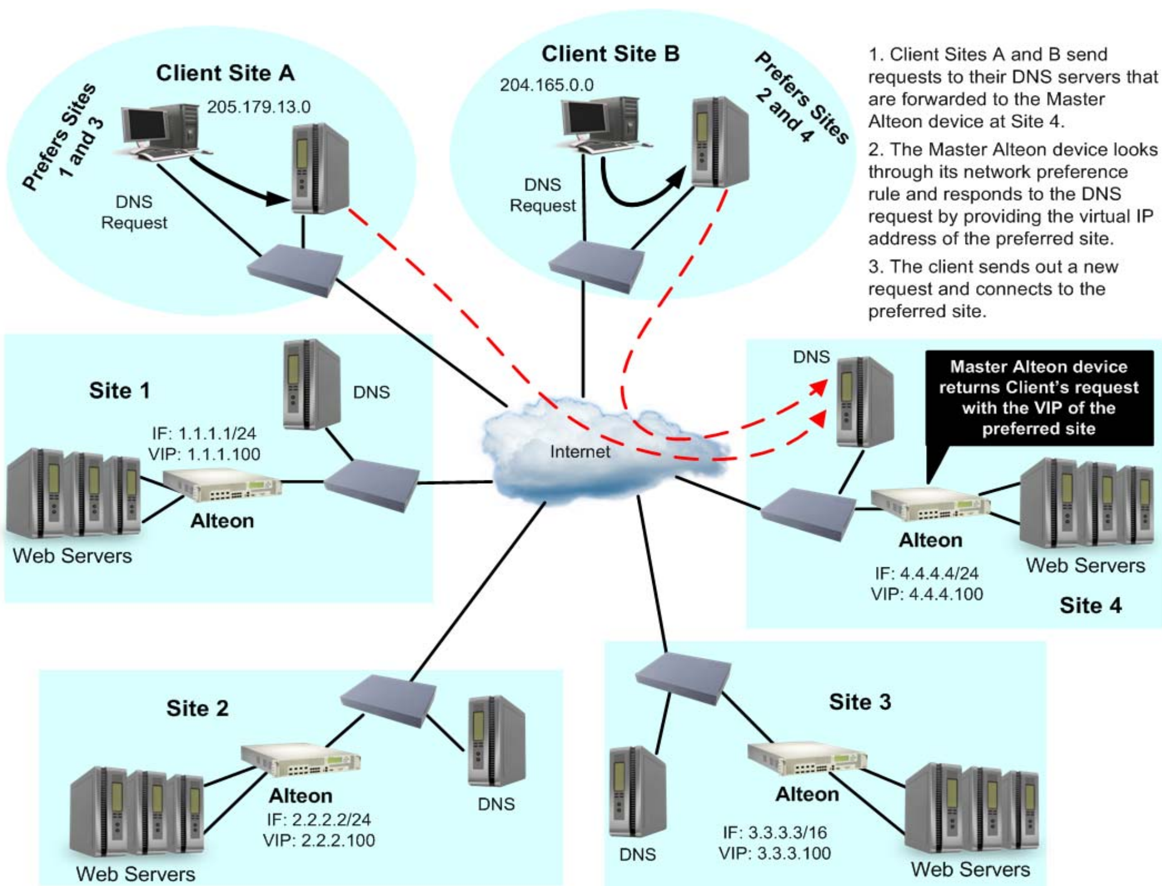
Configuring GSLB Network Preference

Alteon enables clients to select GSLB sites based on where the client is located. This is implemented by configuring network preference. Network preference selects the server based on the preferred network of the source IP address for a given domain. The preferred network contains a subset of the servers for the domain.

The example configuration in Figure 124 - Configuring the Client Proximity Table, page 733 illustrates how to create rules based on client network preference. Two client networks, A and B, are configured in the network preference rule on the master Alteon at Site 4. Client A with a subnet address of 205.178.13.0 is configured with a network preference rule for preferred Sites 1 and 3. Client B, with a subnet address of 204.165.0.0, is configured a network preference rule for preferred Sites 2 and 4.

Client A, with a source IP address of 205.178.13.10, initiates a request that is sent to the local DNS server. The local DNS server is configured to forward requests to the DNS server at Site 4. Alteon at Site 4 looks up its network preference and finds that Client A prefers to connect to Sites 1 or 3. Similarly, Client B's requests are always forwarded to Sites 2 or 4.

Figure 124: Configuring the Client Proximity Table



Note: The Alteon lets you configure up to 128 preferred client networks. Each network can contain up to 1023 real servers.

To configure network preferences on Alteon at Site 4

`>> # /cfg/slb/gslb/net 1/`	(Select Network 1)
`>> Network 1# sip 205.178.13.0`	(Assign source address for Client A)
`>> Network 1# mask 255.255.255.0`	(Assign the mask for Client A)
`>> Network 1# addreal 1/addreal 3`	(Add Real Servers 1 and 3)
`>> # /cfg/slb/gslb/net 2/`	(Select Network 2)
`>> Network 2# sip 204.165.0.0`	(Assign source address for Client B)
`>> Network 2# mask 255.255.0.0`	(Assign the mask for Client B)
`>> Network 2# addreal 2`	(Add Real Server 2)
`>> Network 2# addvir 4`	(Select preferred Site 4)
`>> # /cfg/slb/gslb/rule 1/metric 1`	(Select metric 1-network preference)
`>> Rule 1 Metric 2# addnet 1/addnet 2`	(Add Network 1 and Network 2)

Using this configuration, the DNS request for "radware.com" from client IP 205.178.13.0 receives a DNS response with only the virtual server IP address of Site 1, if Site 1 has less load than Site 3.

Configuring GSLB with Client Proximity

Using GSLB with the client proximity metric, Alteon selects the optimal site for the end-client. This is based on calculated shortest response time from site to site in GSLB mode. The GSLB client proximity metric calculates the response time between each data center site and end-client in Layer 7. The client proximity calculation provides better use of bandwidth and time.

When configuring client proximity:

- Ensure that dbind is enabled. If dbind is disabled, traffic goes to the MP and not the SP, impacting performance.
- Carefully analyze your network mask requirements. Increasing the client IP mask reduces computation time for client proximity, as the clients with the same subnet IP can reuse the client proximity that is already calculated.

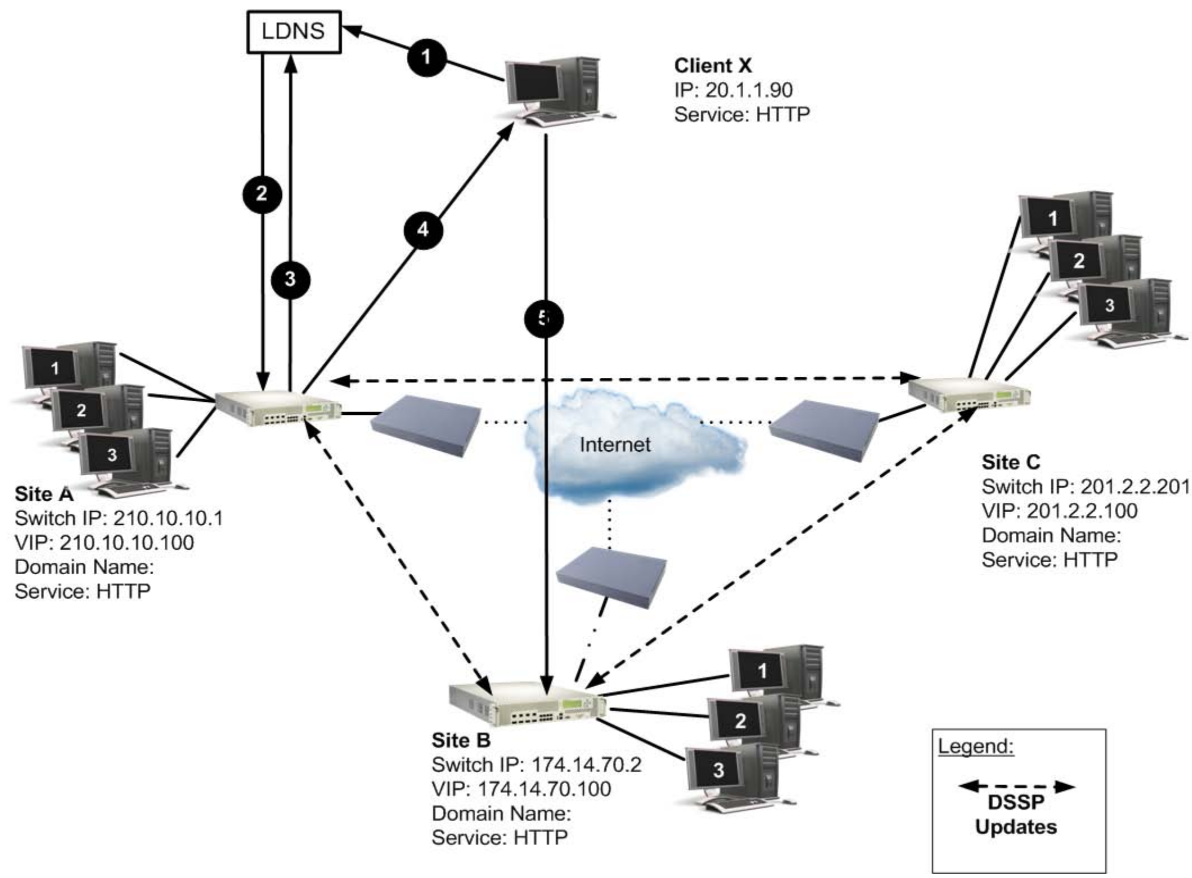
Client proximity entries can be generated statically or dynamically:

- Configuring Static Client Proximity, page 735
- Configuring Dynamic Client Proximity, page 742

Configuring Static Client Proximity

This section details the client proximity workflow and procedures to configure the sites for Figure 125 - GSLB Client Proximity Site with HTTPS Service, page 735.

Figure 125: GSLB Client Proximity Site with HTTPS Service



In this example, the order of preference for Client X is Site C followed by Site B and Site A. When Client X loads the browser and enters the URL www.radware.com/products/index.html, the system sends a DNS getHostByname query to the client's local DNS server for the www,radware.com IP address.

Workflow for GSLB Client Proximity Site with HTTPS Service

The following is the workflow for the example as shown Figure 125 - GSLB Client Proximity Site with HTTPS Service, page 735:

1. The Client X DNS requests the local DNS server to send the www.radware.com IP address.
2. The local DNS server queries the upstream DNS server on Alteon.
3. The Site A Alteon receives a DNS request and acts as the authoritative DNS. Site A responds to the DNS request with a Site A VIP address according to the DNS GSLB configured metric.
4. Client X opens an HTTP application session with an Alteon at Site A.
5. On receiving the request, Site A checks its client proximity table and finds a static entry. It identifies Site C to be the closest site and sends an HTTP 302 redirection with Site C IP address/domain name.
6. On receiving the request, Site C checks its client proximity table and serves the HTTP request.

In the client proximity table, the static client proximity entries are set to Site C as the closest.

Note: When the closest site is down, the client is redirected to the next closest site. In Figure 125 - GSLB Client Proximity Site with HTTPS Service, page 735, if Site A determines that Site C is down, it sends an HTTP redirect message with Site B VIP address/domain name.

To configure Site A

1. Enable SLB, Direct Access Mode (DAM), and GSLB globally.

`>> # /cfg/slb/on`	(Enable SLB)
`>> # /cfg/slb/adv/direct ena`	(Enable DAM)
`>> # /cfg/slb/gslb/on`	(Enable GSLB)

2. Set up DSSP version 4 for client proximity updates.

```
>> # /cfg/slb/gslb/version 4
```

3. Set up local real server. For example, the real server IP address is 10.10.10.12.

`>> # /cfg/slb/real 1`	(Assign local real server IP)
`ena`	
`ipver v4`	(Set the DSSP version to 4)
`rip 10.10.10.12`	

4. Set up remote real servers for Site B and Site C.

`>> # /cfg/slb/real 2/ena`
`ipver v4`
`rip 174.14.70.200`
	(Assign a real server to Site B)

`>> # /cfg/slb/real 2/adv` `remote ena`	(Enable remote the real server for Site B)

`>> # /cfg/slb/real 3/ena`
`ipver v4`
`rip 201.2.2.100`
	(Assign a real server to Site C)

`>> # /c/slb/real 3/adv remote ena`	(Enable remote the real server for Site C)

5. Configure SLB Group 1 with content-based health check.

`>> # /cfg/slb/group 1`	(Configure SLB Group 1)
`ipver v4`	
`health http`	(Enable HTTP-based health check)
`content "index.html"`	(Configure content-based health check for Group 1)
`add 1`	(Add Real Server 1)

`add 2`	(Add remote Real Server 2—Site B)
`add 3`	(add remote Real Server 3—Site C)

6. Enable client and server processing.

`>> # /cfg/slb/port 1` `server ena`	(Enable server processing)
`>> # /cfg/slb/port 8` `client ena`	(Enable client processing)
`server ena`	(Enable server processing for health packet in this port)

7. Configure Virtual IP for the local site.

`>> # /cfg/slb/virt 1`
`ena`
`ipver v4`
	(Configure virtual server)

`vip 210.10.10.100`	(Assign virtual IP address)
`dname "radware.com"`	(Assign domain name)

8. Enable virtual service with dbind and client proximity.

`>> # /cfg/slb/virt 1/service http group 1`	
`dbind ena`	(Enable delayed binding for HTTP service)
`>> # /cfg/slb/virt 1/service http/http` `clntprox http`	(Enable Client proximity for HTTP/ HTTPS service)

9. Configure the interfaces of the remote site for DSSP updates.

`>> # /cfg/slb/gslb/site 1` `ena`	(Enable Site B)
`prima 174.14.70.2`	(Remote Site Interface IP)
`>> # /cfg/slb/gslb/site 2` `ena`	(Enable Site C)
`prima 201.2.2.201`	(Remote Site Interface IP)

10. Create a static entry for each remote site with local VIP as the closest site. This prevents client proximity calculation for health check packets.

```
>> # /cfg/slb/gslb/network 1
ena
sip 174.14.70.2
mask 255.255.0.0
addvirt 1 1
>> # /cfg/slb/gslb/network 2
ena
sip 201.2.2.201
mask 255.255.0.0
addvirt 1 1
```

11. Configure a static entry for client network 20.0.0.0.

```
>> # /cfg/slb/gslb/network 3
ena
sip 20.1.1.10
mask 255.0.0.0
addvirt 1 30                                  (Least preferred site)
addreal 2 20
addreal 3 10                                  (Most preferred site)
```

To configure Site B

1. Enable SLB, DAM, and GSLB globally.

```
>> # /cfg/slb/on                              (Enable SLB)
>> # /cfg/slb/adv                             (Enable DAM)
direct ena
>> # /cfg/slb/gslb/on                         (Enable GSLB)
```

2. Set up DSSP version 4 for client proximity updates.

```
>> # /cfg/slb/gslb/version 4
```

3. Set up a local real server.

```
>> # /cfg/slb/real 1                          (Set the DSSP version to 4)
ena
ipver v4
rip 174.168.10.100                            (Assign local real server IP)
```

4. Set up remote real servers for Site A and Site C.

```
>> # /cfg/slb/real 2/ena                      (Assign real server to Site A)
ipver v4
rip 201.2.2.100
>> # /cfg/slb/real 2/adv                      (Enable remote real server for Site A)
remote ena
>> # /cfg/slb/real 3/ena                      (Assign real server to Site C)
ipver v4
rip 201.2.2.100
```

5. Configure SLB Group 1 with content-based health check.

```
>> # /cfg/slb/group 1                         (Configure SLB Group 1)
ipver v4
health http                                   (Enable HTTP-based health check)
content "index.html"                          (Configure content based health
                                              check for Group 1)
add 1                                         (Add Real Server 1)
add 2                                         (Add remote Real Server 2—Site B)
add 3                                         (add remote Real Server 3—Site C)
```

6. Enable client and server processing.

`>> # /cfg/slb/port 1`	(Enable server processing)
`server ena`	
`>> # /cfg/slb/port 8`	(Enable client processing)
`client ena`	
`server ena`	(Enable server processing for health packet in this port)

7. Configure virtual IP for the local site.

`>> # /cfg/slb/virt 1`	(Configure virtual server)
`ena`	
`ipver v4`	
`vip 174.14.70.100`	(Local VIP – Site B)
`dname "radware.com"`	(Assign domain name)

8. Enable virtual service with dbind and client proximity.

`>> # /cfg/slb/virt 1/service http`	
`group 1`	
`dbind ena`	(Enable delayed binding for HTTP service)
`>> # /cfg/slb/virt 1/service http/http`	(Enable Client proximity for HTTP/HTTPS service)
`clntprox http`	

9. Configure the interfaces of the remote site for DSSP updates.

`>> # /cfg/slb/gslb/site 1`	(Enable Site A)
`ena`	
`prima 210.10.10.1`	(Remote Site Interface IP)
`>> # /cfg/slb/gslb/site 2`	
`ena`	(Enable Site C)
`prima 201.2.2.201`	(Remote Site Interface IP)

10. Create a static entry for each remote site with local VIP as the closest site. This prevents client proximity calculation for health check packets.

```
>> # /cfg/slb/gslb/network 1
ena
sip 210.10.10.1
mask 255.255.0.0
addvirt 1 1
>> # /cfg/slb/gslb/network 2
ena
sip 201.2.2.201
mask 255.255.0.0
addvirt 1 1
```

11. Configure a static entry for client network 20.0.0.0.

```
>> # /cfg/slb/gslb/network 3ena
sip 20.1.1.10 mask 255.0.0.0
addvirt 1 20
addreal 2 10                                   (Most preferred site)
addreal 3 30                                   (Least preferred site)
```

To configure Site C

1. Enable SLB, DAM, and GSLB globally.

```
>> # /cfg/slb/on                               (Enable SLB)
>> # /cfg/slb/adv                              (Enable DAM)
direct ena
>> # /cfg/slb/gslb/on                          (Enable GSLB)
```

2. Set up DSSP version 4 for client proximity updates.

```
>> # /cfg/slb/gslb/version 4
```

3. Set up the local real server.

```
>> # /cfg/slb/real 1                           (Set the DSSP version to 4)
ena
ipver v4
rip 174.168.10.100                             (Assign local real server IP)
```

4. Set up remote real servers for Site A and Site B.

```
>> # /cfg/slb/real 2/ena                       (Assign real server to Site A)
ipver v4
rip 174.14.70.200
>> # /cfg/slb/real 2/adv                       (Enable remote real server for Site
remote ena                                     A)
>> # /cfg/slb/real 3/ena                       (Assign real server to Site B)
ipver v4
rip 201.2.2.100
```

5. Configure SLB group 1 with content-based health check.

```
>> # /cfg/slb/group 1                          (Configure SLB Group 1)
ipver v4
health http                                    (Enable HTTP-based health check)
content "index.html"                           (Configure content-based health
                                               check for Group 1)
add 1                                          (Add Real Server 1)
add 2                                          (Add remote Real Server 2—Site B)
add 3                                          (add remote Real Server 3—Site C)
```

6. Enable client and server processing.

`>> # /cfg/slb/port 1` `server ena`	(Enable server processing)
`>> # /cfg/slb/port 8` `client ena`	(Enable client processing)
`server ena`	(Enable server processing for health packet in this port)

7. Configure virtual IP for the local site.

`>> # /cfg/slb/virt 1`
`ena`
`ipver v4`
	(Configure virtual server)

`vip 201.2.2.100`	(Local VIP—Site C)
`dname "radware.com"`	(Assign domain name)

8. Enable virtual service with dbind and client proximity.

`>> # /cfg/slb/virt 1/service http` `group 1`	
`dbind ena`	(Enable delayed binding for HTTP service)
`>> # /cfg/slb/virt 1/service http/http` `clntprox http`	(Enable client proximity for HTTP/ HTTPS service)

9. Configure the interfaces of the remote site for DSSP updates.

`>> # /cfg/slb/gslb/site 1` `ena`	(Enable Site B)
`prima 174.14.70.1`	(Remote site interface IP)
`>> # /cfg/slb/gslb/site 2` `ena`	(Enable Site A)
`prima 210.10.10.1`	(Remote site interface IP)

10. Create a static entry for each remote site with local VIP as the closest site. This prevents client proximity calculation for health check packets.

```
>> # /cfg/slb/gslb/network 1
ena
sip 174.14.70.1
mask 255.255.0.0
addvirt 1 1
>> # /cfg/slb/gslb/network 2
ena
sip 210.10.10.1
mask 255.255.0.0
addvirt 1 1
```

11. Configure a static entry for client network 20.0.0.0.

```
>> # /cfg/slb/gslb/network 3
ena
sip 20.1.1.10
mask 255.0.0.0
addvirt 1 10                                  (Most preferred site)
addreal 2 20
addreal 3 30                                  (Least preferred site)
```

Configuring Dynamic Client Proximity

To configure dynamic client proximity for all sites according to the example, follow the procedure for Configuring Static Client Proximity, page 735, leaving out step 11.

For configuring the sites, see:

- To configure Site A, page 736
- To configure Site B, page 738
- To configure Site C, page 740

For the example, when Client X loads the browser and enters the URL www.radware.com/products/index.html, the system sends a DNS getHostByname query to the client's local DNS server for the www.radware.com IP address.

Workflow for GSLB Dynamic Client Proximity

The following is the workflow for the example as shown Figure 125 - GSLB Client Proximity Site with HTTPS Service, page 735 using HTTP-based dynamic client proximity:

1. The Client X DNS requests the local DNS server to send the www.radware.com IP address.
2. The local DNS server queries the upstream DNS server on Alteon.
3. The Site A Alteon receives a DNS request and acts as the authoritative DNS. Site A responds to the DNS request with a Site A VIP address according to the DNS GSLB configured metric.
4. The client opens an HTTP application session with Alteon at Site A.
5. Site A receives the HTTP request and checks the client proximity entry. If a client proximity entry does not exist, computation begins for this client network.
6. Alteon at Site A responds with three URL links. The Site A Alteon computes multi trip time (RTT) with the client from current connection and obtains remote site's RTT through DSSP updates. The following are the URL links at Site A:
   — http://<Site A IP address>/products/index.html
   — http://<Site B IP address>/radware_client_proximity_url
   — http://<Site C IP address>/radware_client_proximity_url
7. Client X sends an HTTP request to Site A, Site B, and Site C. Client X establishes a TCP connection with Site B and Site C, and sends a cntpurl request. Site B and C respond with a dummy response and in the process compute the RTT of their TCP connections with the Client X. Site B and Site C update the computed RTTs to Site A. On receiving RTT from Sites B and C, Site A sends the consolidated RTT list to all sites.
8. At this time, Site A serves the request from the client.
9. During the next request from the Client X, Site A redirects the HTTP request to the closest RTT Site (Site C in this example).
10. Client X opens a new connection with Site C.

11. Site C serves the HTTP request.

Note: When the closest site is down, Client X is redirected to the next best site. In the above example, if Site A determines that Site C is down, it sends an HTTP redirect message with the Site B VIP address/domain name.

Configuring GSLB with DNSSEC

Configuring GSLB with DNSSEC is an extension of the standard configuration procedure for GSLB. Alteon provides the zones in which you can configure DNSSEC.

Note: Ensure that the time and date are configured correctly in the GSLB configuration for all Alteons. Radware recommends that you manually configure the time date using NTP.

This section includes the following topics:

Basic DNSSEC Configuration

For DNSSEC to work with GSLB, you must perform the following:

1. Enable DNSSEC.
2. Create a Key Signing Key (KSK) and a Zone Signing Key (ZSK).
3. Associate the ZSK and KSK with a DNS zone.
4. Export the KSK Delegation Signer (DS) to the parent of the zone.

   For example, if you have a domain called *mywebhosting.radware.com*, the parent of the domain resides in *radware.com.*

To configure DNSSEC to work with GSLB

1. Access the *GSLB* menu and turn DNSSEC on.

```
>> Main# /cfg/slb/gslb/dnssec/on
```

2. Create a Key Signing Key (KSK) and define its parameters.

```
>> Main# /cfg/slb/gslb/dnssec/key
Enter key id: examplekey
>> Key examplekey# generate
Enter key type [zsk | ksk]: ksk
Should the key be enabled (yes/no)? [yes|no] [yes]
Enter key size [1024|2048|4096] [2048]:
Enter key algorithm RSA/SHA1, RSA/SHA256, RSA/SHA512 [1|256|512] [1]:
Enter key TTL in seconds [0-86400] [86400]:
Enter key expiration in seconds (0-2147483647) [0]:
Enter key rollover period in seconds (0-2147483647) [0]:
Enter key signature validity period in seconds (0-2147483647) [604800]:
Enter key signature publication period in seconds (0-2147483647) [302400]:
Generating key. Please wait.
Key examplekey added.
```

3. Create a Zone Signing Key (ZSK) and define its parameters by repeating the same procedure with the key type ZSK.

```
>> Main# /cfg/slb/gslb/dnssec/key
Enter key id: examplekey
>> Key examplekey# generate
Enter key type [zsk | ksk]: zsk
Should the key be enabled (yes/no)? [yes|no] [yes]
Enter key size [1024|2048|4096] [2048]:
Enter key algorithm RSA/SHA1, RSA/SHA256, RSA/SHA512 [1|256|512] [1]:
Enter key TTL in seconds [0-86400] [86400]:
Enter key expiration in seconds (0-2147483647) [0]:
Enter key rollover period in seconds (0-2147483647) [0]:
Enter key signature validity period in seconds (0-2147483647) [604800]:
Enter key signature publication period in seconds (0-2147483647) [302400]:
Generating key. Please wait.
Key examplekey added.
```

4. Associate the KSK and ZSK with a DNS zone.

Note: DNS zones are explicitly derived from the **dname** parameter specified in the GSLB configuration.

```
>> Main# /cfg/slb/gslb/dnssec/zonekey
Enter DNS-Zone-to-key entry id: example
Zone example# zone
Current DNS Zone:
Enter new DNS Zone: radware.com
>> Zone example# addksk
Select KSK: examplekey
Association between zone example and KSK examplekey created.
>> Zone example# addzsk
Select ZSK: examplekey
Association between zone example and ZSK examplekey created.
```

5. Export the KSK as text using the DS option.

```
>> Main# /cfg/slb/gslb/dnssec/export
Select key ID to export: examplekey
Enter component type to export [Key|DNSKEY|ds-record]: ds-record
Exporting [ZSK | KSK] examplekey in PEM format.
Export to text or file [text|file]: text
-----BEGIN [KEY|ZONE] SIGNING KEY-----
```

Your zone is DNSSEC configured.

Notes

- The DS export is a manual process that needs administrator validation at both ends (the parent and child zones).
- You can perform this procedure over a secure connection, such as HTTPS or SSH.
- Timers are defined per key, not globally.
- When working with GSLB and DNSSEC enabled, the configuration of remote sites must be identical for all Alteons participating in the GSLB configuration (`/cfg/slb/gslb/site x`). See Example : Configuring Identical Remote Sites with GSLB and DNSSEC, page 745.

Example : Configuring Identical Remote Sites with GSLB and DNSSEC

There are 3 sites:

- Site A—Denver
- Site B—New York
- Site C—London

Although the configuration is asymmetric

- Site A holds www.denver.com and www.london.com.
- Site B holds www.newyork.com, www.denver.com and www.london.com.
- Site C holds www.London.com and www.newyork.com.

In the site DSSP configuration, each site contains the configuration of the other sites (remote IP address). The following is an example set of parameters of the Denver site:

```
# /cfg/slg/gslb/site 1 (London)
Remote site 1# prima 1.2.3.4  (London IP)
Remote site 1# ena
```

All IP addresses of all the sites must be configured on all Alteons participating in the GSLB DNSSEC configuration.

DNSSEC Key Rollover

DNSSEC key maintenance requires administrative logic and deals with issues such as key revocation, key expiration, and key compromise. RFC 4641 (DNSSEC Operational Practices) advises how to manage keys and what are the recommended maintenance procedures.

A rollover is an automated process during which new DNSSEC keys are created, existing records are resigned, old DNSSEC keys are revoked, and new keys are published to the public using the Internet. An automated rollover is initiated periodically by the system administrator. An emergency rollover is initiated as necessary.

Contrary to other cipher key mechanisms that are revoked and created, DNSSEC rollover is an essential part of the RFC definition to ensure the continuous service for global Internet service.

This section includes the following sub-sections:

Preventing Expiration of KSK or ZSK in Rollover Situations

Alteon includes a DNS key rollover mechanism for preventing expiration. The following information is relevant when the ZSK and the KSK are assigned to the same zone. The goal of an automatic rollover process is that the created key is published and RRs are signed before the old key is revoked.

- During key rollovers (automatic, emergency, KSK or ZSK), the KSK must finalize before the ZSK rollover begins.
- To prevent overload on the CPU when creating keys, limit the number of bulk keys to be created to 10 at a time. If more keys are needed, their creation is queued.
- During an emergency rollover, the emergency rollover takes precedence over any other type of rollover. For example, when the administrator has four ZSKs in queue for automatic rollover and activates a ZSK emergency for another ZSK, the emergency ZSK is executed directly. Existing rollovers of the *same key* are cancelled and a console or syslog message is generated.

Automated ZSK Rollover

Alteon includes the following automated ZSK rollover methods:

- Zone Signing Key—As specified in RFC 4641, section 4.2.1.1. Pre-Publish Key Rollover
- Key Signing Key—As specified in RFC 4641, section 4.2.2

The automatic rollover of the DNSSEC keys is performed according to the parameters specified in Table 63:

Table 63: Automated ZSK Rollover as Defined in RFC 4641

Initial DNSKEY	New DNSKEY	New RRSIGs	DNSKEY Removal
SOA0	SOA1	SOA2	SOA3
RRRSIG10(SOA0)	RRRSIG10(SOA1)	RRRSIG10(SOA2)	RRRSIG10(SOA3)
DNSKEY1	DNSKEY1	DNSKEY1	DNSKEY1
DNSKEY10	DNSKEY10	DNSKEY10	DNSKEY10
	DNSKEY11	DNSKEY11	

Table 63: Automated ZSK Rollover as Defined in RFC 4641

Initial DNSKEY	New DNSKEY	New RRSIGs	DNSKEY Removal
RRSIG1 (DNSKEY)	RRSIG1 (DNSKEY)	RRSIG1 (DNSKEY)	RRSIG1 (DNSKEY)
RRSIG10 (DNSKEY)	RRSIG10 (DNSKEY)	RRSIG11 (DNSKEY)	RRSIG11 (DNSKEY)

To initiate a ZSK rollover

- Initiate the automatic rollover using the timer.
- To initiate an immediate rollover, set the timer to 0.

Note: Radware does not recommend the initiation of an immediate rollover.

As a result, the following occurs:

1. A new ZSK is created and stored in the key storage location.
2. The system administrator is notified through SNMP, console,, or e-mail that a new ZSK has been created.
3. The new ZSK is published using DNSKEY.
4. The system administrator is notified through SNMP, console, or e-mail that a new ZSK has been published to the supporting ISP.
5. A timeout of 12 hours, in addition to the TTL of the original ZSK, starts before enabling the DNSKEY publication.
6. All zone records are signed with the new ZSK, including all RRSIGs still existing in cache.
7. The old RRSIGs are removed from storage. The old ZSK remains in storage and is publicly available using DNSKEY.
8. A timeout of 12 hours, in addition to the TTL of the highest signed RRSIG, starts.
9. The old ZSK is revoked and is removed from storage.

Automated KSK Rollover

The *expiration period* is the period for which the key is valid (for example, one month). The *rollover period* is defined in Alteon as the period during which the rollover will be finished before the key expiration period starts. When entering the value, ensure that it is valid and does not overlap with the expiration date.

To initiate a KSK rollover

- Initiate the automatic rollover using the timer.
- To initiate an immediate rollover, set the timer to 0.

Note: Radware does not recommend the initiation of an immediate rollover.

As a result, the following occurs:

1. A new KSK is created and stored in the key storage location.
2. All the relevant keys are signed with the new KSK.
3. The new KSK is published using DNSKEY.
4. The system administrator is notified through SNMP, console, or e-mail that a new KSK has been created.
5. The KSK rollover is counted to zero.
6. The resource record of the parent points to the new DNSKEY.
7. A timeout of 48 hours, in addition to the TTL of the original KSK, starts.
8. The old DNSKEY is removed.
9. The system administrator is notified through SNMP, console, or e-mail that a new KSK is created and in place.

Emergency Rollovers

Emergency rollover is an administrator action.

When an emergency KSK rollover is enabled, Alteon waits for the DS record to be signed by the parent. The timer waits a pre-defined period (KSK Rollover Phase timer). If the administrator does not ensure that the DS was signed, a warning is issued that the DNSSEC service might be disturbed.

To initiate a ZSK emergency rollover

1. Initiate the emergency rollover.

   The system administrator is warned through SNMP, console, or e-mail that an emergency ZSK rollover has been initiated, which can disrupt services.
2. The system administrator must confirm the emergency rollover.

   The system administrator is notified through SNMP, console, or e-mail that a new ZSK has been created.
3. A new ZSK is created and stored in the key storage location.
4. The new ZSK is signed with the existing ZSK.
5. The new ZSK is published using DNSKEY.
6. All zone records are signed with the new ZSK, including all RRSIGs still existing in cache.
7. The old RRSIGs are removed from storage.
8. The old ZSK are revoked and removed from storage.
9. The system administrator is notified through SNMP, console, or e-mail that the emergency rollover is complete.

To initiate a KSK emergency rollover

Initiate the emergency rollover. As a result, the following occurs:

1. A new KSK is created and stored in the key storage location.
2. All the relevant keys are signed with the new KSK.
3. The new KSK is published using DNSKEY.

4. The system administrator is notified through SNMP, console, or e-mail that a new emergency KSK has been created.
5. The KSK rollover is counted to zero.
6. The RR of the Parent must point to the new DNSKEY.
7. A timeout of 48 hours in addition to the TTL of the original KSK starts.
8. The old DNSKEY is removed.
9. The system administrator is notified through SNMP, console, or e-mail that a new emergency KSK is in place.
10. All KSKs linked to this KSK are signed with the expiration that was set by the user.

Importing and Exporting Keys

After a key is created, it is imported and exported as necessary.

- DNSSEC keys are exported either for backup purposes or to export of a DS record to be signed by the parent of the domain. DNSSEC keys can be exported in their entirety (private and public keys), just like SSL keys. For more information regarding SSL keys, see Offloading SSL Encryption and Authentication, page 337.

  In addition, DNSSEC keys can be exported publicly (either a DS or DNSKEY), where only the public key is exported.

  When a private key is exported, it is encrypted with a one-time passphrase supplied at the time of export. This same passphrase is supplied during import for decrypting of the keys.

  When exporting keys, the digital properties of the keys are exported regardless of the zone assignments.

  During a DNSSEC private key export, the digital part of the key (private and public) is exported, and the key does not hold any relevant zone information. The zone information is only part of the DNSKEY Zone assignment.

  When exporting a public key, only the DNSKEY with all the relevant DNSSEC key properties and features (DS, TTLS, zone assignment, timer values and so on) is exported. When exporting a KSK in DS format, the key must be signed by the parent of the domain. Make sure to manually send the DS export to be signed by the parent of the domain.
- When importing keys, you import DNSSEC key properties, such as timers, which require user input. After importing, a DNSKEY is not functional unless it is assigned to a zone.

To import a key

ZSKs and KSKs are imported in the same way.

1. Access the *DNSSEC import* menu.

```
>> /cfg/slb/gslb/dnssec/import
```

2. Select the zone from which the ZSK or KSK are imported.

3. The following is an example set of parameters to enter at each prompt:

```
Select key id: 12
Enter key type (KSK or ZSK) [KSK|ZSK] [ZSK]: zsk
Enter key passphrase:
Import from text or file in PEM format [text|file] [file]: text
Should the key be enabled (yes/no)? [yes|no] [yes]: no
Enter key size (1024, 2048 or 4096) [1024|2048|4096] [1024]:
Enter key hash algorithm (encryption is always RSA) [RSA-SHA1|RSA-SHA256|RSA-
SHA512] [RSA-SHA1]:
Enter key ttl in seconds (0-86400) [86400]:
Enter key expiration in seconds (0-2147483647) [2419200]:
Enter key rollover period in seconds (0-2147483647) [1814400]:
Enter key signature validity period in seconds (0-2147483647) [604800]:
Enter key signature publication period in seconds (0-2147483647) [302400]:
*** At Import (called by user) key_id 12 type 1 passphrase=1234 format=0 ena=0
keysize=0 alg=5 ttl=86400 exp=2419200 rollover=1814400 validity=604800
publication=302400
```

To export a key to a file

1. Access the *DNSSEC export* menu.

```
>> /cfg/slb/gslb/dnssec/export
```

2. Select the zone from which the ZSK or KSK are exported.
3. The following is an example set of parameters to enter at each prompt:

```
Enter key id: 45
Enter component type to export  [key|dnskey] [key]: key
Enter passphrase:
Reconfirm passphrase:
Export to text or file [text|file] [file]: file
Enter hostname or IP address of SCP server: 10.241.1.77
Enter name of file on SCP server: a.c
Enter username for SCP server: anonymouys
Enter password for username on SCP server
```

To export a key to text

1. Access the *DNSSEC export* menu.

```
>> /cfg/slb/gslb/dnssec/export
```

2. Select the zone from which the ZSK or KSK are exported.

3. The following is an example set of parameters to enter at each prompt:

Note: The export type DS format is for KSK export only. For more information on DNSSEC export types, see the *Alteon Application Switch Operating System Command Reference*.

```
Enter key id: 45
Enter component type to export  [key|dnskey] [key]: key
Enter passphrase:
Reconfirm passphrase:
Error: passphrase confirmation failure
Enter passphrase:
Reconfirm passphrase:
Export to text or file [text|file] [file]: text
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,B5FBFDCB02000DFB

dw9lJ4p6GqCmhAnL7EUd7r0KyDHE4t8KnZj2grEoHBQqa+K2dXAfBxbp1jriNqwX
sFJpuTUBT9nLaRL9W8Ays8LC2pbHaSenUn3TrUonltaX/c49s90mHZwlcF5aX16t
WzmoKGqj20KrpG2ksXf3ADoP0eR7OMfUOeFJcDuz6omYk44gcREc1ljYg3N5IfOQ
eY22XNUiFTirlqfBeyKedhohMGGpggKM/0XU45NHVctcwalVJ5BnWPz4dELchjBG
aMPTQjerXg3dI0mKbWsgRgfbpwEm7DNTzhLAOg2+6/eoxN45V3VixeZHYd4rw/u8
RQo/P2rCvVD0ucOYvIgDyVxnCdh+f93JSgIBpmvO3z/zbhCddUzv5iHB1K8Ieujs
7HYUxGgdv/SSuf+ciRhNoxWXHVm/027ZuN84QsxW4KI3NmjTYI4jDkeUARznDeal
TcXkXCH/u18u9NUCvlo04djnzvs3uB/Ryw+qLtMIupFFJiOHu4Ckx+d3WPI5k9Sz
XEsSYhnSfnmGNT7oR4U3SVUkUdmD72wYQzxteWuFaTu4psM4Gi0oXfFmbYKj09AA
CYZ73ElFF0Ce+dpU2o2JYp4h8JTbRc+7KiO3yzzlS27/9WFxOkAR99tYxcII33g2
Q8+zpJr3BEkUClQbQv7II+y3BKZHm1VvObP6BZCfj2awZ+1lbuKBFoRWs6y6vOkr
dY59fCJfIJMVkVyMWm6pTUtNEO0FjCKT+lW6bdZZfSBWDtFgoIrZlSSqa01Itvga
7OsJWN/WydT26dn7M5fQARWvE9zAn/FGopa2qArlahYq6ja+W4QDp7oV1zZyeYvn
nFHEixoAMl159dz68dCpV7y1+VWAnGcSgp/+PE0DU24+uGFMuEa8pdulY9y7IjaY
HEiWqM4E2FOiiv8DhczNSwiZD1p3mN7BI4X9pT4BgldT/KMMkEFqUWhRUgz9iRy1
INYYHZbwfAFcuPY8w6jNTZaYRzkwo0EiLjhXjd6tiGuP5WYN+aLv3rw0OSNMhLaZ
fj4c3VQ0xTOyUrwrRuYOAv0NPK9PyIOlGdO9PvSrEO9+5s8+NcBiQ54RV6RImLb6
RNQqcz+VgEkawgYDIX9oT13G3SwNF4nb+q4+VsZFDOJjTYBdL+TlAol3TvPamGtM
0IShly783BKobXUxAP85Zf3Ce3+zqMjZH5x1cj/N6s3nGQCv/6jf+x+bJLCrXdHi
Yp3T4c+xXRH2tRtXVkFoXyHsijrqfQc/Y8A0ukDhenPupv7qcYmfcxo/AOobQaPI
a6Fk5Kz0DGrowv5+JqP+Jf+KXd/2EDw8Us0v5MdtLXlPKD9zqAhjIpRabglGmHrk
CXl3VZXTGNWF9NNNxDhz/z3tTFHy4A2JVGmIO92O1J3uEnkjElaDhAmRkaN+uxj+
HSJNkBaQybBK1GnsL4ag4EdpFY/uLfpoEysUW7rgJVN14zRxD7W9uMg0TQ9zycI6
lwla/FfYQYG1fQOOXLsaY/RRnLqg6pCliZh2TPy6MVaBOtHNQyZwa4Xykax3g4SJ
1qcjEWbpQL6i1isUWvQy9vsY1glm7ldMUScVGTBDASTVpJpjXgSdaqNGmV02sgJk
-----END RSA PRIVATE KEY-----
```

Deleting Keys

1. Access the *DNSSEC Key* menu.

```
>> /cfg/slb/gslb/dnssec/key
Enter key id:
Enter key id: 123
------------------------------------------------------------
[Key 123 Menu]
      generate - Create new key
      expire   - Set key expiration period
      rollover - Set key rollover period
      sigvalid - Set key signature validity period
      sigpub   - Set key signature publication period
      del      - Delete key
      ena      - Enable entry
      dis      - Disable entry
      cur      - Display current key configuration
```

2. Delete the selected key.

```
>> Key 123# del
Confirm deletion of this key? (y/n) [n]:
```

NSEC and NSEC3 Records

DNSSEC authenticates denial of existence by using NSEC and NSEC3 records. An NSEC is used to prove that a name does not exist. When a record does not exist, the DNS server (Alteon) answers with an NSEC DNS signature using the closest lexicographic name of the request.

Example

The DNS server holds the *example.com* domain and has records for *a.example.com* and *c.example.com*. When someone asks for *b.example.com*, the DNS server responds with an NSEC for *a.example.com* and *c.example.com*.

Automatic NSEC and NSEC3 Record Creation

The following procedure occurs:

1. Alteon receives a DNS query.
2. One of the following occurs:
   - If the domain name and a matching record exists, the regular GSLB DNSSEC procedure is followed.
   - If the domain name exists but no matching record exists, Alteon returns the NSEC or NSEC3 record of the requested name.
   - If neither the domain name nor a matching record exists, Alteon drops the DNS request.

Note: When issuing an NSEC RRSIG answer, the DNS server uses only one record (NSEC or NSEC3).

Configuring GSLB with Proxy IP for Non-HTTP Redirects

Typically, client requests for HTTP applications are redirected to the location with the best response and least load for the requested content. The HTTP protocol has a built-in redirection function that allows requests to be redirected to an alternate site. However, if a client requests a non-HTTP application such as FTP, POP3, or SMTP, then the lack of a redirection functionality in these applications requires that a proxy IP address be configured on the client port. The client port initiates a redirect only if resources are unavailable at the first site.

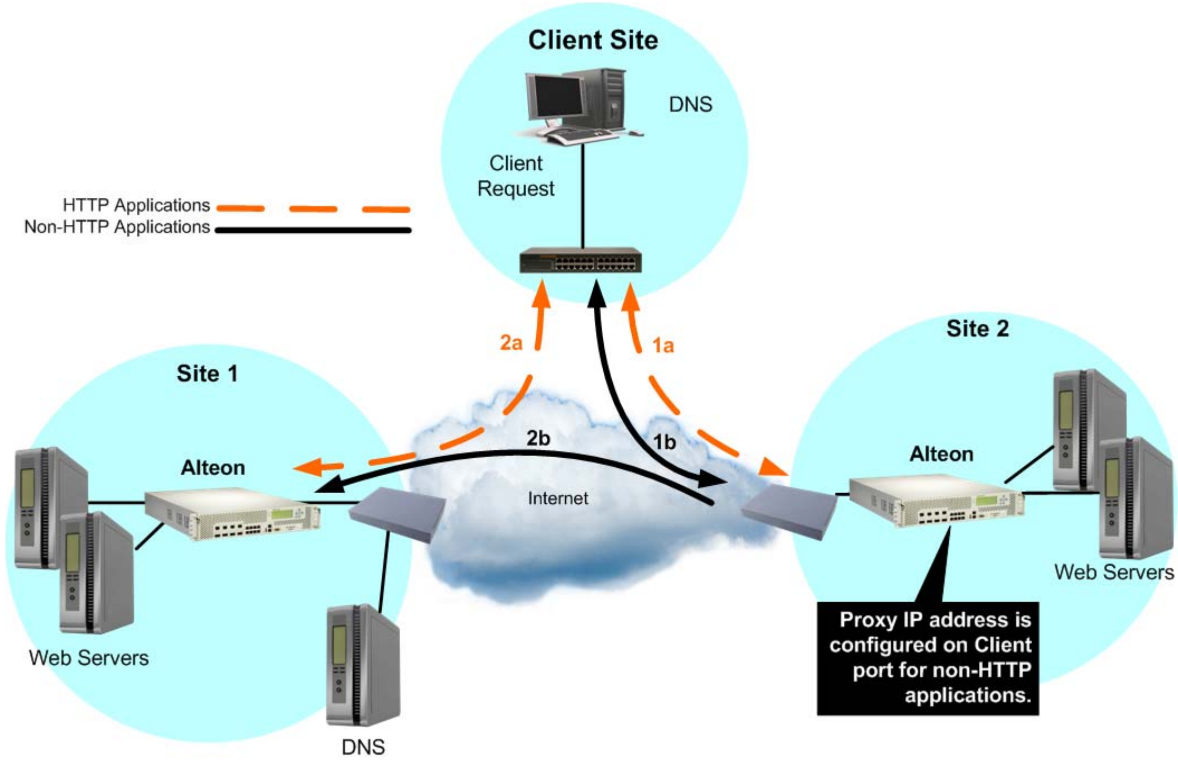
Note: This feature should be used as the method of last resort for GSLB implementations in topologies where the remote servers are usually virtual server IP addresses in other Alteons.

Figure 126 - HTTP and Non-HTTP Redirects, page 754 illustrates the packet-flow of HTTP and non-HTTP redirects in a GSLB environment. The following table explains the HTTP or non-HTTP request from the client when it reaches Site 2, but Site 2 has no available services.

Table 64: HTTP versus Non-HTTP Redirects

Application Type	Site 2 Alteon	Site 1 Alteon
HTTP application (built-in redirection)	**1a**—The client HTTP request reaches Site 2. Resources are unavailable at Site 2. Site 2 sends an HTTP redirect to a client with Site 1's virtual server IP address.	**2a**—The client resends the request to Site 1. Resources are available at Site 1.
Non-HTTP application (no redirection)	**1b**—The client non-HTTP request reaches Site 2. Resources are unavailable at Site 2. Site 2 sends a request to Site 1 with Site 2's proxy address as the source IP address and the virtual server IP address of Site 1 as the destination IP address.	**2b**—Site 1 processes the client proxy IP request. Resources are available at Site 1. Site 1 returns request to proxy IP port on Site 2.

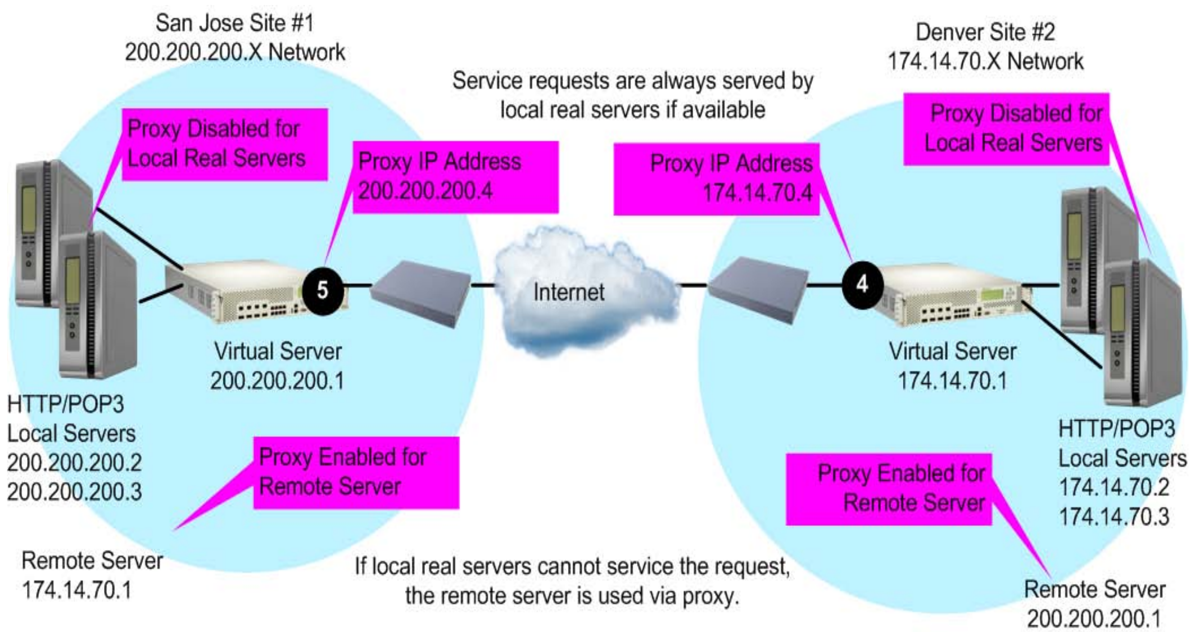
Figure 126: HTTP and Non-HTTP Redirects



How Proxy IP Works

Figure 127 - POP3 Request Fulfilled via IP Proxy, page 755 illustrates two GSLB sites deployed in San Jose and Denver. The applications being load balanced are HTTP and POP3. Any request that cannot be serviced locally is sent to the peer site. HTTP requests are sent to the peer site using HTTP redirect. Any other application request is sent to the peer site using the proxy IP feature.

Figure 127: POP3 Request Fulfilled via IP Proxy



The following procedure explains the three-way handshake between the two sites and the client for a non-HTTP application (POP3):

1. A user POP3 TCP SYN request is received by the virtual server at Site 2. Alteon at that site determines that there are no local resources to handle the request.
2. The Site 2 Alteon rewrites the request such that it now contains a client proxy IP address as the source IP address, and the virtual server IP address at Site 1 as the destination IP address.
3. Alteon at Site 1 receives the POP3 TCP SYN request to its virtual server. The request looks like a normal SYN frame, so it performs normal local load balancing.
4. Internally at Site 1, Alteon and the real servers exchange information. The TCP SYN ACK from Site 1's local real server is sent back to the IP address specified by the proxy IP address.
5. The Site 1 Alteon sends the TCP SYN ACK frame to Site 2, with Site 1's virtual server IP address as the source IP address, and Site 2's proxy IP address as the destination IP address.
6. Alteon at Site 1 receives the frame and translates it, using Site 1's virtual server IP address as the source IP address and the client's IP address as the destination IP address.

This cycle continues for the remaining frames to transmit all the client's mail, until a FIN frame is received.

Configuring Proxy IP Addresses

Alteon at Site 1 in San Jose is configured with port 6 connecting to the default gateway and Real Server 3 represents the remote server in Denver.

To configure the proxy address at Site 1 in San Jose for the remote server in Denver

1. Issue the following commands:

`>> # /cfg/slb/pip`	(Select the *Proxy IP Address* menu)
`>> Proxy IP address# type port`	(Use port-based proxy IP)
`>> Proxy IP address# add 200.200.200.4`	(Set unique proxy IP address)
`>> # /cfg/slb/port 6/proxy enable`	(Enable proxy on the port)
`>> Proxy IP address /cfg/slb/real 1/adv/proxy dis`	(Disable local real server proxy)
`>> Real server 1# /cfg/slb/real 2/adv/proxy dis`	(Disable proxy for the local server)
`>> Real server 2# /cfg/slb/real 3/adv/proxy ena`	(Enable proxy for the remote server)
`>> Real server 3# apply`	(Apply configuration changes)
`>> Real server 3# save`	(Save configuration changes)

For more information on proxy IP addresses, see Client Network Address Translation (Proxy IP), page 190.

2. If you want to configure proxy IP addresses on Site 2, issue the following commands on the Denver Alteon:

`>> # /cfg/slb/pip`	(Select *Proxy IP Address* menu)
`>> Proxy IP address# type port`	(Use port-based proxy IP)
`>> Proxy IP address# add 174.14.70.4`	(Set unique proxy IP address)
`>> # /cfg/slb/port 4/adv/proxy enable`	(Enable proxy on the port)
`>> Proxy IP address# /cfg/slb/real 1/adv/proxy dis`	(Disable local real server proxy)
`>> Real server 1# /cfg/slb/real 2/adv/proxy dis`	(Disable local real server proxy)
`>> Real server 2# /cfg/slb/real 3/adv/proxy ena`	(Enable proxy for the remote server)
`>> Real server 3# apply`	(Apply configuration changes)
`>> Real server 3# save`	(Save configuration changes)

Configuring GSLB Behind a NAT Device

Two Alteons, each behind a separate NAT device, connect using the IP address of each other's NAT device for DSSP communication. When an Alteon performs DNS resolution, the DNS response must include the public (NAT) address of the service, not the internal virtual IP address. When Alteons are installed between NAT devices:

- Alteon must be aware of the public (NAT) address for each of its virtual IP addresses.
- The remote real server must always be configured using public (NAT) addresses.

Figure 128 - Network with GSLB Configuration Behind NAT Devices, page 757 illustrates a configuration where Alteons at Sites A and B are located behind NAT devices, and Alteon at Site C is not.

Figure 128: Network with GSLB Configuration Behind NAT Devices

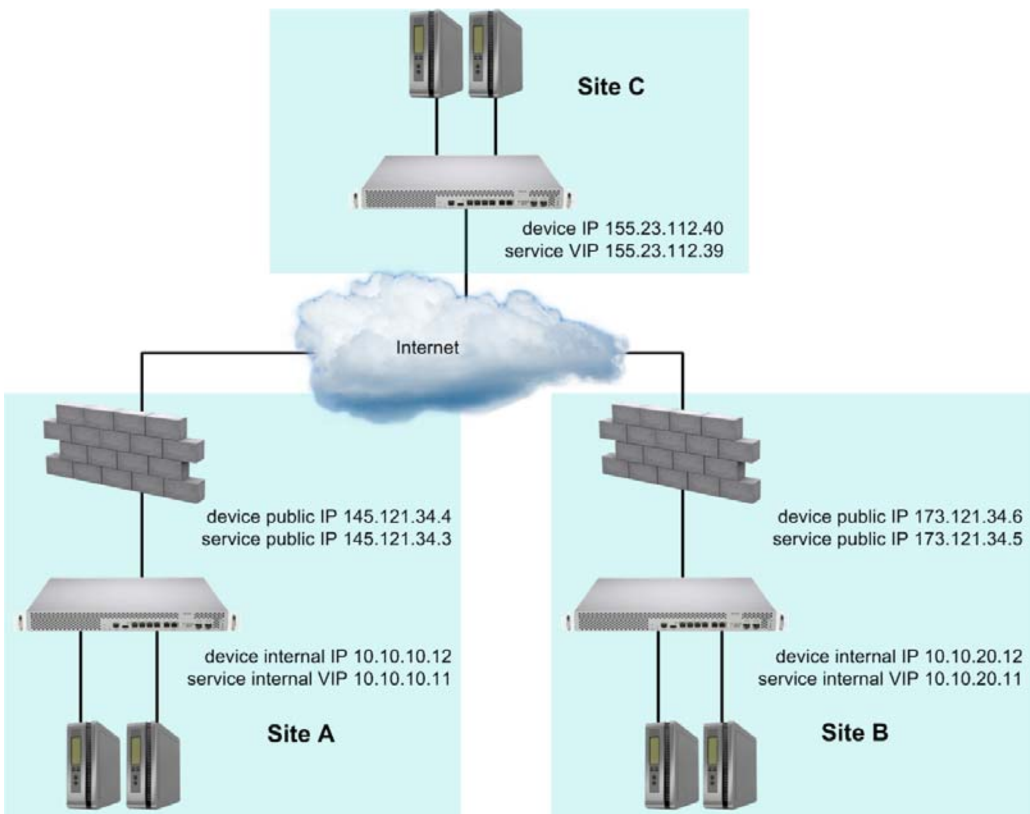


Table 65 summarizes the network configuration.

Table 65: GSLB Configuration Behind NAT Devices

IP Address Type	Site A	Site B	Site C
Alteon internal IP	10.10.10.12	10.10.20.12	155.23.112.40
Remote sites	173.121.34.6 (site B Alteon public IP) 155.23.112.40 (site C Alteon IP)	145.121.34.4 (site A Alteon public IP) 155.23.112.40 (site C Alteon IP)	145.121.34.4 (site A Alteon public IP) 173.121.34.6 (site B Alteon public IP)
Service VIP	10.10.10.11	10.10.20.11	155.23.112.39
Service public IP (NAT)	145.121.34.3	173.121.34.5	

Table 65: GSLB Configuration Behind NAT Devices (cont.)

IP Address Type	Site A	Site B	Site C
Remote servers	173.121.34.5 (site B service public IP) 155.23.112.39 (site C service IP)	145.121.34.3 (site A service public IP) 155.23.112.39 (site C service IP)	145.121.34.3 (site A service public IP) 173.121.34.5 (site B service public IP)
Alteon public IP (NAT)—no need to configure	145.121.34.4	173.121.34.6	

To add a NAT device IPv4 address to an Alteon server

1. Set the network preference to IPv4.

```
>> # /cfg/slb/virt 1/ipver v4
```

2. Add the service public IP address (NAT) of the device to the Alteon server.

```
>> # /cfg/slb/virt 1/nat
>> Virtual Server 1# nat
Current NAT IP address: 0.0.0.0
Enter new NAT IP address: 145.121.34.3
```

To add a NAT device IPv6 address to an Alteon server

1. Set the network preference to IPv6.

```
>> # /cfg/slb/virt 1/ipver v6
```

2. Add the service public IP address (NAT) of the device to the Alteon server.

```
>> # /cfg/slb/virt 1/nat
>> Virtual Server 1# nat
Current NAT IP6 address: 0:0:0:0:0:0:0:0
Enter new NAT IP6 address: 173.121.34.5
```

Using Border Gateway Protocol for GSLB

The Border Gateway Protocol (BGP)-based GSLB uses the Internet's routing protocols to localize content delivery to the most efficient and consistent site. This is done by using a shared IP block that co-exists in each Internet Service Provider (ISP) network and is then advertised, using BGP, throughout the Internet.

Because of the way IP routing works, BGP-based GSLB allows routing protocols to route DNS requests to the closest location, which then return IP addresses of that particular site, locking in the requests to that site. In effect, the Internet is making the decision of the best location for you, avoiding the need for advanced GSLB.

Some corporations use more than one ISP as a way to increase the reliability and bandwidth of their Internet connection. Enterprises with more than one ISP are referred to as being *multihomed*. Instead of multihoming a network to several other networks, BGP-based GSLB enables you to multihome a particular piece of content (DNS information) to the Internet by distributing the IP blocks that contain that content to several sites.

When using DNS to select the site, a single packet is used to make the decision so that the request is not split to different locations. Through the response to the DNS packet, a client is locked into a particular site, resulting in efficient, consistent service that cannot be achieved when the content itself is shared.

For example, in multihoming, you can connect a data center to three different Internet providers and have one DNS server that has the IP address 1.1.1.1. In this case, all requests can be received via any given feed but are funneled to the same server on the local network. In BGP-based GSLB, the DNS server (with the IP address 1.1.1.1) is duplicated and placed local to the *peering point* instead of having a local network direct traffic to one server.

When a particular DNS server receives a request for a record (in this case, Alteon), it returns with the IP address of a virtual server at the same site. This can done using the **`local`** option (**`/cfg/slb/gslb/rule 1/metric 2/gmetric local`**) in the GSLB configuration. If one site is saturated, then Alteon will fail over and deliver the IP address of a more available site.

For more information on configuring Alteon to support BGP routing, see Border Gateway Protocol, page 125.

Verifying GSLB Operation

The following procedure is for verifying GSLB operations.

To verify GSLB operation

1. Use your browser to request the configured service (www.gslb.example.com in Figure 119 - DNS Resolution with GSLB, page 707).
2. Examine the **`/info/slb`** and **`/info/slb/gslb`** information on each Alteon.
3. Check to see that all SLB and GSLB parameters are working as expected. If necessary, make any appropriate configuration changes and then check the information again.
4. Examine the **`/stats/slb`** and **`/stats/slb/gslb`** statistics on each Alteon.

Chapter 26 – Bandwidth Management

Bandwidth Management (BWM) enables Web site managers to allocate a portion of the available bandwidth for specific users or applications. It allows companies to guarantee that critical business traffic, such as e-commerce transactions, receive higher priority versus non-critical traffic. Traffic classification can be based on user or application information. BWM policies can be configured to set lower and upper bounds on the bandwidth allocation.

The following topics are discussed in this chapter:

Using Bandwidth Management

To use the BWM features, you must purchase an additional software license and license string. Contact Radware Technical Support for additional software licenses.

There are two operational license strings for BWM: standard and demo. The demo license automatically expires after a set time period. These license strings may only be enabled if Layer 4 services have been enabled.

Once you have obtained the proper license string to enable BWM, do the following:

1. Connect to the CLI via Telnet or the console port, and log in as the administrator, following the directions in the “Command Line Interface” chapter of the *Alteon Application Switch Operating System Command Reference.*
2. From the CLI, enter the `/oper/swkey` command.

   You are prompted to enter the license string. If it is correct for this MAC address, Alteon accepts the password, permanently records it in non-volatile RAM (NVRAM), and then enables the feature.

Contracts

A contract is created to assign a certain amount of bandwidth for an application. Up to 1024 contracts can be configured on a single Alteon. Alteon uses these contracts to limit individual traffic flows, and can be enabled or disabled as necessary. Contracts can be assigned to different types of traffic, based on whether it is Layer 2, Layer 4, or Layer 7 traffic, as well as by port, VLAN, trunk, filters, virtual IP address, service on the virtual server, URL, and so on. Any item that is configured with a filter can be used for BWM.

Bandwidth classification is performed using the following menus:

- **`/cfg/slb/filt`**— Used to configure classifications based on the IP destination address, IP source address, TCP port number, UDP, UDP port number, 802.1p priority value, or any filter rule.

- **/cfg/slb/virt**—Used to configure classifications based on virtual servers.
- **/cfg/port**—Used to configure classifications based on physical ports.

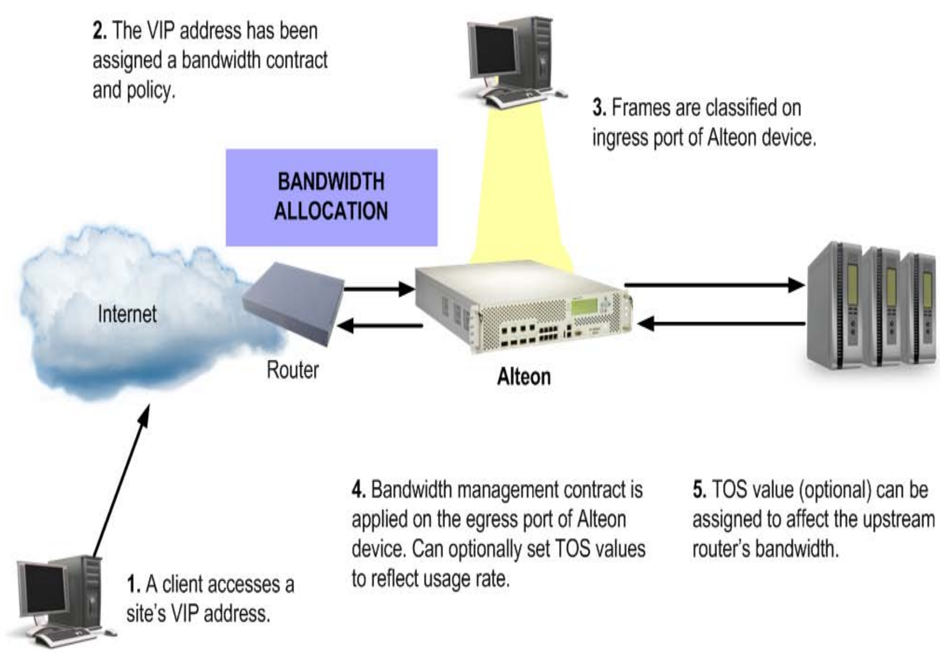
Note: For trunking, use **/cfg/l2/trunk**.

- **/cfg/l2/vlan**—Used to configure classifications based on VLANs.
- **/cfg/slb/layer7/lb**—Used to configure classification based on URL paths.
- **/info/bwm**—Used to display the set of classifications associated with each contract.

To associate a particular classification with a contract, enter the contract index into the **cont** menu option under the applicable configuration menus.

As illustrated in Figure 129 - How Bandwidth Management Works, page 762, when the Virtual Matrix Architecture (VMA) is enabled, traffic classification is performed on the *ingress port* (the port on which the frame is received), and *not* the client port or the server port. If the traffic classification is performed on Layer 4 through Layer 7 traffic (filter-based or SLB traffic), then the classification occurs on the *designated port*.

Figure 129: How Bandwidth Management Works



Classification Rules

In a classification rule, certain frames are grouped together. For frames qualifying for multiple classifications, the contract precedence is also specified per contract. If no precedence is specified, the default order is used (see Classification Precedence, page 763).

The following classifications limit the traffic outbound from the server farm for bandwidth measurement and control:

- **Physical Port**—All frames are from a specified physical port.
- **VLAN**—All frames are from a specified VLAN. If a VLAN translation occurs, the bandwidth policy is based on the ingress VLAN.

- **IP Source Address**—All frames have a specified IP source address or range of addresses defined with a subnet mask.
- **IP Destination Address**—All frames have a specified IP destination address or range of addresses defined with a subnet mask.
- Switch services on the virtual servers.

The following are various Layer 4 groupings:

- A single virtual server
- A group of virtual servers
- A service for a particular virtual server
- A particular port number (service on the virtual server) within a particular virtual server IP address

The following are various Layer 7 groupings:

- A single URL path
- A group of URL paths
- A single cookie

Classification Precedence

There are two mechanisms for frames that qualify for classifications: a per-contract precedence value and a default precedence ordering from 1 to 255, where the higher numbers have the higher precedence. If a contract does not have an assigned precedence value, then the default ordering is applied as follows:

1. Incoming source port/default assignment
2. VLAN
3. Filter
4. Layer 4 services on the virtual server
5. Layer 7 applications (for example, URL, HTTP, headers, cookies, and so on)

If a frame falls into all of classifications (1 through 5), and if the precedence is same for all the applicable contracts, then the Layer 7 applications contract classification (precedence level 5) is assigned because it comes last and has the highest precedence.

Application Bandwidth Control

Classification policies allow bandwidth limitations to be applied to particular applications, meaning that they allow applications to be identified and grouped. Classification can be based on any filtering rule, including the following:

- Layer 7 strings—Strings that identify to which application the traffic belongs.
- TCP Port Number—All frames with a specific TCP source or destination port number.
- UDP—All UDP frames.
- UDP Port Number—All frames with a specific UDP source or destination port number.

Combinations

Combinations of classifications are limited to grouping items together into a contract. For example, if you want to have three different virtual servers associated with a contract, you specify the same contract index on each of the three virtual server IP addresses. You can also combine filters in this manner. Combinations are described further in the following sections:

- Grouped Bandwidth Contracts, page 764—Describes how contracts can be grouped together to aggregate BMW resources.
- IP User Level Contracts for Individual Sessions, page 765—Describes a user-level contract.

Grouped Bandwidth Contracts

Alteon uses the concept of multi-tiered, or grouped, bandwidth management contracts. In earlier releases, a single-level bandwidth management contract was used to manage bandwidth on an Alteon. BWM contract groups are now configured to aggregate contract resources and share unused bandwidth within the contract group. A group level contract should contain two or more individual contracts as defined in Contracts, page 761.

Based on how much traffic is sent in each contract in the group, the hard limits of the contracts are adjusted proportionately to their share in the group.

Example Grouped Bandwidth Contract

A group level contract is configured with four individual contracts with rate limits of 10, 20, 30 and 40 Mbps each. Together, the total rate limit of the member contracts is 100 Mbps. If a particular contract is not using its full bandwidth allocation, Alteon reallocates the bandwidth to the other members of the contract group by polling bandwidth statistics every second, and recalculating the bandwidth allocation.

Table 66 - Bandwidth Reallocation in Grouped Contracts, page 765 illustrates how the hard limits of individual contracts self-adjust when placed into a contract group. The hard limit indicates the actual hard limits set for each individual contract. Since contracts 1 through 4 are part of a contract group, the total hard limit allowed for the group in this example is 100 Mbps.

The actual traffic indicates that contracts 1 and 4 have exceeded their hard limits by a total of 25 Mbps. Contract 3 is underusing its hard limit by 10 Mbps.

Because all contracts are members of the group, the unused bandwidth is divided proportionately between the two contracts that exceeded their hard limits—contracts 1 and 4.

- Contract 1 requests 15 Mbps, which is 5 Mbps over its hard limit. Because contract 1 requests 5 of the 25 Mbps bandwidth over the total bandwidth hard limit for the contract group, it receives one-fifth of the available extra share, or 2 Mbps. The remaining 3 Mbps that contract 1 requests is dropped.
- Contract 4 requests 60 Mbps, which is 20 Mbps over its hard limit. Because contract 4 requests 20 of the 25 Mbps over the total bandwidth hard limit for the contract group, it receives four-fifths of the extra share, or 12 Mbps. The remaining 12 Mbps requested by contract 4 is dropped.

Table 66: Bandwidth Reallocation in Grouped Contracts

Resource	Contract 1	Contract 2	Contract 3	Contract 4	Total
Hard limit	10[1]	20	30	40	100
Actual traffic	15	20	20	60	115
Unused bandwidth	NA	NA	10	NA	10
Bandwidth over Hard	5	0	NA	20	25
Extra share	[2] $\frac{5}{25}x10 = 2^{a}$	0	NA	[3] $\frac{20}{25}x10 = 8^{b}$	10
Adjusted hard limit	12	20	20	48	100

1 – (All units in Mbps)
2 – Denotes the bandwidth over the hard limit in contract 1, divided by the total bandwidth over the hard limit for the contract group, multiplied by the total extra share bandwidth.
3 – Denotes the bandwidth over the hard limit in contract 4, divided by the total bandwidth over the hard limit for the contract group, multiplied by the total extra share bandwidth

Note: The soft and reserved, or Committed Information Rate (CIR), limits of each contract are not part of the grouped contract's calculation, and remain set at their individual contract's levels.

For a group contract configuration example, see Configuring Grouped Contracts for Bandwidth Sharing, page 778.

IP User Level Contracts for Individual Sessions

Bandwidth Management includes *user limits*, which are policies that can be applied to a contract that specify a rate limit for each user who is sending or receiving traffic in that contract. The contract can be configured to identify a user by either the source or the destination IP address in the packets.

The user limit policy monitors the amount of bandwidth used per second, and drops any traffic that exceeds the configured limit. To monitor a user's bandwidth, Alteon creates an IP user entry that records the source or destination IP address, and the amount of bandwidth used.

This feature is used to limiting bandwidth hogging by a few overactive internet users with unimportant traffic (for example peer-to-peer movie sharing), which may end up denying other users with legitimate traffic from their fair share of the bandwidth. Because user limiting is performed on a per-contract basis, different types of traffic can be classified into different contracts and can have different user limits applied according to the class of traffic. Because user limiting for a contract is optional, it can be set for contracts where fair-sharing of bandwidth is important, and not set for the contracts where fair-sharing of bandwidth is not important or desirable.

The following are examples that further explain how user limits work:

Example User Limits are Overwritten by the Contract Hard Limit

The IP user limit is configured in addition to the contract's hard limit. However, the contract's hard limit overrides the individual user entry's user limit.

An example contract has a hard limit of 10 Mbps and a user limit of 1 Mbps. If there are 20 IP users for the contract with an offered traffic rate of 1 Mbps each (for a total offered traffic rate for the contract of 20 Mbps), the total traffic allowed for the contract does not exceed the hard limit (10

Mbps). Therefore, even though the individual IP user limits do not exceed their 1 Mbps hard limit, some or all of the IP users may have some traffic dropped because the contract's hard limit (10 Mbps) is less than the total of the offered traffic rate for all 20 users (20 Mbps).

Example User Limits are Maintained When a Contract has Available Bandwidth

An example contract has a hard limit of 10 Mbps and a user limit of 1 Mbps. There are two IP users for the contract, with an offered traffic rate of 5 Mbps each (for a total offered traffic rate for the contract of 10 Mbps). Even though the offered traffic rate for the whole contract does not exceed the hard limit, Alteon limits the traffic for both the IP users to their user limits (1 Mbps each).

The user limit configured for a contract is the limit for one egress Switch Processor (SP) rather than the entire Alteon. For example, if a contract is configured for a user limit of 64 kbps, and traffic for a user (IP address) is egressing port 1 (SP 1) and port 20 (SP 2), that user (IP address) is restricted to 64 kbps egressing on port 1 and 64 kbps egressing out on port 20.

For an example, see Configuring an IP User-Level Rate Limiting Contract, page 780.

Policies

Bandwidth policies are bandwidth limitations defined for any set of frames, that specify the maximum, best effort, and minimum guaranteed bandwidth rates. A bandwidth policy is assigned to one or more contracts. You can define up to 64 bandwidth policies.

A bandwidth policy is often based on a rate structure where a Web host or co-location provider could charge a customer for bandwidth usage. There are three rates that are configured:

- Committed Information Rate (CIR)/Reserved Limit
- Soft Limit
- Hard Limit

Bandwidth limits are usually entered in Mbps. For better granularity, rates can be entered in kbps by appending **k** to the entered number. For example, 1 Mbps can be entered as either **1** or as **1024k**.

Bandwidth Policy Index

Each BWM contract is assigned a bandwidth policy index and, optionally, a name. You can display this index using the **/cfg/bwm/cont** menu.

Bandwidth Queue Size

A queue size is associated with each policy. The queue size is measured in bytes.

Time Policy

A BWM contract can be configured to apply different time policies defined by ranges of hours or days of the week. The time policy is based on the time set in Alteon's system clock (see **/info/sys/general**).

Configuring Time and Day Policies, page 791 describes how to configure and apply policies to different times and days.

Enforcing Policies

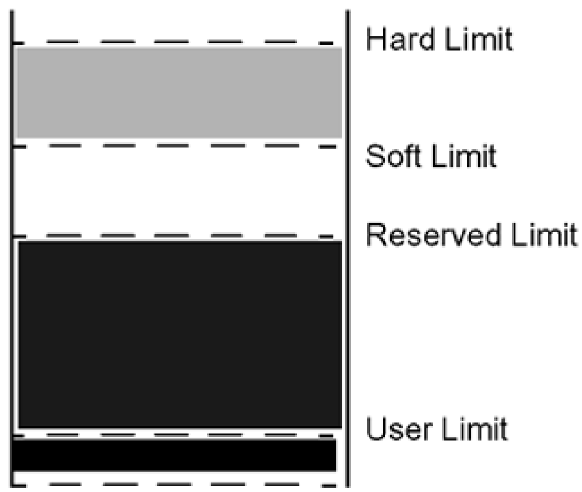
For BWM contracts and policies to take effect, the policies must be enforced using the **`/cfg/bwm/force ena`** command.

Even when BWM is not enforced, Alteon can still collect classification information and report it, allowing an administrator to observe a network before deciding how to configure it. This feature can be disabled using **`/cfg/bwm/force dis`**. When this command is used, no limits will be applied on any contract.

Rate Limiting

A rate limiting contract is controlled by metering the traffic that egresses from Alteon. If the egress rate is below the configured rate limit (hard limit) for the port, the traffic is transmitted immediately without any buffering. If the egress rate is above the configured rate limit the traffic above the rate limit is dropped. This is illustrated in Figure 130 - Bandwidth Rate Limits, page 767.

Figure 130: Bandwidth Rate Limits



For rate limiting contracts, the queue depth is ignored because traffic is not buffered.

Typically, bandwidth management occurs on the Alteon egress port, meaning the port from which the frame is leaving. However, when there are multiple routes or trunk groups, the egress port can actually be one of several ports (from the point-of-view of where the queues are located).

A bandwidth policy specifies four limits, listed and described in Table 67 - Bandwidth Rate Limits, page 768:

Table 67: Bandwidth Rate Limits

Rate Limit	Description
Committed Information Rate (CIR) or reserved limit	This is a rate that a bandwidth classification is always guaranteed. In configuring BWM contracts, ensure that the sum of all committed information rates never exceeds the link speeds associated with ports on which the traffic is transmitted. If the total CIRs exceed the outbound port bandwidth, Alteon performs a graceful degradation of all traffic on the associated ports.
Soft limit	For traffic shaping contracts, this is the desired bandwidth rate—that is, the rate the customer has agreed to pay on a regular basis. When output bandwidth is available, a bandwidth class is allowed to send data at this rate. No exceptional condition is reported when the data rate does not exceed this limit. For rate limiting contracts, the soft limit is ignored.
Hard limit	This is a "never exceed" rate. A bandwidth class is never allowed to transmit above this rate. Typically, traffic bursts between the soft limit and the hard limit are charged a premium. The maximum hard limit for a bandwidth policy is 1 Gbps, even when multiple gigabit ports are trunked. To ensure a specific amount of throughput on a port, configure hard and soft limits close together. For example, to ensure 20 Mbps of throughput on a 100 Mbps port, create a policy on a contract that sets the hard limit to 20M and the soft limit to 19M. If you apply this contract to a filter on the egress port, 20 Mbps of throughput can be ensured.
User limit	A user limit is a hard limit rate for individual users. It is defined as a policy and is applied and enabled for an individual contract. It is based on either a source IP or destination IP address. Setting user limits requires that a contract be configured that enables IP limiting (**`/cfg/bwm/cont <x> /iplimit ena`**), and sets the type of limiting to source IP or destination IP address (**`/cfg/bwm/cont <x> /iptype {sip|dip}`**). When configured, an individual IP address can be limited to traffic between 0 kbps and 1000 Mbps. A user limit based on source IP address should be set if the goal is to limit the amount of data being transmitted from a source IP address in your network. A user limit based on the destination IP address should be set if the goal is to limit the amount of data being downloaded from a destination IP address in your network.

Application Session Capping

Application session capping is a feature that allows limits to be placed on the number of sessions on a user per contract or per contract basis. This results in bandwidth contracts having an additional maximum sessions parameter that will define the upper limit at which the application will be capped.

Note: Session capping per contract is applied on a per SP basis. Session capping per-user is applied on a per-Alteon basis.

Application session capping is applied in the following ways:

- **Contract Capping**—Session capping per contract is applied per SP.
- **User Capping**—Session capping per user is applied.

Application session capping is especially relevant in today's world of peer-to-peer applications that require a large amount of network bandwidth. It enables the administrator to cap the number of sessions of an application assigned to each user. In this way, peer-to-peer (and other such non-business applications) can be limited or completely eliminated on the network.

Note: For the purposes of this feature, a user is defined as a unique source IP address and the application is identified based on a bandwidth contract

Application session capping functions by creating an entry in the session table that designates the contract/user combination. Whenever a new session is created, this entry is checked against existing sessions in the session table and, if a match is made, the maximum sessions value is queried. If the maximum sessions value has been reached, the new session is dropped. If the value has not been reached, the session count is incremented and the session is allowed to continue.

Notes

- Application session capping is not supported when a contract is assigned to a port, VLAN, trunk, or virtual service.
- Application session capping does not support an iplimit contract based on DIP. It does, however, support an iplimit contract based on SIP.

Rate Limiting Timeslots

For rate limiting contracts, metering of individual traffic flows is done using several timeslots per second. The timeslot traffic limit is the traffic that is sent for a particular contract for every timeslot corresponding to the contract's rate limit, or the hard limit as initially calculated.

For any contract there is one timeslot traffic limit for each egress port. The timeslot traffic limit is calculated from the hard limit. The timeslot traffic limit is the amount of traffic that corresponds to the hard limit per second, divided by the number of timeslots per second.

Traffic is transmitted for every timeslot as long as the traffic is below the timeslot traffic limit for the contract. Any traffic that exceeds the timeslot traffic is discarded.

Traffic Shaping

A traffic shaping contract establishes queues and schedules when frames are sent from each queue. Traffic is shaped by pacing the packets according to the hard, soft, and reserve limits. Each frame is put into a managed buffer and placed on a contract queue. The time that the next frame is supposed to be transmitted for the contract queue is calculated according to the configured rate of the contract, the current egress rate of the ports, and the buffer size set for the contract queue. The scheduler then organizes all the frames to be sent according to their time-based ordering and meters them out to the port.

When packets in a contract queue have not yet been sent and the buffer size set for the queue is full, any new frames attempting to be placed in the queue are discarded.

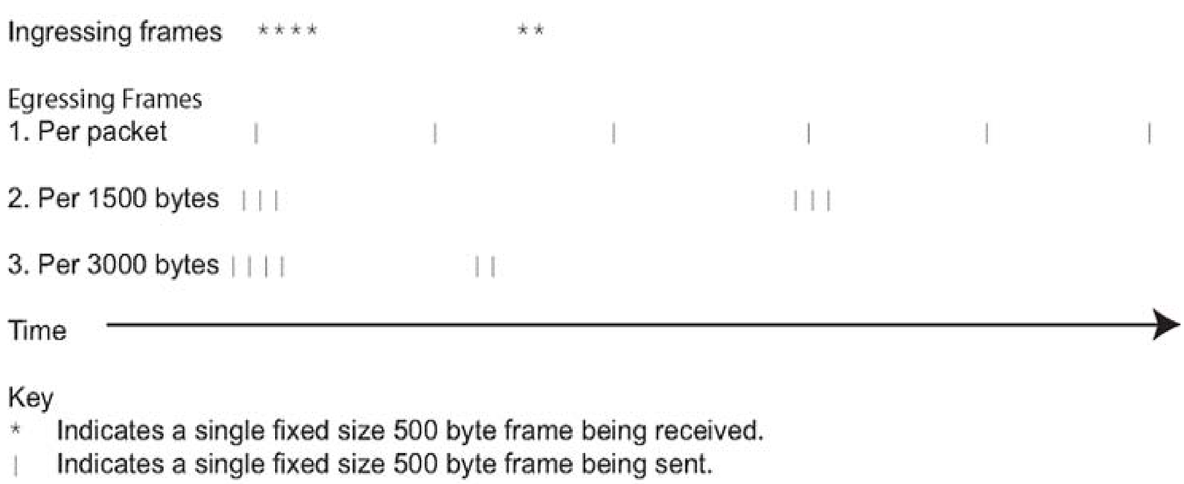
For traffic shaping contracts, a queue depth is also associated with a policy. A queue depth is the size of the queue that holds the data. It can be adjusted to accommodate delay-sensitive traffic (such as audio) versus drop-sensitive traffic (such as FTP).

Data Pacing for Traffic Shaping Contracts

The mechanism used to keep the individual traffic flows under control in a traffic shaping contract is called *data pacing*. It is based on the concept of a real-time clock and theoretical departure times (TDT). The actual calculation of the TDT is based initially on the configured soft limit rate. The soft limit can be thought of as a target limit for the ISP's customer. As long as bandwidth is available and the classification queue is not being filled at a rate greater than the soft limit, the TDT is met for both incoming frames and outgoing frames, and no borrowing or bandwidth limitation is necessary. If the classification queue exceeds the soft limit, a frame is queued for transmittal and the TDT is increased by the size of the frame multiplied by the transmittal rate of the queue.

Figure 131 - Real-time Clocks and Theoretical Departure Times, page 770 illustrates how data may be paced in a traffic shaping contract. Six arriving frames are processed differently depending on rate of the queue. Queue 1 processes each packet evenly. Queue 2 processes per 1500 bytes and inserts some delay as it processes the first three 500 byte frames and then the next three frames. Queue 3 processes at 3000 bytes per second and has ample capacity to process egress frames at the same rate as the ingress frames.

Figure 131: Real-time Clocks and Theoretical Departure Times



If the data is arriving more quickly than it can be transmitted at the soft limit, and sufficient bandwidth is still available, the rate is adjusted upward based on the depth of the queue, until the rate is fast enough to reduce the queue depth or the hard limit is reached. If the data cannot be transmitted at the soft limit, then the rate is adjusted downward until the data can be transmitted or the CIR is hit. If the CIR is overcommitted among all the contracts configured for Alteon, graceful degradation reduces each CIR until the total bandwidth allocated fits within the total bandwidth available.

Bandwidth Management Information

Statistics are stored in the individual Switch Processors (SP) and then collected every second by the MP (Management Processor). The MP combines the statistics, as statistics for some classifications may be spread across multiple SPs.

Viewing BWM Statistics

The **`/stats/bwm/dump`** command displays the total number of octets sent, octet discards, and times over the soft limit are kept, for each contract. The history buffer maintains the average queue size for the time interval and the average rate for the interval.

Packet counters also maintain bandwidth management statistics for packets on a per-contract basis as well as calculation of the average packet size.

Configuring BWM History

History is maintained only for the contracts for which the history option is enabled, using the **`/cfg/bwm/cont x/hist`** command.

Sending BWM History

The MP maintains global statistics, such as total octets, and a window of historical statistics. When the history buffer of 128K is ready to over flow, it can be sent from Alteon using either an e-mail or direct socket transfer mechanism.

To configure sending Bandwidth Management statistics

1. Select the statistics delivery method. Bandwidth Management statistics can be sent through e-mail or by socket to a reporting server.
   - To send BWM statistics through e-mail, issue this command:

```
>> Main# /cfg/bwm/email enable
```

   - To send BWM statistics by socket to a reporting server, issue the following commands:

`>> Main# /cfg/bwm/email/ disable`	(E-Mail statistics delivery must be disabled)
`>> Main# /cfg/bwm/report <Reporting Server IP Address>`	

   BWM statistics are sent to TCP port 49152 of the specified reporting server.

2. Configure the selected delivery method.
   - To configure e-mail usage, issue these commands:

```
>> Main# /cfg/bwm/user <SMTP User Name>
>> Main# /cfg/sys/smtp <SMTP host name or IP address>
```

— To configure socket delivery usage, issue the following command:

`>> Main# /cfg/sys/mmgmt/report {mgmt | data}`	(Select to use the management or data port to communicate with the reporting server).

Note: To obtain graphs with this data, the data must be collected and processed by an external entity through SNMP.

Statistics and Management Information Bases

- **For existing BWM classes**—The MP maintains per-contract rate usage statistics. These are obtainable via a private MIB.
- **When BWM services are not enabled**—Even when BWM is not enforced, the MP can still collect classification information and report it, allowing an administrator to observe a network for a while before deciding how to configure it. This feature can be disabled using **`/cfg/bwm/force dis`**. When this command is used, no limits are applied on any contract.

Synchronizing BWM Configurations in VRRP

BWM configurations are optionally synchronized to a backup Alteon during VRRP synchronization. However, port contracts and VLAN contracts are not synchronized. For more information on VRRP and synchronized configurations, see Configuring VRRP Peers for Synchronization, page 569.

Packet Coloring (TOS bits) for Burst Limit

Whenever the soft limit is exceeded, optional packet coloring can be done to allow downstream routers to use *diff-serv* mechanisms (that is, writing the Type-Of-Service (TOS) byte of the IP header) to delay or discard these *out-of-profile* frames. Frames that are not out-of-profile are marked with a different, higher priority value. This feature can be enabled or disabled on a per-contract basis, using the **`wtos`** option under the contract menu (**`/cfg/bwm/cont <x> /wtos`**) to enable/disable overwriting IP TOS.

The actual values used by Alteon for overwriting TOS values (depending on whether traffic is over or under the soft TOS limit) are set in the bandwidth policy menu (**`/cfg/bwm/pol <x>`**) with the **`utos`** and **`otos`** options. The values allowed are 0 through 255. Typically, the values specified should match the appropriate diff-serv specification, but can be different, depending on the customer environment.

Contract-Based Packet Mirroring

Contract-based packet mirroring allows an egress packet that matches a contract to be mirrored to a specified port. This feature can be used for troubleshooting and analysis as well as a tool to identify new signatures for Internet Traffic Management (ITM) functionality.

You enable packet mirroring on a contract by configuring a valid mirroring port. When a packet is classified, if a mirroring port is configured for that contract, a copy of the packet is mirrored to the configured port. The packet is mirrored at the egress port after all modifications are made to the packet.

Note: This feature is available in *maintenance mode* only.

To set a mirroring port for a contract

```
>> Main# /cfg/bwm/cont <contract number> /pmirr <port>
```

To disable a mirroring port on a contract

```
>> Main# /cfg/bwm/cont <contract number> /pmirr none
```

Note: Mirroring occurs before the application of the limiting contract. Packets that would have been otherwise discarded by the contract are also copied to the mirroring port.

Configuring Bandwidth Management

The following procedure provides general instructions for configuring BWM on Alteon. Specific configuration examples begin on Additional BWM Configuration Examples, page 776.

To configure Bandwidth Management

1. Configure Alteon as you normally would for SLB. Configuration includes the following tasks:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface.
   - Define each real server.
   - Define a real server group.
   - Define a virtual server.
   - Define the port configuration.

For more information about SLB configuration, see Server Load Balancing, page 165.

2. Enable BWM.

```
>># /cfg/bwm/on
```

Note: If you purchased the Bandwidth Management option, be sure to enable it by typing **/oper/swkey** and entering the license string. For more information, see Using Bandwidth Management, page 761.

3. Select a bandwidth policy. Each policy must have a unique number from 1 to 64.

```
>> Bandwidth Management # pol 1
```

4. Set the hard, soft, and reserved rate limits for the policy, in Mbps.

   Typically, charges are applied for burst rates between the soft and hard limit. Each limit must be set between 64K and 1000M.

   Note: For rates less than 1 Mbps, append a **k** suffix to the number.

```
>> Policy 1# hard 6                     (Set "never exceed" rate)
>> Policy 1# soft 5                     (Set desired bandwidth rate)
>> Policy 1# resv 4                     (Set committed information rate)
```

5. Optionally, set the Type-Of-Service (TOS) byte value, between 0 and 255, for the policy underlimit and overlimit.

   There are two parameters for specifying the TOS bits: underlimit (**utos**) and overlimit (**otos**). These TOS values are used to overwrite the TOS values of IP packets if the traffic for a contract is under or over the soft limit, respectively. These values only have significance to a contract if TOS overwrite is enabled in the *Bandwidth Management Contract* menu (**/cfg/bwm/cont <x> /wtos ena**).

   Note: You should use care when selecting the TOS values because of their greater impact on the downstream routers.

```
>> Policy 1# utos 204                   (Set BWM policy underlimit)
>> Policy 1# otos 192                   (Set BWM policy overlimit)
```

6. Set the buffer limit for the policy. Set a value between 8192 and 128000 bytes. The buffer depth for a BWM contract should be set to a multiple of the packet size.

   Note: The total buffer limit for the Bandwidth Management policy is 128K.

```
>> Policy 1# buffer 16320
```

7. On Alteon, select a BWM contract and, optionally, a name for the contract. Each contract must have a unique number from 1 to 256.

```
>> Policy 1# /cfg/bwm/cont 1
>> BWM Contract 1# name BigCorp
```

8. Optionally, set a precedence value for the BWM contract.

   Each contract can be given a precedence value from 1 to 255. The higher the number, the higher the precedence. If a frame is applicable to different classifications, then the contract with the higher precedence is assigned to the frame. If the precedence is the same for the applicable contracts, then the following order will be used to assign the contract to the frame:

   a. Incoming port
   b. VLAN
   c. Filter
   d. Service on the virtual server
   e. URL/cookie

```
>> BWM Contract 1# prec 1
```

9. Optionally, enable TOS overwriting for the BWM contract.

```
>> BWM Contract 1# wtos ena
```

10. Set the bandwidth policy for this contract. Each bandwidth management contract must be assigned a bandwidth policy.

```
>> BWM Contract 1# pol 1
```

11. Optionally, enable traffic shaping. Rate limiting is enabled by default. Enabling traffic shaping disables rate limiting. For more information, see Traffic Shaping, page 769.

```
>> BWM Contract 1# shaping e
```

12. Enable the BWM contract.

```
>> BWM Contract 1# ena
```

13. Classify the frames for this contract and assign the BWM contract to the filter or virtual IP address.

    Each BWM contract must be assigned a classification rule. The classification can be based on a filter or services on the virtual server. Filters are used to create classification policies based on the IP source address, IP destination address, TCP port number, UDP, and UDP port number.

    In this case, all frames that match filter 1 or Virtual Server 1 will be assigned Contract 1.

```
>> BWM Contract 1# /cfg/slb/virt 1/cont 1
>> Virtual Server 1# /cfg/slb/filt 1/adv/cont 1
```

14. On Alteon, apply and verify the configuration.

```
>> Filter 1 Advanced# apply
>> Filter 1 Advanced# /cfg/bwm/cur
```

    Examine the resulting information. If any settings are incorrect, make any appropriate changes.

15. On Alteon, save your new configuration changes.

```
>> Bandwidth Management# save
```

16. On Alteon, check the BWM information.

```
>> Bandwidth Management# /info/bwm <contract number>     (View BWM information)
>> Bandwidth Management# /stats/bwm <contract number>    (View BWM statistics)
```

Check that all BWM contract parameters are set correctly. If necessary, make any appropriate configuration changes and then check the information again.

Additional BWM Configuration Examples

The following examples are provided for the following Bandwidth Management applications:

- Configuring User/Application Fairness, page 776
- Configuring Grouped Contracts for Bandwidth Sharing, page 778
- Configuring an IP User-Level Rate Limiting Contract, page 780
- Configuring BWM Preferential Services, page 781
- Configuring Content-Intelligent Bandwidth Management, page 783
- Configuring Cookie-Based Bandwidth Management, page 786
- Configuring Security Management, page 789
- Configuring Time and Day Policies, page 791
- Egress Bandwidth Tuning for Lower Speed Networks, page 792
- Overwriting the TCP Window Size, page 793

Note: Ensure BWM is enabled on Alteon (**`/cfg/bwm/on`**).

Example Configuring User/Application Fairness

Bandwidth Management can be applied to prevent heavy bandwidth bursters from locking out other users, such as the following:

- Customers using broadband access (such as DSL) blocking dial-up customers.
- Customers using the same hosting facility locking out each other because of a flash crowd.
- FTP locking out Telnet.
- Rate limits of particular applications.

In this example, BWM is configured to prevent broadband customers from affecting dial-up customer access. This is accomplished by setting higher bandwidth policy rate limits for the port that processes broadband traffic.

- Policy 1 is for dial-up customers with lower bandwidth allocation needs.

- Policy 2 is for broadband customers with higher bandwidth allocation needs.

1. Select the first bandwidth policy for dialup customers. Each policy must have a number from 1 to 512. Ensure BWM is enabled on Alteon (**/cfg/bwm/on**).

```
>> # /cfg/bwm/pol 1
```

2. Set the hard, soft, and reserved rate limits for the bandwidth policy, in Mbps.

```
>> Policy 1# hard 5                (Set "never exceed" rate)
>> Policy 1# soft 4                (Set desired bandwidth rate)
>> Policy 1# resv 3                (Set committed information rate)
```

3. On Alteon, select a BWM contract and name the contract. Each contract must have a unique number from 1 to 1024.

```
>> Policy 1# /cfg/bwm/cont 1
>> BWM Contract 1# name dial-up
```

4. Set the bandwidth policy for this contract. Each BWM contract must be assigned a bandwidth policy.

```
>> BWM Contract 1# pol 1
```

5. Enable this BWM contract.

```
>> BWM Contract 1# ena
```

6. Select the second bandwidth policy for broadband customers.

```
>> BWM Contract 1# /cfg/bwm/pol 2
```

7. Set the hard, soft, and reserved rate limits for this policy, in Mbps.

```
>> Policy 2# hard 30               (Set "never exceed" rate)
>> Policy 2# soft 25               (Set desired bandwidth rate)
>> Policy 2# resv 20               (Set committed information rate)
```

8. On Alteon, select the second BWM contract and name the contract.

```
>> Policy 2# /cfg/bwm/cont 2
>> BWM Contract 2# name broadband
```

9. Set the bandwidth policy for this contract. Each BWM contract must be assigned a bandwidth policy.

```
>> BWM Contract 2# pol 2
```

10. Enable this BWM contract.

```
>> BWM Contract 2# ena
```

11. On Alteon, apply and verify the configuration.

```
>> Port 2# apply
>> Port 2# /cfg/bwm/cur
```

Examine the resulting information. If any settings are incorrect, make any appropriate changes.

12. On Alteon, save your new configuration changes.

```
>> Bandwidth Management# save
```

13. On Alteon, check the BWM information.

```
>> Bandwidth Management# /info/bwm <contract number>
```

Check that all BWM contract parameters are set correctly. If necessary, make any appropriate configuration changes and then check the information again.

Example Configuring Grouped Contracts for Bandwidth Sharing

In this example, BWM is configured to allow sharing of BWM resources by configuring a group contract. While the group hard limit is essentially the aggregate of the hard limits defined for each contract in the group, any unused bandwidth may be shared amongst all member contracts.

For example, a group level contract is defined with four individual contracts that have committed information rates (CIR) of 10, 20, 30, and 40 Mbps each. Together, the total CIR of the member contracts is 100 Mbps. Based on how much traffic is actually being sent by all the contracts in the group, the hard limits of each contract are readjusted every few seconds, in proportion to each contract's share in the group. In effect, the contract with only 10 Mbps may be allowed at times to share any unused resources in the group and burst up to a higher hard limit. If that contract is removed from the group, the contract reverts to its individual hard limits, and any traffic above its configured hard limit is dropped as usual. For a more detailed explanation on how hard limits for contracts behave in a contract group, see Table 66 - Bandwidth Reallocation in Grouped Contracts, page 765.

Note: While *traffic shaping* contracts may be added to a group level contract, their soft and reserved limits are not readjusted.

1. Ensure BWM is enabled on Alteon.

```
>> /cfg/bwm/on
```

2. Configure Alteon as you normally would for SLB. Configuration includes the following tasks:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface on Alteon.
   - Define each real server.
   - Define a real server group.
   - Define a virtual server.
   - Define the port configuration.
3. Select the first bandwidth policy and set the hard, soft, and reserved rate limits for the bandwidth policy, in Mbps.

`>> # /cfg/bwm/pol 1`	(Select BWM Policy 1)
`>> Policy 1# hard 10M`	(Set "never exceed" rate)
`>> Policy 1# soft 5M`	(Set desired bandwidth rate)
`>> Policy 1# resv 1M`	(Set committed information rate)

4. Configure BWM contract 1. Each contract must have a unique number from 1 to 1024.

```
>> Policy 1# /cfg/bwm/cont 1
```

5. Assign the bandwidth policy 1 to Contract 1.

```
>> BWM Contract 1# pol 1
```

6. Enable Contract 1.

```
>> BWM Contract 1# ena
```

7. Select Bandwidth Policy 2.

```
>> BWM Contract 1# /cfg/bwm/pol 2
```

8. Set the hard, soft, and reserved rate limits for this policy, in Mbps.

`>> Policy 2# hard 20`	(Set "never exceed" rate)
`>> Policy 2# soft 15`	(Set desired bandwidth rate)
`>> Policy 2# resv 10`	(Set committed information rate)

9. On Alteon, select BWM Contract 2.

```
>> Policy 2# /cfg/bwm/cont 2
```

10. Assign Bandwidth Policy 2 to Contract 2. Each BWM contract must be assigned a bandwidth policy.

```
>> BWM Contract 2# pol 2
```

11. Enable Contract 2.

```
>> BWM Contract 2# ena
```

12. As described in step 7 through step 11, configure Policy 3 with hard, soft, and reserved limits of 30, 25, and 20 Mbps respectively. Then create Contract 3 and apply Policy 3 to this contract.
13. Configure Policy 4 with hard, soft, and reserved limits of 40, 35, and 30 Mbps respectively. Then create Contract 4 and apply Policy 4 to this contract.
14. Configure BWM Contract Group 1 and add all four contracts to this group.

`>> /cfg/bwm/group 1`	(Select Contract Group 1)
`>> BW Group 1# add 1`	(Add Contract 1 to Group 1)
`Contract 1 added to group 1.`	
`>> BW Group 1# add 2`	(Add Contract 2 to Group 1)
`Contract 2 added to group 1.`	(Add Contract 3 to Group 1)
`>> BW Group 1# add 3`	

```
Contract 3 added to group 1.
>> BW Group 1# add 4                          (Add Contract 4 to Group 1)
Contract 4 added to group 1.
```

15. Apply and verify the configuration.

```
>> Port 2# apply
>> Port 2# /cfg/bwm/cur
```

Examine the resulting information. If any settings are incorrect, make any appropriate changes.

16. Save your new configuration changes.

```
>> Bandwidth Management# save
```

17. Check the BWM information.

```
>> Bandwidth Management# /info/bwm <contract number>
```

Check that all BWM contract parameters are set correctly. If necessary, make any appropriate configuration changes and then check the information again.

Example Configuring an IP User-Level Rate Limiting Contract

This example is for university that wants to restrict the amount of TCP traffic for individual students and for the student body as a whole. Contract 1 is configured as follows:

- Each student (IP address) is limited to 64 kbps.
- All members of the student body are limited to maximum (hard limit) of 10 Mbps.
- If the number of octets sent out exceeds the value of the entire contract (10 Mbps), excess octets are dropped.
- If the number of octets is below the value of the contract (10 Mbps), a session is created on Alteon that records the student's IP address, the egress port number, and the contract number, as well as the number of octets transferred for that second. The session updates the number of octets being transferred every second, thus maintaining traffic within the configured user limit of 64 kbps.

1. Select the first bandwidth policy.

   Each policy must have a number from 1 to 512.

```
>> # /cfg/bwm/pol 1
```

2. Configure the BWM policy with a hard limit of 10 Mbps and a "user limit" of 64 kbps. Apply that policy to Contract 1.

```
>> Policy 1# hard 10m
>> Policy 1# userlim 64k
>> Policy 1# /cfg/bwm/cont 1                 (Select Contract 1)
>> BW Contract 1# policy 1                   (Apply policy 1 to this contract)
```

3. Configure a filter to match the source IP address range of the student body, and assign BWM Contract 1 to that filter.

`/cfg/slb/filt 20/sip 150.150.0.0/smask 255.255.0.0/action allow`	(Allow student traffic)
`>> Filter 20 # adv`	(Select the *Filter 20 Advanced* menu)
`>> Filter 20 Advanced# cont 1`	(Apply BWM Contract 1 to this filter)

4. Add the filter to an ingress port on Alteon.

```
>> /cfg/slb/port 1/filt ena/add 20
```

5. In the BWM configuration, enable IP limiting.

```
>> /cfg/bwm/cont 1/iplimit
```

6. Determine whether the user should be identified by source or destination IP address.
   — If the contract is used for traffic going out to the Internet, define it by the source IP address: **`iptype sip`**.
   — If the contract is used to limit the amount of traffic downloaded from the user by a client on the Internet, define it by the destination IP address: **`iptype dip`**.

```
>> BW Contract 1# iptype sip
```

7. Disable traffic shaping on this contract. Traffic shaping cannot be used in user-level rate limiting contracts.

```
>> /cfg/bwm/cont 1/shaping dis
```

8. Apply and save the configuration.
9. View the current per-user BWM sessions for the active contract.

```
/stats/bwm/port 1/cont 1
```

Example Configuring BWM Preferential Services

BWM can be used to provide preferential treatment to certain traffic, based on source IP blocks, applications, URL paths, or cookies. You may find it useful to configure higher policy rate limits for specific sites, for example, those used for e-commerce.

In this example, there are two Web sites, "A.com" and "B.com." BWM is configured to give preference to traffic sent to Web site "B.com:"

1. Configure Alteon as you normally would for SLB. Configuration includes the following tasks:
   — Assign an IP address to each of the real servers in the server pool.
   — Define an IP interface on Alteon.
   — Define each real server.
   — Define a real server group.
   — Define a virtual server.
   — Define the port configuration.

For more information about SLB configuration, refer to Server Load Balancing, page 165.

Note: Ensure BWM is enabled on Alteon (**/cfg/bwm/on**).

2. Select bandwidth Policy 1.

   Each policy must have a number from 1 to 512.

```
>> # /cfg/bwm/pol 1
```

3. Set the hard, soft, and reserved rate limits for the bandwidth policy in Mbps.

```
>> Policy 1# hard 10          (Set "never exceed" rate)
>> Policy 1# soft 8           (Set desired bandwidth rate)
>> Policy 1# resv 5           (Set committed information rate)
```

4. Select a BWM contract and name the contract. Each contract must have a unique number from 1 to 1024.

```
>> Policy 1# /cfg/bwm/cont 1
>> BWM Contract 1# name a.com
```

5. Assign the bandwidth policy to this contract. Each BWM contract must be assigned a bandwidth policy.

```
>> BWM Contract 1# pol 1
```

6. Enable this BWM contract.

```
>> BWM Contract 1# ena
```

7. Select Bandwidth Policy 2.

```
>> BWM Contract 1# /cfg/bwm/policy 2
```

8. Set the hard, soft, and reserved rate limits for this policy, in Mbps.

```
>> Policy 2# hard 18          (Set "never exceed" rate)
>> Policy 2# soft 15          (Set desired bandwidth rate)
>> Policy 2# resv 10          (Set committed information rate)
```

9. Select the second BWM contract and name the contract.

```
>> Policy 2# /cfg/bwm/cont 2
>> BWM Contract 2# name b.com
```

10. Assign the bandwidth policy to this contract. Each BWM contract must be assigned a bandwidth policy.

```
>> BWM Contract 2# pol 2
```

11. Enable this BWM contract.

```
>> BWM Contract 2# ena
```

12. Create a virtual server that is used to classify the frames for Contract 1 and assign the virtual server IP address for this server. Assign the BWM contract to the virtual server. Repeat this procedure for a second virtual server.

    Note: This classification applies to the services within the virtual server and not to the virtual server itself.

    The classification rule for these BWM contracts is based on a virtual service. One of the BWM contracts is applied to any frames that are sent to the virtual server associated with that contract.

`>> BWM Contract 2# /cfg/slb/virt 1/service 80/cont 1`	(Assign contract to Virtual Server 1)
`>> Virtual Server 1# vip 100.2.16.2`	(Set virtual server VIP address)
`>> Virtual Server 1# ena`	(Enable this virtual server)
`>> Virtual Server 1# /cfg/slb/virt 2/cont 2`	(Assign contract to virtual server)
`>> Virtual Server 2# vip 100.2.16.3`	(Set virtual server IP address)
`>> Virtual Server 2# ena`	(Enable this virtual server)

13. Apply and verify the configuration.

```
>> Virtual Server 2# apply
>> Virtual Server 2# cfg/bwm/cur
```

    Examine the resulting information. If any settings are incorrect, make the appropriate changes.

14. Save your new configuration changes.

```
>> Bandwidth Management# save
```

15. Check the bandwidth management information.

```
>> Bandwidth Management# /info/bwm <contract number>
```

    Check that all BWM contract parameters are set correctly. If necessary, make any appropriate configuration changes and then check the information again.

Example Configuring Content-Intelligent Bandwidth Management

Content-intelligent BWM allows the network administrator or Web site manager to control bandwidth based on Layer 7 content such as URLs, HTTP headers, or cookies.

All three types of Bandwidth Management are accomplished by following the configuration guidelines on content load balancing described in Content-Intelligent Server Load Balancing, page 219 and Application Redirection, page 455. You also need to assign a contract to each defined string, where the string is contained in a URL, an HTTP header, or a cookie.

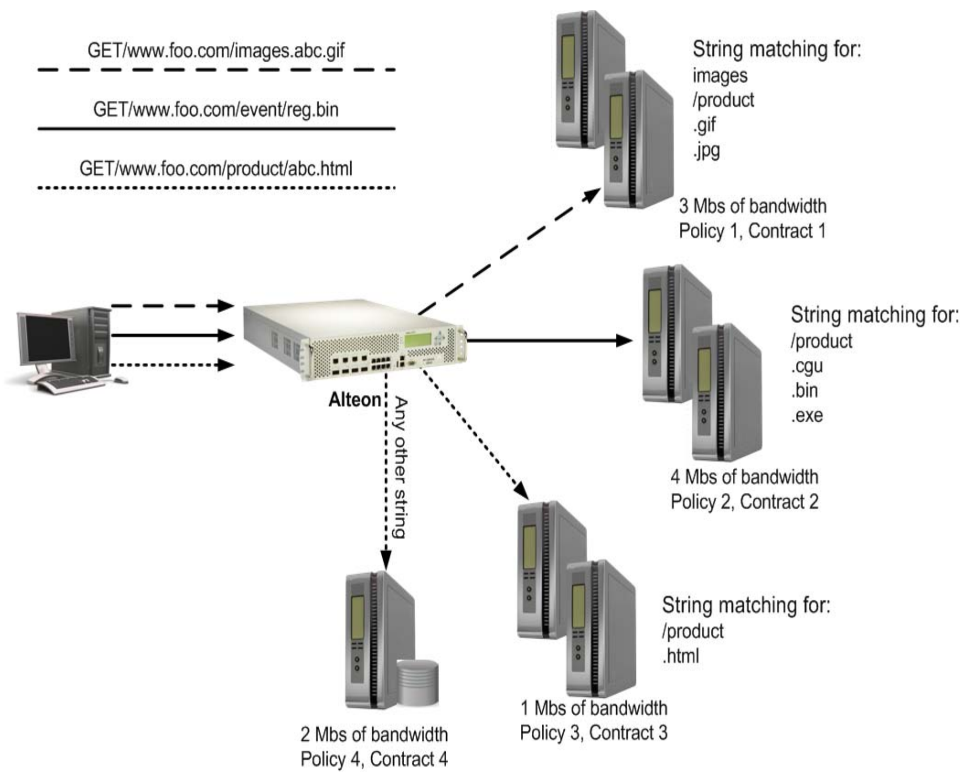
BWM based on Layer 7 content gives Web site managers the following capabilities:

- Ability to allocate bandwidth based on the type of request.

- Alteon allocates bandwidth based on certain strings in the incoming URL request. For example, if a Web site has 10 Mbps of bandwidth, the site manager can allocate 1 Mbps of bandwidth for static HTML content, 3 Mbps of bandwidth for graphic content and 4 Mbps of bandwidth for dynamic transactions, such as URLs with cgi-bin requests and .asp requests.
- Ability to prioritize transactions or applications.
- By allocating bandwidth, Alteon can guarantee that certain applications and transactions get better response time.
- Ability to allocate a certain amount of bandwidth for requests that can be cached.

As shown in Figure 132 - URL-Based SLB with Bandwidth Management, page 784, users are able to allocate a certain percentage of bandwidth for Web cache requests by using the URL parsing and bandwidth management feature.

Figure 132: URL-Based SLB with Bandwidth Management



This example assumes you have configured URL-based SLB and the layer 7 strings as described in Content-Intelligent Server Load Balancing, page 219. For URL-based SLB, a user has to first define strings to monitor. Each of these strings is attached to real servers, and any URL with the string is load balanced across the assigned servers. The best way to achieve URL-based bandwidth management is to assign a contract to each defined string. This allocates a percentage of bandwidth to the string or URL containing the string.

1. Configure Content-Intelligent Server Load Balancing, page 219.
2. Configure BWM policies with the desired bandwidth limits. In this example, four policies are configured, as illustrated in Figure 132 - URL-Based SLB with Bandwidth Management, page 784.

```
>> Main# /cfg/bwm/pol 1/hard 3M/soft 2M/res 1M
>> Policy 1# /cfg/bwm/pol 2/hard 4M/soft 3M/res 2M
>> Policy 2# /cfg/bwm/pol 3/hard 1M/soft 500k/res 250k
>> Policy 3# /cfg/bwm/pol 4/hard 2M/soft 1M/res 500k
```

3. Configure BWM contracts and apply the appropriate policies to the contracts. In this example, the policy numbers correspond to the contract numbers.

```
>> Main# /cfg/bwm/cont 1/policy 1                    (Apply Policy 1 to Contract 1)
>> BW Contract 1# /cfg/bwm/cont 2/policy 2
>> BW Contract 2# /cfg/bwm/cont 3/policy 3
>> BW Contract 3# /cfg/bwm/cont 4/policy 4
```

4. Identify the defined string IDs that were configured.

```
>> # /cfg/slb/layer7/slb/cur
```

For easy configuration and identification, each defined string is assigned an ID number, as shown in the following table. The third column shows the BWM contracts to assign to the strings in this example.

ID	SLB String	BWM Contract
1	any	4
2	.gif	1
3	.jpg	1
4	.cgi	2
5	.bin	2
6	.exe	2
7	.html	3

5. Assign BWM contracts to each string using the syntax shown.

```
>> Main# /cfg/slb/layer7/slb/cont <String ID> < BWM Contract number>
```

6. Verify that the strings and contracts are assigned properly.

```
>> Server Load Balance Resource# cur
Number of entries: 2
1: any, cont 4
2: .gif, cont 1
3: .jpg, cont 1
4: .cgi, cont 2
5: .bin, cont 2
6: .exe, cont 2
7: .html, cont 3
```

7. Configure a real server to handle the URL request.

```
>> # /cfg/slb/real 2/layer7/addlb <SLB string ID>
```

SLB string ID is the identification number of the defined string as displayed when you enter the **cur** command. For example: `/cfg/slb/real 2/layer7/addlb 3`

8. Either enable Direct Access Mode (DAM) on Alteon or configure a proxy IP address on the client port. To turn on DAM.

```
>> # /cfg/slb/adv/direct ena
```

To turn off DAM and configure a proxy IP address on the client port.

```
>> # /cfg/slb/adv/direct dis
>> # /cfg/slb/port 2/proxy ena              (Enable use of proxy IP on this port)
>> # /cfg/slb/pip/type port
>> # /cfg/slb/pip/add 12.12.12.12           (Add this proxy IP address to Port 2)
```

For more information on proxy IP addresses, see Client Network Address Translation (Proxy IP), page 190.

Port mapping for content-intelligent SLB can be performed by enabling DAM on Alteon, or disabling DAM and configuring a proxy IP address on the client port.

9. Turn on HTTP SLB processing on the virtual server. Configure everything under the virtual server as in Example Configuring User/Application Fairness, page 776.

```
>> # /cfg/slb/virt 1/service 80/http/httpslb urlslb
```

If the same string is used by more than one service, and you want to allocate a certain percentage of bandwidth to this URL string for this service on the virtual server, then define a rule using the **urlcont** command.

```
>> # /cfg/slb/virt 1/service 80/http/urlcont <SLB string ID> <BW Contract
number>
```

This contract is tied to service 1. The **urlcont** command overrides the contract assigned to the URL string ID.

10. Enable SLB.

```
>> # /cfg/slb/on
```

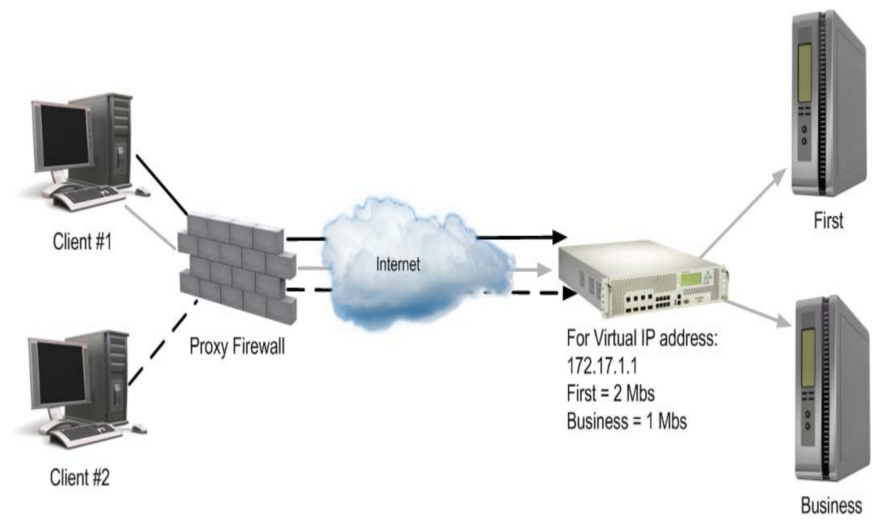
11. **Apply** and **save** the configuration.

Example Configuring Cookie-Based Bandwidth Management

Cookie-based BWM enables Web site managers to prevent network abuse by bandwidth-hogging users. Using this feature, bandwidth can be allocated by type of user or other user-specific information available in the cookie.

Cookie-based Bandwidth Management enables service providers to create tiered services. For example, Web site managers can classify users as first class, business class, and coach, and allocate a larger share of the bandwidth for preferred classes.

Figure 133: Cookie-Based Bandwidth Management



Note: Cookie-based BWM does not apply to cookie-based persistency or cookie passive/active mode applications.

In this example, you assign bandwidth based on cookies. First, configure cookie-based SLB, which is very similar to URL-based load balancing. Any cookie containing the specified string is redirected to the assigned server.

A In this scenario, the Web site has a single virtual server IP address and supports multiple classes of users. Turn on cookie parsing for the service on the virtual server.

```
>> # /cfg/slb/virt 1/service 80
>> Virtual Server 1 http Service# http/httpslb
Application: urlslb|host|cookie|browser|urlhash|headerhash|version|others|none
Select Application: cookie
Operation: and|or|none
Select Operation: ena
Enter Cookie Name:
Enter the starting point of the cookie value [1-64]: 1
Enter the number of bytes to be extract [1-64]: 8
Look for cookie in URL [e|d]:
```

1. Define one or more load-balancing strings.

```
>> # /cfg/slb/layer7/slb/addstr <l7lkup|pattern> <SLB string>
```

For example:

```
>> # /cfg/slb/layer7/slb/addstr 171kup “Business”
# add 171kup “First”
# add 171kup “Coach”
```

2. Allocate bandwidth for each string. To do this, assign a BWM contract to each defined string.

```
>> # /cfg/slb/layer7/slb/cont <SLB string ID> <BWM Contract number>
```

3. Configure a real server to handle the cookie. To add a defined string where *SLB string ID* is the identification number of the defined string:

```
>> # /cfg/slb/real 2/layer7/addlb <SLB string ID>
```

For example:

```
>> # /cfg/slb/real 2/layer7/addlb
```

4. Either enable DAM on Alteon or configure a proxy IP address on the client port. To turn on DAM:

```
>> # /cfg/slb/adv/direct ena
```

To turn off DAM and configure a Proxy IP address on the client port:

```
>> # /cfg/slb/adv/direct dis
>> # /cfg/slb/pip
>> Proxy IP address# type port                    (Use port-based proxy IP)
>> Proxy IP Address# add 12.12.12.12
>> # /cfg/slb/port 2
>> SLB Port 2# proxy ena
```

For more information on proxy IP addresses, see Client Network Address Translation (Proxy IP), page 190.

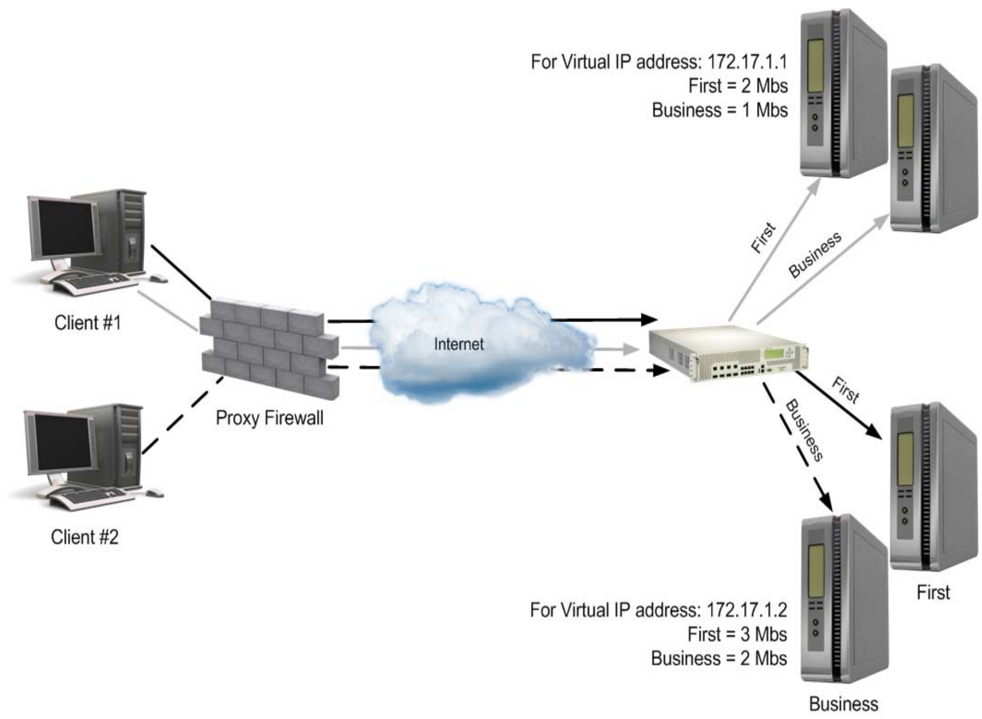
Note: By enabling DAM on Alteon or, alternatively, disabling DAM and configuring a proxy on the client port, port mapping for URL-based load balancing can be performed.

5. Enable SLB.

```
>> # /cfg/slb/on
```

B In this scenario, the Web site has multiple virtual server IP addresses, and the same user classification or multiple sites use the same string name. There are two virtual IP (VIP) addresses: 172.17.1.1 and 172.17.1.2. Both the virtual servers and sites have first class and business class customers, with different bandwidth allocations, as shown in Figure 134 - Cookie-Based Preferential Services, page 789:

Figure 134: Cookie-Based Preferential Services



The configuration to support this scenario is similar to Example A, page 787. Note the following:

1. Configure the string and assign contracts for the strings and services.
2. If the same string is used by more than one service, and you want to allocate a certain percentage of bandwidth to a user class for this service on the virtual server, then define a rule using the **urlcont** command.

```
>> # /cfg/slb/virt 1/service 80/http/urlcont <URL path ID> <BW Contract number>
```

Note: When assigning **/cfg/slb/virt 1/service 80/http/urlcont** (Contract 1) and **/cfg/slb/layer7/lb/cont** (Contract 2) to the same URL, **urlcont** will override Contract 2, even if Contract 2 has higher precedence.

Example Configuring Security Management

BWM can be used to prevent Denial of Service (DoS) attacks that generate a flooding of "necessary evil" packets. BWM limits the rate of TCP SYN, ping, and other disruptive packets. BWM can alert the network manager when soft limits are exceeded.

In this example, a filter is configured to match ping packets, and BWM is configured to prevent DoS attacks by limiting the bandwidth policy rate of those packets:

1. Configure Alteon as usual for SLB (see Server Load Balancing, page 165):
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface on Alteon.

— Define each real server.
— Define a real server group.
— Define a virtual server.
— Define the port configuration.

Note: Ensure BWM is enabled on Alteon (**/cfg/bwm/on**).

2. Select a bandwidth policy. Each policy must have a number from 1 to 512.

```
>> # /cfg/bwm/pol 1
```

3. Set the hard, soft, and reserved rate limits for this policy in kilobytes.

```
>> Policy 1# hard 250k        (Set "never exceed" rate)
>> Policy 1# soft 250k        (Set desired bandwidth rate)
>> Policy 1# resv 250k        (Set committed information rate)
```

4. Set a parameter between 8192 and 128000 bytes. The buffer depth for a BWM contract should be set to a multiple of the packet size.

```
>> Policy 1# buffer 8192
```

5. On Alteon, select a BWM contract and name the contract. Each contract must have a unique number from 1 to 1024.

```
>> Bandwidth Management# /cfg/bwm/cont 1
>> BWM contract 1# name icmp
```

6. Set the bandwidth policy for the contract. Each BWM contract must be assigned a bandwidth policy.

```
>> BWM Contract 1# pol 1
```

7. Enable the BWM contract.

```
>> BWM Contract 1# ena
```

8. Create a filter that will be used to classify the frames for this contract and assign the BWM contract to the filter.

   The classification rule for this BWM contract is based on a filter configured to match ICMP traffic. The contract will be applied to any frames that match this filter.

```
>> BW Contract 1# /cfg/slb/filt 1/proto icmp   (Define protocol affected by filter)
>> Filter 1# adv/icmp any                       (Set the ICMP message type)
>> Filter 1 Advanced# cont 1                    (Assign BWM Contract 1 to this
                                                 filter)
>> Filter 1 Advanced# /cfg/slb/filt 1/ena       (Enable this filter)
>> Filter 1# apply                              (Port and enable filtering)
```

9. On Alteon, apply and verify the configuration.

```
>> Filter 1 Advanced# apply
>> Filter 1 Advanced# /cfg/bwm/cur
```

Examine the resulting information. If any settings are incorrect, make the appropriate changes.

10. On Alteon, save your new configuration changes.

```
>> Bandwidth Management# save
```

11. On Alteon, check the BWM information.

```
>> Bandwidth Management# /info/bwm <contract number>
```

Check that all BWM contract parameters are set correctly. If necessary, make any appropriate configuration changes and then check the information again.

Example Configuring Time and Day Policies

Bandwidth Management contracts can be configured to have different limits depending on the time of day and day of the week. For example, in office networks that are typically busy during a workday, higher bandwidth limits can be applied during peak work hours. Lower bandwidth limits can be applied during hours with minimal traffic, such as on evenings or weekends.

Up to two time policies can be applied to each contract. The default settings for each time policy are: "**`Day everyday, From Hour 12am, To Hour 12am, Policy 512, time policy disabled`**"

If both Time Policy 1 and Time Policy 2 are enabled on a contract, and both policies match the current time set in Alteon's system clock, Time Policy 1 will take effect.

Note: When configuring time policies, the "To" hour cannot be earlier than the "From" hour, as in a time policy set from 7PM to 7AM. Alteon does not calculate time policies that cross the 24-hour day boundary.

1. Configure three BWM policies for high, low, and default bandwidth. These policies will be applied to different time policies in step 5.

```
>> /cfg/bwm/policy 1/hard 10M/soft 5M          (For peak working hours)
>> /cfg/bwm/policy 1/hard 5M/soft 1M           (For weekday evening hours)
>> /cfg/bwm/policy 3/hard 4M/soft 2M           (For all other times)
```

2. Create a BWM contract that will contain the time policies.

```
>> /cfg/bwm/cont 1
```

3. Create the first time policy under Contract 1, for peak working hours.

```
>> # /cfg/bwm/cont 1/timepol 1
>> BW Contract 1 Time Policy 1# day weekday
Current Time Policy Day: everyday
Pending new Time Policy Day: weekday
>> BW Contract 1 Time Policy 1# from 7am
Current Time Policy from hour: 12am
Pending new Time Policy from hour: 7am
>> BW Contract 1 Time Policy 1# to 7pm
Current Time Policy to hour: 2am
Pending new Time Policy to hour: 7pm
>> BW Contract 1 Time Policy 1# policy 1        (Assign highest BWM policy to this
                                                time policy)
>> BW Contract 1 Time Policy 1# ena
Current status: disabled
New status: enabled
```

4. Create the second time policy under Contract 1, for weekday evening hours.

```
>> # /cfg/bwm/cont 1/timepol 2/day weekday/from 7pm/to 11pm/policy 2/ena
```

5. Apply the default BWM Policy 3 to this contract. This BWM policy will be in effect at all other times beyond the specifications of the two time policies.

```
>> # /cfg/bwm/cont 1/policy 3/ena
```

6. Assign the contract to an ingress port on Alteon.

```
>> Main# /cfg/port 1
>> Port 1# cont 1
Current BW Contract: 256
New pending BW Contract: 1
```

7. Apply and save the configuration.

Example Egress Bandwidth Tuning for Lower Speed Networks

When an Alteon is connected to a router that feeds into lower speed networks, the egress traffic from Alteon should be throttled down to prevent the packets from being dropped from the router as it forwards traffic into the slower network.

For example, an Alteon may be connected to a router with high bandwidth of 1 Gbps. However, that router may be connected into a Wide Area Network (WAN) using a T1 line (1.544 Mbps) or a T3 line (44.736 Mbps). Any packets that exceed the capacity of the WAN are dropped.

Egress bandwidth tuning is only available on 10/100/1000Base-T ports. To tune down the egress bandwidth to T3 speeds, enter the following commands:

```
>> # /cfg/port 1                            (Select the desired port)
>> Port 1# egbw 44M                         (Change the egress bandwidth
                                            to 44 Mbps)
>> Current port egress bandwidth: 0K
New pending port egress bandwidth: 44M
```

Example Overwriting the TCP Window Size

The TCP window size set in the packet indicates how many bytes of data the receiver of that TCP packet can send without waiting for acknowledgement. In network environments where congestion is a common problem and traffic usually exceeds the configured BWM soft limit in a BWM contract, the TCP window size may be overwritten to better accommodate the prevailing traffic rates. It would be beneficial if the TCP traffic was slowed down by modifying the TCP window size rather than by dropping TCP packets, which would cause retransmissions.

By default, the TCP window size is overwritten only when traffic exceeds the soft limit of the BWM contract, and when the window size is above 1500 bytes. To overwrite TCP window size on a contract, enter the following command:

```
>> # /cfg/bwm/cont 1/wtcpwin e
```

Chapter 27 – XML Configuration API

Alteon supports an Extensible Markup Language (XML) configuration application programming interface (API). This support provides a common interface for applications to operate with an Alteon. XML was chosen for its wide adoption and usage. XML is also supported by the Alteon Threat Protection System.

This chapter includes the following sections:

Software Components

This feature uses two distinct software components that work together to interpret XML files sent to Alteon. These two software components are:

- **Schema document**—The schema document is the roadmap that enables Alteon to interpret the XML documents that are sent to it. This schema document defines the markup tags that appear in the XML document and what each means. The following is an example schema document used by the XML Configuration API:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs=
"http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
   <xs:element name="AlteonConfig">
     <xs:annotation>
       <xs:documentation> Comment describing your root element</
xs:documentation>
     </xs:annotation>
     <xs:complexType>
       <xs:sequence>
         <xs:element name="Cli" maxOccurs="unbounded">
           <xs:complexType>
             <xs:attribute name="Command" type="xs:string"use="required"/
>          </xs:complexType>
         </xs:element>
       </xs:sequence>
       <xs:attribute name="Version" type="xs:int" use="required"/>
     </xs:complexType>
   </xs:element>
</xs:schema>
```

- **XML Parser**—An XML parser is embedded in the software. This parser is used to interpret an XML file into usable CLI commands.

XML Configuration File

The XML configuration file conforms to the rules laid out by the DTD document. The configuration file can either be produced by an application equipped to do so, or manually in a text editor. For information about the form and format of the Extensible Markup Language, refer to the World Wide Web Consortium XML Web site at http://www.w3.org/XML/.

The following is an example of an XML file that could be used to configure Alteon:

```
<?xml version="1.0" encoding="UTF-8"?>
<AlteonConfig xmlns:xsi=
"http://www.w3.org/2001/XMLSchemainstance"
xsi:noNamespaceSchemaLocation="AOSConfig.xsd" Version="1">
 <Cli Command="/c/ip/if 1/en"/>
   <Cli Command="/c/l3/if 1 addr 47.81.24.189"/>
   <Cli Command="/c/l3/if 1 mask 255.255.255.0"/>
   <Cli Command="/c/l3/if 1 broad 47.81.24.255"/>
</AlteonConfig>
```

XML File Transmission

Secure Socket Layer (SSL) is used as the transport medium for sending XML configuration files to Alteon. An SSL client is needed to connect to Alteon using certificate authentication. This SSL client can be a standalone application or embedded as part of another application. After authentication takes place, the file can be sent securely.

Notes

- Certificates used for authentication purposes must be in PEM format. Self-signed certificates are supported for this purpose.
- A certificate can be either obtained via TFTP/FTP or by simply pasting the certificate directly through the CLI:

  FTC1 - ADC-VX - Main# /cfg/sys/access/xml/gtcert

  Import from text or file in PEM format [text|file] [text]:
- Running the “gtcert” is only allowed when you are using SSH to access Alteon, if you are using telnet you will get the following error:

  FTC1 - ADC-VX - Main# /cfg/sys/access/xml/gtcert

  “Access Denied: This operation can only be performed over a secure connection such as HTTPS or SSH. Connect to Alteon using a secure protocol and retry.”

XML Configuration

The following is an example procedure to enable and use the XML Configuration API.

To use the XML configuration feature

Note: All CLI commands with an enable option also have a corresponding disable option.

1. Globally enable XML configuration. The XML Configuration API is disabled by default. To enable this feature, enter the following command:

```
>> Main# /cfg/sys/access/xml/xml enable
```

2. Optionally, set the XML transport port number. Since SSL is the transport mechanism for the XML configuration file, the port used by Alteon to receive these files is the SSL port by default. You can change the default by using the following command:

```
>> Main# /cfg/sys/access/xml/port <port number>
```

Note: Since both HTTPS and XML use SSL as a transport layer, the two are closely tied together. Both HTTPS and XML must use the same port if both are enabled.

3. Import client certificate. Certificate authentication is required to send an XML configuration file to Alteon. To import a client certificate, do the following:

```
>> Main# /cfg/sys/access/xml/gtcert <TFTP/FTP IP Address>
 <Certificate File Name> <FTP User Name> <FTP Password>
```

After entering the required information, the client certificate is downloaded to Alteon.

Additional Feature Commands

The following commands are used to maintain and monitor the XML Configuration API:

To delete the client certificate

```
>> Main /cfg/sys/access/xml/delcert
```

To display the current client certificate

```
>> Main# /cfg/sys/access/xml/dispcert
```

To enable XML debug operations

```
>> Main# /cfg/sys/access/xml/debug/ enabled
```

Enabling XML debug operations results in all commands in the XML file to be displayed on the console with one of the following prefaces:

— "running XML cmd:"

— "Invalid XML cmd:"

All responses to these commands are also displayed on the console.

To display the current XML API configuration

```
>> Main# /cfg/sys/access/xml/cur
```

Chapter 28 – AppShape++ Scripting

This chapter introduces the AppShape++ scripting feature. For more information on the AppShape++ API and scripts, see the *Alteon Application Switch AppShape++ Reference Guide*.

The following topics are addressed in this chapter:

AppShape++ Overview

AppShape++ is a framework for customizing application delivery using user-written scripts.

AppShape++ provides the flexibility to control application flows and fully meet business requirements in a fast and agile manner.

The AppShape++ framework enables you to:

- Extend Radware's ADC Fabric services with delivery of new applications
- Quickly deploy new services
- Mitigate application problems without changing the application
- Preserve infrastructure investment by adding new capabilities without additional equipment investment

AppShape++ provides specific API extension to the Tool Command Language (Tcl) to query and manipulate data, and take actions such as server selection. For more informaton on Tcl, see www.tcl.tk/.

The AppShape++ scripts can be attached to virtual service thus allowing to perform protocol content switching decisions and modification on any TCP/UDP protocol.

AppShape++ Script Repository

AppShape++ scripts need to be uploaded to the Alteon repository before they can be used. Up to 1024 scripts are supported.

When the Apply command is invoked, all new or edited scripts are validated.

AppShape++ Script Activation

An AppShape++ script is activated when attached to a virtual service. Up to 16 AppShape++ scripts can be attached to the same virtual service, but each one must have a different priority level. The priority level determines the order in which Alteon executes the scripts.

Each AppShape++ script can be attached to any number of scripts.

Note: When attaching an AppShape++ script to a non-HTTP service, legacy content-based load balancing for that service must be disabled.

Example Configuring a Virtual Service with an AppShape++ Script

1. Make sure that Alteon is configured for basic SLB:
   - Define an IP interface.
   - Enable SLB.
   - Assign an IP address to each of the real servers in the server pool.
   - Define each real server.
   - Assign servers to real server groups.
   - Define server port and client port.
   - Define virtual server
   - Define virtual service

   For more information on how to configure your network for SLB, see Server Load Balancing, page 165.
2. Write the AppShape++ script which will complete the virtual service behavior. Radware recommends using a Tcl-enabled editor.
3. Import the script to Alteon the switch.

```
>> Main # /cfg/slb/appshape/script myscript
>> AppShape++ script myscript# import
Import script from text or file in PEM format [text|file] [text]: file
Enter hostname (and IP version) or IP address of FTP/TFTP/SCP server:
192.162.1.1
Enter name of file on FTP/TFTP/SCP server: myscript.tcl
Enter username for FTP/SCP server or hit return for TFTP server:
Enter password for username on FTP/SCP server:
Enter "scp" or hit return for FTP server:
```

4. Enable the script.

```
>> AppShape++ script myscript# ena
```

5. Attach the script to the virtual service.

```
>> Main# /cfg/slb/virt 1/service 80/appshape/add 1 myscript
```

Appendix A – Layer 7 String Handling

This appendix describes how to create and manage the Layer 7 content used for configuring content-intelligent load balancing and redirection features described throughout this user guide.

The following topics are discussed in this appendix:

Note: For all content-intelligent load balancing features, enable Direct Access Mode (DAM) or configure proxy IP addresses. For more information, see Direct Access Mode, page 200

Exclusionary String Matching for Real Servers

URL-based SLB and application redirection can match or exclude up to 128 strings. Examples of strings are:

- "/product"—Matches URLs that starts with /product.
- "product"—Matches URLs that have the string "product" anywhere in the URL.

You can assign one or more strings to each real server. When more than one URL string is assigned to a real server, requests matching any string are redirected to that real server. There is also a special string known as **any** that matches all content.

Alteon also supports *exclusionary string matching*. Using this option, you can define a server to accept any requests regardless of the URL, *except* requests with a specific string.

Note: Once exclusionary string matching is enabled, clients cannot access the URL strings that are added to that real server. This means you cannot configure a dedicated server to receive a certain string and, at the same time, have it exclude other URL strings. The exclusionary feature is enabled per server, not per string.

For example, the following strings are assigned to a real server:

```
string 1 = cgi
string 2 = NOT cgi/form_A
string 3 = NOT cgi/form_B
```

As a result, all cgi scripts are matched except **form_A** and **form_B**.

Configuring Exclusionary URL String Matching

This configuration example illustrates how to configure a server to handle any requests *except* requests that contain the string "test", *or* requests that start with "/images" or "/product".

To configure exclusionary URL string matching

1. Before you can configure URL string matching, ensure that Alteon has already been configured for basic SLB:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface on Alteon.
   - Define each real server.
   - Assign servers to real server groups.
   - Define virtual servers and services.
   - Enable SLB.
   - Enable URL-based HTTP SLB.

   For information on how to configure your network for SLB, see Server Load Balancing, page 165.
2. Add the load balancing strings (for example `test`, `/images`, and `/product`) to the real server:

```
>> # /cfg/slb/layer7/slb/addstr "test"
>> Server Loadbalance Resource# addstr "/images"
>> Server Loadbalance Resource# addstr "/product"
```

3. Apply and save the configuration.
4. Identify the IDs of the defined strings.

```
>> Server Loadbalance Resource# cur
```

ID	SLB String
1	any
2	test
3	/images
4	/product

5. Assign the URL string ID to the real server.

```
>> Real Server 1 Layer 7 commands# addlb 2
>> Real Server 1 Layer 7 commands# addlb 3
>> Real Server 1 Layer 7 commands# addlb 4
```

6. Enable the exclusionary string matching option.

```
>> Real Server 1 Layer 7 commands# exclude enable
```

If you configured an "any" string and enabled the exclusion option, the server does not handle any requests. This has the same effect as disabling the server.

Regular Expression Matching

Regular expressions are used to describe patterns for string matching. They enable you to match the exact string, such as URLs, hostnames, or IP addresses. It is a powerful and effective way to express complex rules for Layer 7 string matching. Both Layer 7 HTTP SLB and cache redirection can use regular expressions as a resource. Configuring regular expressions can enhance content-based load balancing in the following areas:

- HTTP header matching
- URL matching

Standard Regular Expression Characters

Table 68 includes a list of standard regular expression special characters that are supported by Alteon:

Table 68: Standard Regular Expression Special Characters

Construction	Description
*	Matches any string of zero or more characters
.	Matches any single character
+	Matches one or more occurrences of the pattern it follows
?	Matches zero or one occurrences of its followed pattern
$	Matches the end of a line
\	Escape the following special character
[abc]	Matches any of the single character inside the bracket
[^abc]	Matches any single character *except* those inside the bracket
^	Matches the pattern exactly only if it appears at the beginning of a line

Use the following rules when defining patterns for string matching:

- Only one layer of parenthesis is supported.
- Only a single "$" (match at end of line) is supported, which must appear at the end of the string. For example, "abc$*def" is not supported.
- The size of the user input string must be 40 characters or less.
- The size of the regular expression structure after compilation cannot exceed 43 bytes for load-balancing strings, and 23 bytes for cache redirection. The size of regular expressions after compilation varies, based on the regular expression characters used in the user input string.
- Use "/" at the beginning of the regular expression. Otherwise a regular expression will have "*" prefixed to it. For example, "html/*\.htm" appears as "*html/*\.htm".
- Incorrectly or ambiguously formatted regular expressions are rejected instantly. For example:
  - Where a "+" or "?" follows a special character, such as the "*" character.
  - A single "+" or "?" sign.
  - Unbalanced brackets and parenthesis.

Configuring Regular Expressions

The regular expression feature is applicable to both path strings used for URL-based server load balancing, and expression strings used for URL-based application redirection.

To configure regular expressions

```
>> # /cfg/slb/layer7/slb/addstr
```

As a result, both HTTP SLB and application redirection can use regular expression as the resource.

Note: The more complex the structure of the string, the longer it will take for the server to load balance the incoming packets.

Content Precedence Lookup

The Layer 7 Precedence Lookup feature enables you to give precedence to one Layer 7 parameter over another, and selectively decide which parameter should be analyzed first.

You can combine up to two Layer 7 load balancing mechanisms. You can specify which types of Layer 7 content to examine, the order in which they are examined, and a logical operator (and/or) for their evaluation.

The following Layer 7 content types can be specified:

- URL SLB
- HTTP Host
- Cookie
- Browsers (user agent)
- URL hash
- Header hash

Using these content types with the **and** and **or** operators, Alteon is configured to refine HTTP-based server load-balancing multiple times on a single client HTTP request in order to bind it to an appropriate server. Typically, when you combine two content types with an operator (and/or), URL hash and header hash are used in combination with host, cookie, or browser content types.

For example, the following types of load balancing can be configured:

- Virtual host and/or URL-based load balancing
- Cookie persistence and URL-based load balancing
- Cookie load balancing and/or URL-based load balancing
- Cookie persistence and HTTP SLB together in the same service
- Multiple HTTP SLB process type on the same service

Note: Cookie persistence can also be combined with the Layer 7 content types. For more information on cookie persistence, see Persistence, page 583

The following are example scenarios for which to use the Content Precedence Lookup feature:

- If the client request is sent without a cookie and if no HTTP SLB is configured, then Alteon binds the request to the real server using normal SLB.
- If the client request is sent without a cookie, but HTTP SLB is configured on Alteon, then the request is bound to real server based on HTTP SLB.
- If the client request is sent with a cookie, and a real server associated to the cookie is found in the local session table, then the request stays bound to that real server.

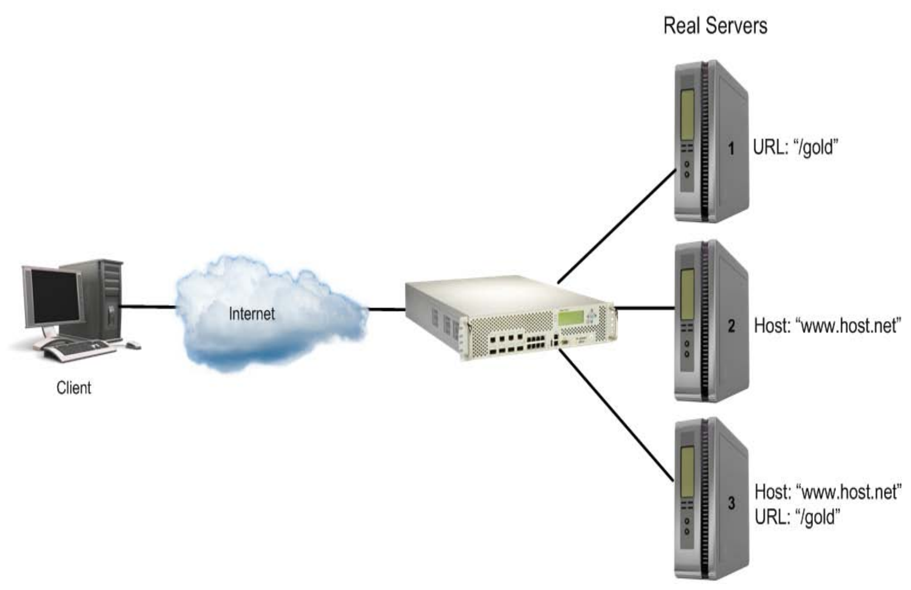
Requirements

- Enable Direct Access Mode (DAM), or configure proxy IP address if DAM is disabled.
- Enable delayed binding.

Using the or / and Operators

Figure 135 - Content Precedence Lookup Protectors Example, page 805 illustrates a network with Real Servers 1 and 3 configured for URL SLB, and Real Servers 2 and 3 configured for HTTP Host SLB.

Figure 135: Content Precedence Lookup Protectors Example



If you have configured Content Precedence Lookup with the **or** and **and** operators, the request from the client is as follows:

- **HTTP Host or URL SLB**—The HTTP Host header takes precedence because it is specified first. If there is no Host Header information, and because **or** is the operator, the URL string is examined next.
  - — If a request from a client contains no Host Header but has a URL string (such as "/gold"), the request is load balanced on Server 1 or Server 3.
  - — If a request from a client contains a Host Header, then the request is load balanced between Server 2 and Server 3. The URL string is ignored because the HTTP Host was specified and matched first.

- **HTTP Host and URL SLB**—The HTTP Host header takes precedence because it is specified first. Because **and** is the operator, both a Host Header and URL string are required. If either is not available, the request is dropped.
  - If a request from a client contains a URL string (such as "/gold") but not a Host Header, it is not served by any real server.
  - If a request from a client contains a URL string (such as "/gold") and Host Header, it is served only by real server 3.

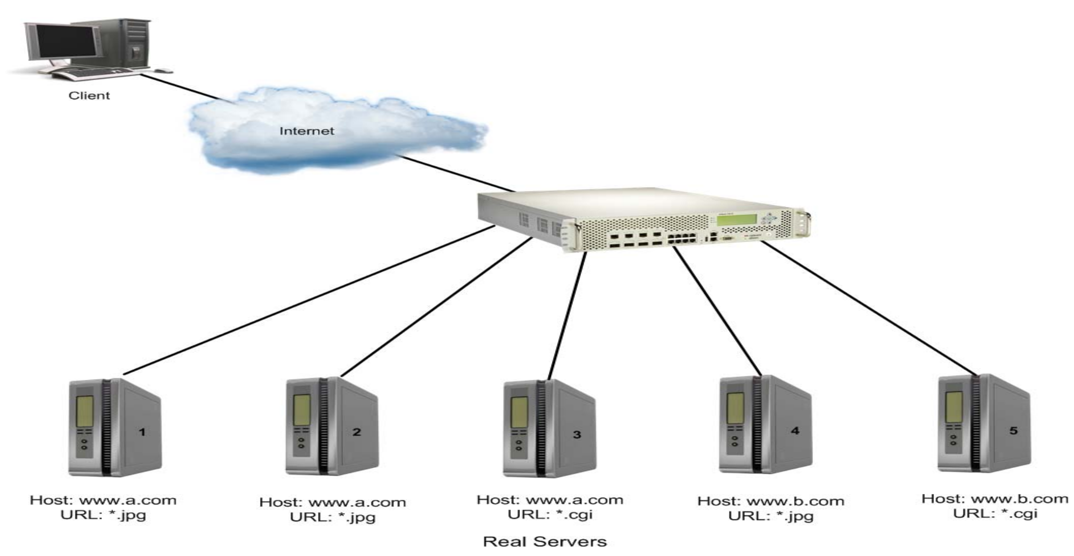
Assigning Multiple Strings

Figure 136 - Content Precedence Lookup Multiple Strings Example, page 806 illustrates an example of a company providing content for two large customers: Customers A and B. Customer A uses www.a.com as their domain name and Customer B uses www.b.com.

The company has a limited number of public IP addresses and wants to assign them on a very conservative basis. As a result, the company implements virtual hosting by advertising a single virtual server IP address that includes both customers' Web sites. Additionally, the hosting company assigns only one service (HTTP port 80) to support the virtual server.

The virtual hosting company wants to maintain the flexibility to allow different types of content to be placed on different servers. To make most efficient use of their server resources, they separate their servers into two groups, using their fastest servers to process dynamic content (such as.cgi files) and their slower servers to process all static content (such as .jpg files).

Figure 136: Content Precedence Lookup Multiple Strings Example



To configure Content Precedence Lookup for this example, the hosting company groups all the real servers into one real server group even though different servers provide services for different customers and different types of content. In this case, the servers are set up for the purpose as illustrated in Table 69.

Table 69: Real Server Content

Server	Customer	Content
Server 1	Customer A	Static .jpg files
Server 2	Customer A	Static .jpg files

Table 69: Real Server Content

Server	Customer	Content
Server 3	Customer A	Dynamic .cgi files
Server 4	Customer B	Static .jpg files
Server 5	Customer B	Dynamic .cgi files

When a client request is received with **www.a.com** in the Host Header and **.jpg** in the URL, the request is load balanced between Server 1 and Server 2. For this configuration to work properly, you must assign multiple strings (a Host Header string and a URL string) for each real server.

String Case Insensitivity

By default, Alteon supports case-sensitive matching when performing lookup of Layer 7 string content.

For example, if the following strings were configured for a real server, any incoming request containing "**GET /Default.asp**" would not bind to string 1 because of the capitalized D in Default.asp:

```
1. default.asp
2. search.asp
```

String case sensitivity may be disabled, so that any incoming request containing **GET /Default.asp**, **GET /DEFAULT.ASP**, and other case combinations, all map to string 1.

```
>> #  /cfg/slb/layer7/slb/case disable
```

Configurable HTTP Methods

Various types of HTTP methods to be processed by the Layer 7 engine are configurable.

To view the currently supported HTTP methods

```
>> # /cfg/slb/layer7/slb/cur

HTTP method types:
 1 GET           2 POST
 3 HEAD          4 BCOPY
 5 BMOVE         6 BDELETE
 7 BPROPPATCH    8 COPY
 9 CONNECT      10 DELETE
11 LINK         12 MKCOL
13 MOVE         14 OPTIONS
15 POLL         16 PUT
17 PROPFIND     18 PROPPATCH
19 SEARCH       20 SUBSCRIBE
21 TRACE        22 UNLINK
```

To add an HTTP method type

Select the method by its index number from the list in To view the currently supported HTTP methods, page 807.

```
>> #  /cfg/slb/layer7/slb/addmeth 2
```

The list of supported HTTP methods is updated regularly in Alteon as the HTTP protocol evolves.

Appendix B – Content-Intelligent Server Load Balancing Not Using Layer 7 Content Switching Rules

Alteon lets you load balance HTTP requests based on different HTTP header information, such as the "Cookie:" header for persistent load balancing, the "Host:" header for virtual hosting, or the "User-Agent" for browser-smart load balancing.

Note: When Layer 7 load balancing is configured, Alteon does not support IP fragments. If IP fragments were supported in this mode, Alteon would have to buffer, re-assemble, and inspect packets before making a forwarding decision.

URL-Based Server Load Balancing

URL-based SLB lets you optimize resource access and server performance. Content dispersion can be optimized by making load-balancing decisions on the entire path and filename of each URL.

Note: Both HTTP 1.0 and HTTP 1.1 requests are supported.

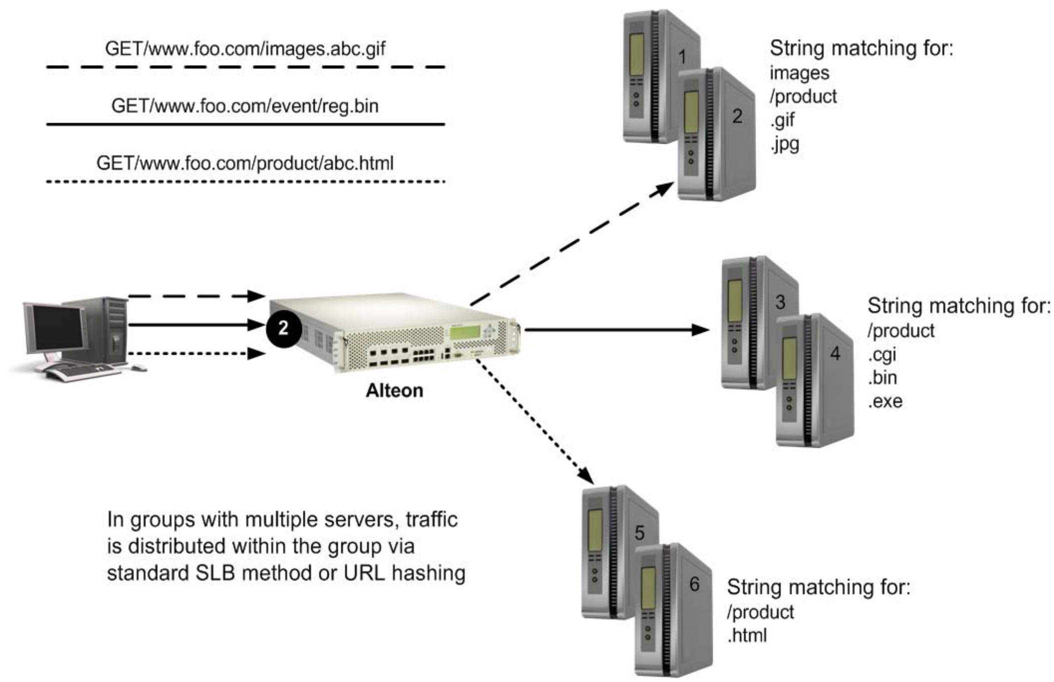
For URL matching you, can configure up to 1024 strings comprised of 40 bytes each. Each URL request is then examined against the URL strings defined for each real server. URL requests are load balanced among multiple servers matching the URL, according to the load-balancing metric configured for the real server group (**leastconns** is the default).

Consider an example where the following criteria are specified for content load balancing:

- Requests with ".cgi" in the URL are forwarded to Real Servers 3 and 4.
- Requests with the string "images" in the URL are sent to Real Servers 1 and 2.
- Requests with URLs starting with "/product:" are sent to Real Servers 2, 3, and 5.

Requests containing URLs with anything else are sent to Real Servers 1, 2, 3, and 4. These servers have been defined with the "any" string.

Figure 137: Requests with ".cgi" in the URL



Configuring URL-Based Server Load Balancing

To configure URL-based SLB

1. Before you can configure SLB string-based load balancing, ensure that Alteon has already been configured for basic SLB with the following tasks:

   Note: When URL-based SLB is used in an active/active redundant setup, use a proxy IP address instead of Direct Access Mode (DAM) to enable the URL parsing feature.

   — Assign an IP address to each of the real servers in the server pool.
   — Define an IP interface.
   — Define each real server.
   — Define a real server group and set up health checks for the group.
   — Define a virtual server on virtual port 80 (HTTP), and assign the real server group to service it.
   — Enable SLB.
   — Enable client processing on the port connected to the clients.

   For information on how to configure your network for SLB, see Server Load Balancing, page 165.

2. Define the strings to be used for URL load balancing.

```
>> # /cfg/slb/layer7/slb/addstr | remstr <l7lkup | pattern>
```

— addstr—Add string or a pattern.

— remstr—Remove string or a pattern.

A default string **any** indicates that the particular server can handle all URL or cache requests. Refer to the following examples:

- Example 1: String with the Forward Slash (/), page 811
- Example 2: String without the Forward Slash (/), page 811
- Example 3: String with the Forward Slash (/) Only, page 811

Example 1: String with the Forward Slash (/)

A string that starts with a forward slash (/), such as "**/images**," indicates that the server processes requests that start out with the "**/images**" string only.

The **/images** string allows the server to process these requests:

- /images/product/b.gif
- /images/company/a.gif
- /images/testing/c.jpg

This string would not allow the server to process these requests, however:

- /company/images/b.gif
- /product/images/c.gif
- /testing/images/a.gif

Example 2: String without the Forward Slash (/)

A string that does not start with a forward slash (/) indicates that the server will process any requests that contain the defined string.

The **images** string allows the server to process these requests:

- /images/product/b.gif
- /images/company/a.gif
- /images/testing/c.jpg
- /company/images/b.gif
- /product/images/c.gif
- /testing/images/a.gif

Example 3: String with the Forward Slash (/) Only

If a server is configured with the load balance string (/) only, it will only handle requests to the root directory.

The server would process any request to items in the root directory such as the following:

- /
- /index.htm
- /default.asp

- /index.shtm

3. Apply and save your configuration changes.
4. Identify the defined string IDs.

```
>> # /cfg/slb/layer7/slb/cur
```

For easy configuration and identification, each defined string is assigned an ID number, as shown below:

ID	SLB String
1	any
2	.gif
3	/sales
4	/xitami
5	/manual
6	.jpg

5. Configure one or more real servers to support URL-based load balancing.

Add the defined strings to the real server, where *ID* is the identification number of the defined string.

```
>> # /cfg/slb/real 2/layer7/addlb <ID>
```

Note: If you do not add a defined string (or add the defined string **any**) the server handles any request.

A server can have multiple defined strings such as:

— /images
— /sales
— .gif

With these defined strings, this particular server can handle requests that start with **/images** or **/sales** and any requests that contain **.gif**.

6. Enable SLB.

```
>> # /cfg/slb/on
```

7. Enable DAM or configure proxy IP addresses and enable proxy on the client port.

DAM and proxy IPs allow you to perform port mapping for URL load balancing.

— Enable DAM

```
>> # /cfg/slb/adv/direct ena
```

— Configure a proxy IP address and enable proxy on the client port.

```
>> # /cfg/slb/direct dis
>> # /cfg/slb/pip
```

```
>> Proxy IP Address# add 12.12.12.12
>> Proxy IP Address# type port                    (Use port-based proxy IP)
>> # /cfg/slb/port 2/proxy ena                    (Enable proxy on client port)
```

For more information on proxy IP addresses, see Client Network Address Translation (Proxy IP), page 190.

8. Enable URL-based SLB on the virtual servers.

```
>> # /cfg/slb/virt <virtual server number> /service 80/http/httpslb urlslb
```

Statistics for URL-Based Server Load Balancing

To show the number of hits to the SLB or cache server

```
>> # /stats/slb/layer7/str
```

The following are sample statistics generated by this command:

ID	SLB String	Hits
1	any	73881
2	.gif	0
3	/sales	0
4	/xitami	162102
5	/manual	0
6	.jpg	0

Virtual Hosting

Alteon allows individuals and companies to have a presence on the Internet in the form of a dedicated Web site address. For example, you can have a "www.site-a.com" and "www.site-b.com", instead of "www.hostsite.com/site-a" and "www.hostsite.com/site-b."

Service providers, on the other hand, do not want to deplete the pool of unique IP addresses by dedicating an individual IP address for each home page they host. By supporting an extension in HTTP 1.1 to include the host header, Alteon enables service providers to create a single virtual server IP address to host multiple Web sites per customer, each with their own host name.

Note: For SLB, one HTTP header is supported per virtual server.

The following list provides more details on virtual hosting with configuration information:

- An HTTP/1.0 request sent to an origin server (*not* a proxy server) is a partial URL instead of a full URL.

  An example of the request that the origin server would see as follows:

  ```
  GET /products/Alteon/ HTTP/1.0User-agent: Mozilla/3.0Accept: text/html, image/gif, image/jpeg
  ```

The GET request does not include the hostname. From the TCP/IP headers, the origin server knows the requests hostname, port number, and protocol.

- With the extension to HTTP/1.1 to include the HTTP HOST: header, the above request to retrieve the URL www.radware.com/products/Alteon would look like this:

```
GET /products/Alteon/ HTTP/1.1
Host: www.radware.com
User-agent: Mozilla/3.0
Accept: text/html, image/gif, image/jpeg
```

The Host: header carries the hostname used to generate the IP address of the site.

- Based on the Host: header, Alteon forwards the request to servers representing different customer Web sites.
- The network administrator needs to define a domain name as part of the 128 supported URL strings.
- Alteon performs string matching. That is, the string "radware.com" or "http://www.radware.com/" " matches ""http://www.radware.com/".

Virtual Hosting Configuration Overview

The following is the sequence of events for configuring virtual hosting based on HTTP Host: headers:

1. The network administrator defines a domain name as part of the 128 supported URL strings.

   Both domain names "www.company-a.com" and "www.company-b.com" resolve to the same IP address. In this example, the IP address is for a virtual server on Alteon.
2. "www.company-a.com" and "www.company-b.com" are defined as URL strings.
3. Server Group 1 is configured with Servers 1 through 8.

   Servers 1 through 4 belong to "www.company-a.com" and Servers 5 through 8 belong to "www.company-b.com."
4. The network administrator assigns string "www.company-a.com" to Servers 1 through 4 and string "www.company-b.com" to Servers 5 through 8.
5. Alteon inspects the HTTP host header in requests received from the client.
   - If the host header is "www.company-a.com," Alteon directs requests to one of the Servers 1 through 4.
   - If the host header is "www.company-b.com," Alteon directs requests to one of the Servers 5 through 8.

Configuring the Host Header for Virtual Hosting

To support virtual hosting, configure Alteon for Host header-based load balancing

1. Before you can configure host header-based SLB, ensure that Alteon has already been configured for basic SLB:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface.
   - Define each real server.
   - Assign servers to real server groups.
   - Define virtual servers and services.

   For information on how to configure your network for SLB, see Server Load Balancing, page 165.

2. Turn on URL parsing for the virtual server for virtual hosting.

`>> # /cfg/slb/virt 1`	(Select the virtual IP for host header-based SLB)
`>> Virtual Server 1 # service 80`	(Select the HTTP service)
`>> Virtual Server 1 http Service # http/httpslb host`	

3. Define the host names.

```
>> # /cfg/slb/layer7/slb/addstr "www.customer1.com"
>> Server Loadbalance Resource# addstr "www.customer2.com"
>> Server Loadbalance Resource# addstr "www.customer3.com"
```

4. Configure the real servers to handle the appropriate load balancing strings. To add a defined string where *ID* is the identification number of the defined string.

`>># /cfg/slb/real 2`	(Select the real server)
`>> Real Server 2 # Layer7`	
`>> Real Server 2 Layer 7 Commands # addlb <ID>`	(Specify the string ID)

Note: The server handles any request if no string or the string any is defined.

Cookie-Based Preferential Load Balancing

Cookies can be used to provide preferential services for customers, ensuring that certain users are offered better access to resources than other users when site resources are scarce. For example, a Web server could authenticate a user via a password and then set cookies to identify them as "Gold," "Silver," or "Bronze" customers. Using cookies, you can distinguish individuals or groups of users and place them into groups or communities that get redirected to better resources and receive better services than all other users.

Note: Cookie-based persistent load balancing is described in Persistence, page 583.

Cookie-based preferential services enable the following support:

- Redirect higher priority users to a larger server or server group.
- Identify a user group and redirect them to a particular server.
- Serve content-based on user identity.
- Prioritize access to scarce resources on a Web site.
- Provide better services to repeat customers, based on access count.

Clients that receive preferential service can be distinguished from other users by one of the following methods:

- **Individual User**—Distinguish a specific user by IP address, login authentication, or permanent HTTP cookie.
- **User Communities**—Identify some set of users, such as "Premium Users" for service providers who pay higher membership fees than "Normal Users" by source address range, login authentication, or permanent HTTP cookie.

- **Applications**—Identify users by the specific application they are using. For example, priority can be given to HTTPS traffic that is performing credit card transactions versus HTTP browsing traffic.
- **Content**—Identify users by the specific content they are accessing.

Based on one or more of these criteria, you can load balance requests to different server groups.

Configuring Cookie-Based Preferential Load Balancing

To configure cookie-based preferential load balancing

1. Before you can configure header-based load balancing, ensure that Alteon has already been configured for basic SLB with the following tasks:
   - Assign an IP address to each of the real servers in the server pool.
   - Define an IP interface.
   - Define each real server.
   - Assign servers to real server groups.
   - Define virtual servers and services.

   For information on how to configure your network for SLB, see Server Load Balancing, page 165.
2. Turn on URL parsing for the virtual server.

```
>> # /cfg/slb/virt 1
>> Virtual Server 1 # service 80
>> Virtual Server 1 http Service # http
>> HTTP Load Balancing# httpslb
>> Application:
>> urlslb|host|cookie|browser|urlhash|headerhash|version|others|none
>> Select Application:cookie
>> Operation: and|or|none
>> Select Operation: ena
>> Enter Cookie Name: sid
>> Enter the starting point of the Cookie value [1-64]: 1
>> Enter the number of bytes to extract [1-64]: 6
>> Look for Cookie in URI [e:d]: d
```

   In this example
   - *sid* is the cookie name
   - *1* is the offset (the starting position of the value to be used for hashing)
   - *6* is the length (the number of bytes in the cookie value)
   - *d* looks for the cookie in the cookie header instead of the URI (disables searching for cookie in the URI)
3. Define the cookie values.

```
>> # /cfg/slb/layer7/slb/addstr "Gold"
>> # addstr "Silver"
>> # addstr "Bronze"
```

   Because a session cookie does not exist in the first request of an HTTP session, a default server or **any** server is needed to assign cookies to a **None** cookie HTTP request.

Example

- Real Server 1—**Gold** handles gold requests.
- Real Server 2—**Silver** handles silver request.
- Real Server 3—**Bronze** handles bronze request.
- Real Server 4—**any** handles any request that does not have a cookie or matching cookie.

With servers defined to handle the requests listed above, the following occurs:

- Request 1 comes in with no cookie; it is forwarded to Real Server 4 to get cookie assigned.
- Request 2 comes in with a "Gold" cookie; it is forwarded to Real Server 1.
- Request 3 comes in with a "Silver" cookie; it is forwarded to Real Server 2.
- Request 4 comes in with a "Bronze" cookie; it is forwarded to Real Server 3.
- Request 5 comes in with a "Titanium" cookie; it is forwarded to Real Server 4, since it does not have an exact cookie match (matches with "any" configured at Real Server 4).

4. Configure the real servers to handle the appropriate load balancing strings. Add a defined string, where *ID* is the identification number of the string:

```
>> # /cfg/slb/real 2/layer7/addlb <ID>
```

Note: If you do not add a defined string (or add the defined string **any**), the server handles any request.

5. Enable DAM on Alteon or configure proxy IP addresses and enable proxy on the client port.

   To use cookie-based preferential load balancing without DAM, you must configure proxy IP addresses.

   Enable proxy load balancing on the port used for cookie-based preferential load balancing. If Virtual Matrix Architecture (VMA) is enabled, you can choose to configure the remaining ports with proxy disabled.

Browser-Smart Load Balancing

HTTP requests can be directed to different servers based on browser type by inspecting the "User-Agent" header. For example:

```
GET /products/Alteon/ HTTP/1.0
User-agent: Mozilla/3.0
Accept: text/html, image/gif, image/jpeg
```

To allow Alteon to perform browser-smart load balancing

1. Before you can configure browser-based load balancing, ensure that Alteon has already been configured for basic SLB with the following tasks:
   — Assign an IP address to each of the real servers in the server pool.
   — Define an IP interface.
   — Define each real server.
   — Assign servers to real server groups.

— Define virtual servers and services.

2. Turn on URL parsing for the virtual server for "User-Agent:" header.

```
>> # /cfg/slb/virt 1/service 80/http/httpslb browser
```

3. Define the hostnames.

```
>> # /cfg/slb/layer7/slb/addstr "Mozilla"
>> Server Loadbalance Resource# addstr "Internet Explorer"
>> Server Loadbalance Resource# addstr "Netscape"
```

4. Configure the real servers to handle the appropriate load balancing strings.

Note: If you do not add a defined string (or add the defined string any), the server handles any request.

Use the following command to add a defined string, where *ID* is the identification number of the defined string.

```
>> # /cfg/slb/real 2/layer7/addlb <ID>
```

Configure SLB Strings for HTTP Redirection

All of the following HTTP filtering redirection examples require configuring the SLB strings listed in Table 70. Each defined string has an associated ID number. A filter is then configured to redirect from one configured string ID to another.

Note: Not all strings are used in each example.

Table 70: Example HTTP Redirection Strings

ID	SLB String
1	`any, cont 256`
2	`HTTPHDR=Host:wap.example.com`
3	`HTTPHDR=Host:wap.yahoo.com`
4	`HTTPHDR=Host:wap.google.com`
5	`HTTPHDR=Host:wap.p-example.com`
6	`HTTPHDR=Host:10.168.224.227=/top`
7	`jad, cont 256`
8	`jar, cont 256`
9	`HTTPHDR=Accept:text/vnd.foo.j2me.app-descriptor`
10	`HTTPHDR=Host:mobile.example.com=/4g/w?url=$HOST_URL`
11	`HTTPHDR=Host:any`
12	`HTTPHDR=Host:any:90`
13	`HTTPHDR=Host:any:8080`
14	`HTTPHDR=X-Foo-ipaddress:10.168.100.*`
15	`HTTPHDR=Host:www.abc.com, cont 256`
16	`HTTPHDR=Host:any:443, cont 256`
17	`HTTPHDR=Host:mobile.example.com=/4g/w?url=$HOST/nava/toggle.jad, nre, cont 1024`
18	`HTTPHDR=Host:mobile.example.com=/4g/w?url=dev.example.com/$URL, nre, cont 1024`

1. Configure Alteon with the basic SLB requirements as described in Server Load Balancing Configuration Basics, page 171.
2. Configure the filter strings.

```
>> # /cfg/slb/layer7/slb/
>> Server Loadbalance Resource# addstr          (Add the first SLB string)
Enter type of string [l7lkup|pattern]:  l7lkup
Select Application (http|dns|other) [other]:
http
Configure HTTP header string? (y/n) [n] y
Enter HTTP header name: Host
Enter SLB header value string:  wap.example.com
Configure URL string? (y/n) [n] n
>> # /cfg/slb/layer7/slb/                       (Select the Server Loadbalance
                                                 Resource menu)
>> Server Loadbalance Resource# add             (Add the second SLB string)
Configure HTTP header string? [y/n] y
```

`Enter HTTP header name:`	`Host`	(Define HTTP header name Host)
`Enter SLB header value string:`	`wap.yahoo.com`	

3. Use the same commands as step 2 to configure the rest of the filter strings shown in Server Load Balancing Configuration Basics, page 171.
4. Identify the ID numbers of the defined strings.

```
>> # /cfg/slb/layer7/slb/cur
Number of entries: 1
41: any, cont 256
2: HTTPHDR=Host:wap.example.com, cont 256
3: HTTPHDR=Host:wap.yahoo.com, cont 256
4: HTTPHDR=Host:wap.google.com, cont 256
5: HTTPHDR=Host:wap.p-example.com, cont 256
6: HTTPHDR=Host:10.168.224.227=/top, cont 256
7: jad, cont 256
8: jar, cont 256
9: HTTPHDR=Accept:text/vnd.foo.j2me.app-descriptor, cont 256
10: HTTPHDR=Host:mobile.example.com=/4g/w?url=$HOST_URL, cont 256
11: HTTPHDR=Host:any, cont 256
12: HTTPHDR=Host:any:90, cont 256
13: HTTPHDR=Host:any:8080, cont 256
14: HTTPHDR=X-Foo-ipaddress:10.168.100.* , cont 256
15: HTTPHDR=Host:www.abc.com, cont 256
16: HTTPHDR=Host:any:443, cont 256
17: HTTPHDR=Host:mobile.example.com=/4g/w?url=$HOST/nava/toggle.jad, nre, cont
1024
18: HTTPHDR=Host:mobile.example.com=/4g/w?url=dev.example.com/$URL, nre, cont
1024
```

5. Configure a port for client traffic. This configuration assumes client traffic enters Alteon on port 1. Enabled filtering on the client port.

`>> /cfg/slb/port 1`	(Select the *SLB Port 1* menu)
`>> SLB port 1# filt en`	(Enable filtering on the port)
`Current port 1 filtering: disabled`	
`New port 1 filtering:     enabled`	

The basic HTTP redirection configuration is now complete and can be used for each of the redirection options described in the following sections.

Example IP based HTTP Redirection

In this example, Alteon redirects Web pages requested from a mobile phone to a specific gateway based on the client's IP address. A mobile phone is set to access its home page via the default device gateway.

The following is the client phone configuration used for the example:

```
Device Gateway IP address  10.168.107.101
Home page: http://wap.example.com
WAP port  9001, CSD number as 18881234567
username: john
```

The following rules filter client requests from different WAP gateways:

- **Filter 1**—If the client IP address is between **10.168.43.0-255** and the requested URL is **http://wap.example.com**, then redirect the client request to **http://wap.yahoo.com**.
- **Filter 2**—If the Client IP address is between **10.46.6.0.0-255** and the requested URL is **http://wap.example.com**, then redirect the client request to **http://wap.google.com**.
- **Filter 3**—If the client IP address is between **10.23.43.0- 255** and the requested URL is **http://wap.p-example.com**, then redirect the client request to **http://10.168.224.227/top**.

Assuming that each client is in a different subnet, configure Alteon with three filters to redirect client requests from each subnet, to the URLs specified above. Use the string index numbers in Table 70 - Example HTTP Redirection Strings, page 819 to configure a redirection map for each filter.

1. Identify the ID numbers of the defined strings. The strings in bold in the filters defined above are used in this example.

```
>> # /cfg/slb/layer7/slb/cur
Number of entries: 14
1: any, cont 256
2: HTTPHDR=Host:wap.example.com, cont 256
3: HTTPHDR=Host:wap.yahoo.com, cont 256
4: HTTPHDR=Host:wap.google.com, cont 256
5: HTTPHDR=Host:wap.p-example.com, cont 256
6: HTTPHDR=Host:10.168.224.227=/top, cont 256
7: jad, cont 256
8: jar, cont 256
9: HTTPHDR=Accept:text/vnd.foo.j2me.app-descriptor, cont 256
10: HTTPHDR=Host:mobile.example.com=/4g/w?url=$HOST_URL, cont 256
11: HTTPHDR=Host:any, cont 256
12: HTTPHDR=Host:any:90, cont 256
13: HTTPHDR=Host:any:8080, cont 256
14: HTTPHDR=X-Foo-ipaddress:10.168.100.* , cont 256
15: HTTPHDR=Host:www.abc.com, cont 256
16: HTTPHDR=Host:any:443, cont 256
17: HTTPHDR=Host:mobile.example.com=/4g/w?url=$HOST/nava/toggle.jad, nre, cont
1024
18: HTTPHDR=Host:mobile.example.com=/4g/w?url=
dev.example.com/$URL, nre, cont 1024
```

2. Configure Filter 1.

```
>> /cfg/slb/filt 1
>> Filter 1 # sip 10.168.43.0                    (From this source IP address range)
Current source address:     any
New pending source address: 10.168.43.0
>> Filter 1 # smask 255.255.255.0
Current source mask:     0.0.0.0
New pending source mask: 255.255.255.0
```

`>> Filter 1 # proto tcp`	(For TCP protocol traffic)
`Enter protocol or any: udp`	
`Pending new protocol:   tcp`	
`>> Filter 1 # dport http`	(To destination port HTTP)
`Current destination port or range:      any`	
`Pending new destination port or range:  http`	
`>> Filter 1 # action redir`	(Redirect the traffic)
`Current action: allow`	
`Pending new action:     redir`	
`>> Filter 1 # /cfg/slb/filt/adv/layer7`	(Access the *Advanced Layer 7* menu)
`>> Layer 7 Advanced# addrd`	
`Enter filtering string ID (1-1024) to redirect from:`	2 (Redirect string 2...)
`Enter filtering string ID (2-1024) to redirect to:`	3 (to string 3)

3. Configure Filter 2.

```
>> /cfg/slb/filt 2
>> Filter 2 # sip 10.46.6.0.0
Current source address:     any
New pending source address: 10.46.6.0.0
>> Filter 2 # smask 255.255.255.0
Current source mask:     0.0.0.0
New pending source mask: 255.255.255.0
>> Filter 2 # proto tcp
Enter protocol or any: udp
Pending new protocol:   tcp
>> Filter 2 # dport http
Current destination port or range:      any
Pending new destination port or range:  http
>> Filter 2 # action redir
Current action: allow
Pending new action:     redir
>> Filter 2 # /cfg/slb/filt/adv/layer7
>> Layer 7 Advanced# addrd
Enter filtering string ID (1-1024) to redirect from: 2
Enter filtering string ID (2-1024) to redirect to:   4
```

4. Configure Filter 3.

```
>> /cfg/slb/filt 3
>> Filter 3 # sip 10.23.43.0
Current source address:     any
New pending source address: 10.23.43.0
>> Filter 3 # smask 255.255.255.0
Current source mask:     0.0.0.0
New pending source mask: 255.255.255.0
>> Filter 3 # proto tcp
Enter protocol or any: udp
Pending new protocol:   tcp
>> Filter 3 # dport http
Current destination port or range:      any
Pending new destination port or range:  http
>> Filter 3 # action redir
Current action: allow
Pending new action:     redir
>> Filter 3 # /cfg/slb/filt/adv/layer7
>> Layer 7 Advanced# addrd
Enter filtering string ID (1-1024) to redirect from: 5
Enter filtering string ID (2-1024) to redirect to:   6
```

5. **Apply** and **save** the configuration.

Example TCP Service Port Based HTTP Redirection

In this example, Alteon redirects traffic entering Alteon on one TCP service port, and redirects it through another service port. Use the provided string index numbers to configure a redirection map for each filter.

- **Filter 4**—Configure a filter to examine the URL request **http://10.46.6.231:80/Connect1.jad** on TCP service port 80, and redirect that URL to TCP service port 90.
- **Filter 5**—Configure a filter that intercepts all traffic entering on TCP service port 80, and send it to **10.168.120.129** on TCP service port 8080.

1. Identify the ID numbers of the defined strings. The strings in bold in the filters defined above are used in this example.

```
>> # /cfg/slb/layer7/slb/cur
Number of entries: 141: any, cont 256
2: HTTPHDR=Host:wap.example.com, cont 256
3: HTTPHDR=Host:wap.yahoo.com, cont 256
4: HTTPHDR=Host:wap.google.com, cont 256
5: HTTPHDR=Host:wap.p-example.com, cont 256
6: HTTPHDR=Host:10.168.224.227=/top, cont 256
7: jad, cont 256
8: jar, cont 256
9: HTTPHDR=Accept:text/vnd.foo.j2me.app-descriptor , cont 256
10: HTTPHDR=Host:mobile.example.com=/4g/w?url=$HOST_URL, cont 256
11: HTTPHDR=Host:any, cont 256
12: HTTPHDR=Host:any:90, cont 256
13: HTTPHDR=Host:any:8080, cont 256
14: HTTPHDR=X-Foo-ipaddress:10.168.100.* , cont 256
15: HTTPHDR=Host:www.abc.com, cont 256
16: HTTPHDR=Host:any:443, cont 256
17: HTTPHDR=Host:mobile.example.com=/4g/w?url=
$HOST/nava/toggle.jad, nre, cont 1024
18: HTTPHDR=Host:mobile.example.com=/4g/w?url=dev.example.com/$URL, nre, cont
1024
```

2. Configure Filter 4.

```
>> /cfg/slb/filt 4
>> Filter 4 # dip 10.46.6.231
Current destination address:     any
New pending destination address: 10.46.6.231
>> Filter 4 # smask 255.255.255.255
Current source mask:     0.0.0.0
New pending source mask: 255.255.255.255
>> Filter 4 # proto tcp
Enter protocol or any: udp
Pending new protocol:   tcp
>> Filter 4 # dport http
Current destination port or range:      any
Pending new destination port or range:  http
>> Filter 4 # action redir
Current action: allow
Pending new action:     redir
>> Filter 4 # /cfg/slb/filt/adv/layer7
>> Layer 7 Advanced# addrd
Enter filtering string ID (1-1024) to redirect from: 11
Enter filtering string ID (2-1024) to redirect to:   12
```

3. Configure Filter 5.

```
>> /cfg/slb/filt 5
>> Filter 5 # dip 10.46.6.231
Current destination address:     any
New pending destination address: 10.46.6.231
>> Filter 5 # smask 255.255.255.255
Current source mask:     0.0.0.0
New pending source mask: 255.255.255.255
>> Filter 5 # proto tcp
Enter protocol or any: udp
Pending new protocol:   tcp
>> Filter 5 # dport http
Current destination port or range:      any
Pending new destination port or range:  http
>> Filter 5 # action redir
Current action: allow
Pending new action:     redir
>> Filter 5 # /cfg/slb/filt/adv/layer7
>> Layer 7 Advanced# addrd
Enter filtering string ID (1-1024) to redirect from: 11
Enter filtering string ID (2-1024) to redirect to:   13
```

4. Apply and save the configuration.

Example MIME Type Header-Based Redirection

In this example, Alteon receives a URL request from a mobile client and examines the Multipurpose Internet Mail Extensions (MIME) type header in the URL. If the URL contains a pre-defined MIME type, text, or URL, Alteon replaces the URL. Use the string index numbers to configure a redirection map for the filter.

- **Filter 6**—The mobile client executes a request for a URL **http://dev.example.com/java/toggle.jad**. If the MIME type is **text/vnd.foo.j2me.app-descriptor**, or if the URL contains **jad** or **jar** as an extension, it will replace the URL with: **http://mobile.example.com/4g/w?url=dev.example.com/nava/toggle.jad.**

1. Identify the ID numbers of the defined strings. The strings in bold are used in this example.

```
>> # /cfg/slb/layer7/slb/cur
Number of entries: 14
1: any, cont 256
2: HTTPHDR=Host:wap.example.com, cont 256
3: HTTPHDR=Host:wap.yahoo.com, cont 256
4: HTTPHDR=Host:wap.google.com, cont 256
5: HTTPHDR=Host:wap.p-example.com, cont 256
6: HTTPHDR=Host:10.168.224.227=/top, cont 256
7: jad, cont 256
8: jar, cont 256
9: HTTPHDR=Accept:text/vnd.foo.j2me.app-descriptor , cont 256
10: HTTPHDR=Host:mobile.example.com=/4g/w?url=$HOST_URL, cont  256
11: HTTPHDR=Host:any, cont 256
12: HTTPHDR=Host:any:90, cont 256
13: HTTPHDR=Host:any:8080, cont 256
14: HTTPHDR=X-Foo-ipaddress:10.168.100.* , cont 256
15: HTTPHDR=Host:www.abc.com, cont 256
16: HTTPHDR=Host:any:443, cont 256
17: HTTPHDR=Host:mobile.example.com=/4g/w?url=$HOST/nava/toggle.jad, nre, cont
1024
18: HTTPHDR=Host:mobile.example.com=/4g/w?url=dev.example.com/$URL, nre, cont
1024
```

2. Configure Filter 6. The filter intercepts string 7, 8, and 9 and then redirects them based on strings 10, 17, and 18 information. The $HOST_URL is replaced with the incoming request from the HOST and URL strings. The $HOST is replaced with the incoming request from HOST string. The $URL is replaced with the incoming request from the URL string.

```
>> /cfg/slb/filt 6
>> Filter 6 # dip 10.46.6.231
Current destination address:     any
New pending destination address: 10.46.6.231
>> Filter 6 # smask 255.255.255.255
Current source mask:     0.0.0.0
New pending source mask: 255.255.255.255
>> Filter 6 # proto tcp
Enter protocol or any: udp
Pending new protocol:   tcp
>> Filter 6 # dport http
Current destination port or range:      any
Pending new destination port or range:  http
>> Filter 6 # action redir
Current action: allow
Pending new action:     redir
>> Filter 6 # /cfg/slb/filt/adv/layer7
>> Layer 7 Advanced# addrd
Enter filtering string ID (1-1024) to redirect from: 7
Enter filtering string ID (2-1024) to redirect to:   10
>> Layer 7 Advanced# addrd
Enter filtering string ID (1-1024) to redirect from: 8
Enter filtering string ID (2-1024) to redirect to:   17
>> Layer 7 Advanced# addrd
Enter filtering string ID (1-1024) to redirect from: 9
Enter filtering string ID (2-1024) to redirect to:   18
```

3. **Apply** and **save** the configuration.

```
>> Layer 7 Advanced# apply
>> Layer 7 Advanced# save
```

Example URL-Based Redirection

A request for a URL can be redirected to another URL as follows:

— **Filter 7**—URL **http://wap.example.com** is redirected to **http://10.168.224.227/top**.

1. Identify the ID numbers of the defined strings. The strings in bold in the filter defined above are used in this example.

```
>> # /cfg/slb/layer7/slb/cur
Number of entries: 14
1: any, cont 256
2: HTTPHDR=Host:wap.example.com, cont 256
3: HTTPHDR=Host:wap.yahoo.com, cont 256
4: HTTPHDR=Host:wap.google.com, cont 256
5: HTTPHDR=Host:wap.p-example.com, cont 256
6: HTTPHDR=Host:10.168.224.227=/top, cont 256
7: jad, cont 256
8: jar, cont 256
9: HTTPHDR=Accept:text/vnd.foo.j2me.app-descriptor, cont 256
10: HTTPHDR=Host:mobile.example.com=/4g/w?url=$HOST_URL, cont 256
11: HTTPHDR=Host:any, cont 256
12: HTTPHDR=Host:any:90, cont 256
13: HTTPHDR=Host:any:8080, cont 256
14: HTTPHDR=X-Foo-ipaddress:10.168.100.* , cont 256
15: HTTPHDR=Host:www.abc.com, cont 256
16: HTTPHDR=Host:any:443, cont 256
17: HTTPHDR=Host:mobile.example.com=/4g/w?url=$HOST/nava/toggle.jad, nre, cont
1024
18: HTTPHDR=Host:mobile.example.com=/4g/w?url=dev.example.com/$URL, nre, cont
1024
```

2. Configure Filter 7 to redirect the URL as described above. By default, filter protocol is **any**. Change it to **udp**.

```
>> /cfg/slb/filt 7
>> Filter 7 # dip 10.46.6.231
Current destination address:     any
New pending destination address: 10.46.6.231
>> Filter 7 # smask 255.255.255.255
Current source mask:     0.0.0.0
New pending source mask: 255.255.255.255
>> Filter 7 # proto tcp
Enter protocol or any: udp
Pending new protocol:   tcp
>> Filter 7 # dport httpCurrent destination port or range:      any
Pending new destination port or range:  http
>> Filter 7 # action redirCurrent action: allow
Pending new action:     redir
>> Filter 7 # /cfg/slb/filt/adv/layer7
>> Layer 7 Advanced# addrd
Enter filtering string ID (1-1024) to redirect from: 2
Enter filtering string ID (2-1024) to redirect to:   6
```

3. **Apply** and **save** the configuration.

```
>> Layer 7 Advanced# apply
>> Layer 7 Advanced# save
```

Example Source IP from HTTP Header and Host Header-Based Redirection

In this example, a filter is configured as follows:

— **Filter 8**—If **X-Foo-ipaddress: 10.168.100.*** and the request is to **http://wap.example.com**, then redirect the request to **wap.yahoo.com**.

1. Identify the ID numbers of the defined strings. The strings in bold in the filter defined above are used in this example.

```
>> # /cfg/slb/layer7/slb/cur
Number of entries: 14
1: any, cont 256
2: HTTPHDR=Host:wap.example.com, cont 256
3: HTTPHDR=Host:wap.yahoo.com, cont 256
4: HTTPHDR=Host:wap.google.com, cont 256
5: HTTPHDR=Host:wap.p-example.com, cont 256
6: HTTPHDR=Host:10.168.224.227=/top, cont 256
7: jad, cont 256
8: jar, cont 256
9: HTTPHDR=Accept:text/vnd.foo.j2me.app-descriptor , cont 256
10: HTTPHDR=Host:mobile.example.com=/4g/w?url=$HOST_URL, cont 256
11: HTTPHDR=Host:any, cont 256
12: HTTPHDR=Host:any:90, cont 256
13: HTTPHDR=Host:any:8080, cont 256
14: HTTPHDR=X-Foo-ipaddress:10.168.100.* , cont 256
15: HTTPHDR=Host:www.abc.com, cont 256
16: HTTPHDR=Host:any:443, cont 256
17: HTTPHDR=Host:mobile.example.com=/4g/w?url=$HOST/nava/toggle.jad, nre, cont
1024
18: HTTPHDR=Host:mobile.example.com=/4g/w?url=dev.example.com/$URL, nre, cont
1024
```

2. Configure Filter 8 redirect URL as described above. By default, filter protocol is **any**. Change it to **udp**.

```
>> /cfg/slb/filt 8
>> Filter 8 # sip 10.46.6.231
Current source address:     any
New pending source address: 10.46.6.231
>> Filter 8 # smask 255.255.255.255
Current source mask:     0.0.0.0
New pending source mask: 255.255.255.255
>> Filter 8 # proto tcp
Enter protocol or any: udp
Pending new protocol:   tcp
>> Filter 8 # dport http
Current destination port or range:      any
Pending new destination port or range:  http
>> Filter 8 # action redir
Current action: allow
Pending new action:     redir
>> Filter 8 # /cfg/slb/filt/adv/layer7
>> Layer 7 Advanced# addrd
Enter filtering string ID (1-1024) to redirect from: 2
Enter filtering string ID (2-1024) to redirect to:   14
```

3. Apply and save the configuration.

```
>> Layer 7 Advanced# apply
>> Layer 7 Advanced# save
```

Example HTTP to HTTPS Redirection

To redirect HTTP requires to HTTPS connections, the following filters can be set:

- **Filter 9**—Configure a filter that intercepts HTTP traffic to **http://www.abc.com** and redirects it to **https://www.abc.com**
- **Filter 10**—Configure a filter that intercepts HTTP traffic directed to **205.10.10.10** and redirects it to HTTPS.

1. Define Layer 7 strings and identify their ID numbers. The strings in bold in the filters defined above are used in this example

```
/c/slb/layer7/slb/cur
          ren 2 "HTTPHDR=Host:any"
          ren 3 "HTTPHDR=Host:www.abc.com,"
          ren 4 "HTTPHDR=Host:any:443"
```

2. Configure Filter 9 and Filter 10.

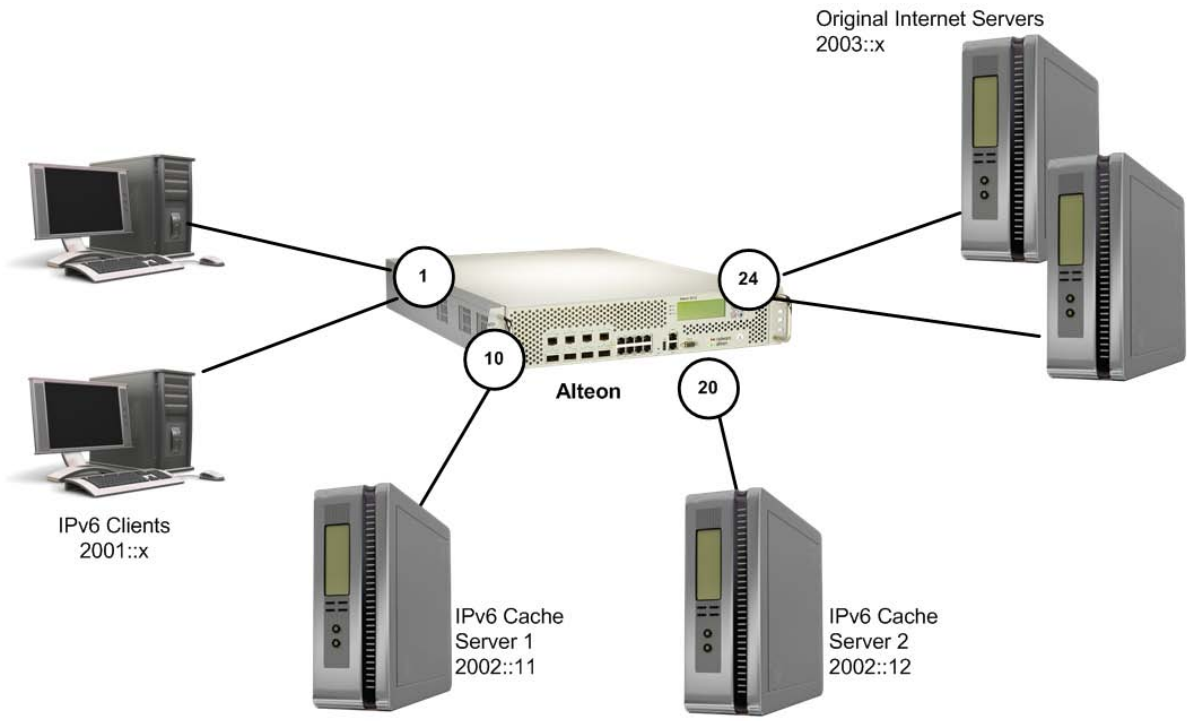
```
/c/slb/filt 9
        ena
        action redir
        ipver v4
        proto tcp
        dport http
/c/slb/filt 9/adv/layer7
        l7lkup ena
        addrd 3>4

/c/slb/filt 10
        ena
        action redir
        ipver v4
        dip 205.10.10.10
        proto tcp
        dport http
/c/slb/filt 10/adv/layer7
        l7lkup ena
        addrd 2>4
```

3. **Apply** and **save** the configuration.

Example IPv6 Redirection Filter

Figure 138 - TCP Service Port Based HTTP Redirection, page 832 illustrates an IPv6 redirection filter:

Figure 138: TCP Service Port Based HTTP Redirection



1. Configure the client VLAN.

```
>> Main# /cfg/l2/vlan 2/en/name "Client_VLAN"/add 1
```

2. Configure the client interface.

```
>> Main# /cfg/l3/if 2/en/vlan 2/ipv v6/add 2001::1/mask 64
```

3. Configure the cache server VLAN.

```
>> Main# /cfg/l2/vlan 3/en/name "Cache_VLAN"/add 10/add 20
```

4. Configure the cache server interface.

```
>> Main# /cfg/l3/if 3/en/vlan 3/ipv v6/add 2002::1/mask 64
```

5. Configure the original server VLAN (VLAN to Internet).

```
>> Main# /cfg/l2/vlan 4/en/name "Internet_VLAN"/add 24
```

6. Configure the interface to the Internet.

```
>> Main# /cfg/l3/if 4/en/vlan 4/ipv v6/add 2003::1/mask 64
```

7. Enable SLB.

```
>> Main# /cfg/slb/on
```

8. Configure Cache Server 1.

```
>> Main# /cfg/slb/re 1/en/ipv v6/rip 2002::11
```

9. Configure Cache Server 2.

```
>> Main# /cfg/slb/re 2/en/ipv v6/rip 2002::12
```

10. Add the two cache servers to the real group.

```
>> Main# /cfg/slb/gr 1/ipv v6/add 11/add 12
```

11. Configure the IPv6 redirection filter to redirect all HTTP traffic to the cache servers.

```
>> Main# /cfg/slb/fi 1/en/name "IPv6_HTTP_Redir_Filter"/ipv v6/act
redir/proto tcp/dport http/group 1/
```

12. Configure IPv6 default filter to allow other traffic.

```
>> Main#  /cfg/slb/fi 2048/en/name "IPv6_Allow_Filter"/ipv v6/act allow
```

13. Enable filter processing on client ports and add the two filters to the client ports.

```
>> Main# /cfg/slb/po 1/fi en/add 1/add 2048
```

14. Apply the configuration.

```
>> Main# apply
>> Main# save
```

Appendix C – IPv6

This appendix describes the basic configuration and management of IPv6. For IPv6 implementation with specific Alteon features, refer to the appropriate chapters for details on the level of support.

This appendix includes the following topics:

IPv4 versus IPv6

Internet Protocol version 6 (IPv6) is a network layer protocol for packet-switched internetworks. It is designated as the successor of IPv4, the current version of the Internet Protocol, for general use on the Internet.

The main improvement brought by IPv6 is the increase in the number of addresses available for networked devices, allowing, for example, each cell phone and mobile electronic device to have its own address. IPv4 supports 232 (about 4.3 billion) addresses, which is inadequate for giving even one address to every living person, let alone supporting embedded and portable devices. IPv6, however, supports 2128 addresses; this is approximately 5 × 1028 addresses for each of the roughly 6.5 billion people alive today.

Table 71 includes a summary of the key differences between IPv4 and IPv6 protocols:

Table 71: Differences Between IPv4 and IPv6 Protocols

IPv4	IPv6
Source and destination addresses are 32 bits (4 bytes) in length.	Source and destination addresses are 128 bits (16 bytes) in length.
IPSec support is optional.	IPSec support is required.
No identification of packet flow for QoS handling by routers is present within IPv4.	Packet flow identification for QoS handling by routers is present within the IPv6 header using the Flow Label field.
Fragmentation is performed by the sending host, and at the routers, thus slowing performance.	Fragmentation is performed only by the sending host.
No link-layer packet size requirements and has to reassemble 576-byte packet.	Link layer must support 1,280 byte packet and rea sse mb el a 1,500 byte packet.
Header includes a checksum.	Header does not include a checksum.
Header includes options.	All optional data is moved to IPv6 extension headers.
ARP uses Broadcast ARP Request frames to resolve an IPv4 address to a link layer address.	ARP Request frames are replaced with multicast Neighbor Solicitation (Discovery) messages.
IGMP is used to manage local subnet group membership.	IGMP is replaced with Multicast Listener Discovery (MLD) messages.

Table 71: Differences Between IPv4 and IPv6 Protocols (cont.)

IPv4	IPv6
ICMP Router Discovery is used to determine the IPv4 address of the best default gateway and is optional.	ICMPv4 Router Discovery is replaced with ICMPv6 Router Solicitation (Discovery) and Router Advertisement messages and is required.
Broadcast addresses are used to send traffic to all nodes on the subnet.	There are no IPv6 broadcast addresses. Instead a link-local scope all-nodes multicast address is used.
Must be configured either manually or through DHCP for IPv4.	IPV6 does not require manual or DHCP configuration.
Uses host address (A) resource records in DNS to map host names to IPv4 addresses.	Uses AAAA records in the DNS to map host names to IPv6 addresses.
Uses pointer (PTR) resource records in the IN-ADDR.ARPA DNS domain to map IPv4 addresses to host names.	Uses pointer (PTR) resource records in the IP6.INT DNS domains to map IPv6 addresses to host names.

IPv6 Address Format

The IPv6 address is 128 bits long, and is represented as a sequence of eight 16-bit hex values, separated by colons. The preferred format is `xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`.

Example

`FEDC:BA98:7654:BA98:FEDC:1234:ABCD:5412`

Compressing Long Sequences of Zeros

Some addresses can contain long sequences of zeros. A contiguous sequence of zeros can be compressed to :: (double colon).

Example

The address `FE80:0:0:0:2AA:FF:FA:4CA2` can be compressed to `FE80::2AA:FF:FA:4CA2`.

Unlike IPv4, a subnet mask is not used for IPv6 addresses.

Prefix Length for a Network Identifier

IPv6 uses prefix length for network identifier.

Example

In this example, 64 is the network prefix:

`21DA:D300:0000:2F3C::/64`

IPv6 Address Types

There are three types of IPv6 addresses:

- Unicast, page 837
- Multicast, page 837
- Anycast, page 837

Unicast

There are two types of unicast addresses:

- **Global unicast address**—This is an address that can be reached and identified globally. Global unicast addresses use the high-order bit range from 2000 to 3FFF. If the last 64 bits of the address are not configured, Alteon defaults to the EUI-64 (Extended Unique Identifier 64-bit) address format. *RFC 3513* defines the expanding of the Ethernet MAC address based on a 48-bit format into a 64-bit EUI-64 format.

  The interface ID must be unique within the same subnet.
- **Link-local unicast address**—This is an address used to communicate with a neighbor on the same link. Link-local addresses use the high-order bit range from FE80 to FEBF. Link-local unicast addresses are configured on the interface by using the link-local prefix FE80::/10 and the interface identifier in EUI-64 format for its low-order 64-bit. Link-local packets are not routed between subnets.

Multicast

A multicast address (FF00 to FFFF) is an identifier for a group interface. The multicast address most often encountered is a solicited-mode multicast address using prefix FF02::1:FF00:0000/104 with the low-order 24 bits of the unicast or anycast address.

Anycast

Anycast addresses can be global unicast, site-local or link-local addresses used for a one-to-nearest node member of the anycast group communication. Alteon does not support anycast addresses.

Pinging IPv6 Addresses

The following are examples of pinging IPv6 addresses:

To ping an IPv6 address

```
>> Main# /info/l3/nbrcache
>> IP6 Neighbor Discovery Protocol# ping6 3000::1
3000:0:0:0:0:0:0:1 is alive
```

To specify the interface number when pinging to a IPv6 link-local unicast address

```
>> Main# /info/l3/nbrcache
>> IP6 Neighbor Discovery Protocol# ping6 fe80::20d:56ff:fe22:df09
Enter interface number: (1-256) 200
fe80:0:0:0:20d:56ff:fe22:df09 is alive
```

Verifying an IPv6 Configuration

The following are commands used to display and verify an IPv6 configuration.

To verify an IPv6 configuration

- General IPv6 information:

```
>> Main# /info/l3/ip
```

- IPv6 routing table:

```
>> Main# /info/l3/route6
>> IP6 Routing# dump
```

- IPv6 neighbor discovery protocol table:

```
>> Main# /info/l3/nbrcache
>> IP6 Neighbor Discovery Protocol# dump
```

Verifying IPv6 Statistics

The following is the command to display and verify IPv6 statistics.

To display IPv6 statistics

```
>> Main# /stats/l3/ip6
```

Radware Ltd. End User License Agreement

By accepting this End User License Agreement (this “License Agreement”) you agree to be contacted by Radware Ltd.’s (“Radware”) sales personnel.

If you would like to receive license rights different from the rights granted below or if you wish to acquire warranty or support services beyond the scope provided herein (if any), please contact Radware's sales team.

THIS LICENSE AGREEMENT GOVERNS YOUR USE OF ANY SOFTWARE DEVELOPED AND/OR DISTRIBUTED BY RADWARE AND ANY UPGRADES, MODIFIED VERSIONS, UPDATES, ADDITIONS, AND COPIES OF THE SOFTWARE FURNISHED TO YOU DURING THE TERM OF THE LICENSE GRANTED HEREIN (THE “SOFTWARE”). THIS LICENSE AGREEMENT APPLIES REGARDLESS OF WHETHER THE SOFTWARE IS DELIVERED TO YOU AS AN EMBEDDED COMPONENT OF A RADWARE PRODUCT (“PRODUCT”), OR WHETHER IT IS DELIVERED AS A STANDALONE SOFTWARE PRODUCT. FOR THE AVOIDANCE OF DOUBT IT IS HEREBY CLARIFIED THAT THIS LICENSE AGREEMENT APPLIES TO PLUG-INS, CONNECTORS, EXTENSIONS AND SIMILAR SOFTWARE COMPONENTS DEVELOPED BY RADWARE THAT CONNECT OR INTEGRATE A RADWARE PRODUCT WITH THE PRODUCT OF A THIRD PARTY (COLLECTIVELY, “CONNECTORS”) FOR PROVISIONING, DECOMMISSIONING, MANAGING, CONFIGURING OR MONITORING RADWARE PRODUCTS. THE APPLICABILITY OF THIS LICENSE AGREEMENT TO CONNECTORS IS REGARDLESS OF WHETHER SUCH CONNECTORS ARE DISTRIBUTED TO YOU BY RADWARE OR BY A THIRD PARTY PRODUCT VENDOR. IN CASE A CONNECTOR IS DISTRIBUTED TO YOU BY A THIRD PARTY PRODUCT VENDOR PURSUANT TO THE TERMS OF AN AGREEMENT BETWEEN YOU AND THE THIRD PARTY PRODUCT VENDOR, THEN, AS BETWEEN RADWARE AND YOURSELF, TO THE EXTENT THERE IS ANY DISCREPANCY OR INCONSISTENCY BETWEEN THE TERMS OF THIS LICENSE AGREEMENT AND THE TERMS OF THE AGREEMENT BETWEEN YOU AND THE THIRD PARTY PRODUCT VENDOR, THE TERMS OF THIS LICENSE AGREEMENT WILL GOVERN AND PREVAIL. PLEASE READ THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT CAREFULLY BEFORE OPENING THE PACKAGE CONTAINING RADWARE'S PRODUCT, OR BEFORE DOWNLOADING, INSTALLING, COPYING OR OTHERWISE USING RADWARE'S STANDALONE SOFTWARE (AS APPLICABLE). THE SOFTWARE IS LICENSED (NOT SOLD). BY OPENING THE PACKAGE CONTAINING RADWARE'S PRODUCT, OR BY DOWNLOADING, INSTALLING, COPYING OR USING THE SOFTWARE (AS APPLICABLE), YOU CONFIRM THAT YOU HAVE READ AND UNDERSTAND THIS LICENSE AGREEMENT AND YOU AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE AGREEMENT. FURTHERMORE, YOU HEREBY WAIVE ANY CLAIM OR RIGHT THAT YOU MAY HAVE TO ASSERT THAT YOUR ACCEPTANCE AS STATED HEREINABOVE IS NOT THE EQUIVALENT OF, OR DEEMED AS, A VALID SIGNATURE TO THIS LICENSE AGREEMENT. IF YOU ARE NOT WILLING TO BE BOUND BY THE TERMS OF THIS LICENSE AGREEMENT, YOU SHOULD PROMPTLY RETURN THE UNOPENED PRODUCT PACKAGE OR YOU SHOULD NOT DOWNLOAD, INSTALL, COPY OR OTHERWISE USE THE SOFTWARE (AS APPLICABLE). THIS LICENSE AGREEMENT REPRESENTS THE ENTIRE AGREEMENT CONCERNING THE SOFTWARE BETWEEN YOU AND RADWARE, AND SUPERSEDES ANY AND ALL PRIOR PROPOSALS, REPRESENTATIONS, OR UNDERSTANDINGS BETWEEN THE PARTIES. “YOU” MEANS THE NATURAL PERSON OR THE ENTITY THAT IS AGREEING TO BE BOUND BY THIS LICENSE AGREEMENT, THEIR EMPLOYEES AND THIRD PARTY CONTRACTORS. YOU SHALL BE LIABLE FOR ANY FAILURE BY SUCH EMPLOYEES AND THIRD PARTY CONTRACTORS TO COMPLY WITH THE TERMS OF THIS LICENSE AGREEMENT.

1. **License Grant**. Subject to the terms of this Agreement, Radware hereby grants to you, and you accept, a limited, nonexclusive, nontransferable license to install and use the Software in machine-readable, object code form only and solely for your internal business purposes (“Commercial License”). If the Software is distributed to you with a software development kit (the “SDK”), then, solely with regard to the SDK, the Commercial License above also includes a limited, nonexclusive, nontransferable license to install and use the SDK solely on computers within your organization, and solely for your internal development of an integration or interoperation of the Software and/or other Radware Products with software or hardware products owned, licensed and/or controlled by you (the “SDK Purpose”). To the extent an SDK is distributed to you together with code samples in source code format (the “Code Samples”) that are meant to illustrate and teach you how to configure, monitor and/or control the Software and/or any other Radware Products, the Commercial License above further includes a limited,

nonexclusive, nontransferable license to copy and modify the Code Samples and create derivative works based thereon solely for the SDK Purpose and solely on computers within your organization. The SDK shall be considered part of the term "Software" for all purposes of this License Agreement. You agree that you will not assign, sublicense, transfer, pledge, lease, rent or share your rights under this License Agreement nor will you distribute copies of the Software or any parts thereof. Rights not specifically granted herein, are specifically prohibited.

2. **Evaluation Use**. Notwithstanding anything to the contrary in this License Agreement, if the Software is provided to you for evaluation purposes, as indicated in your purchase order or sales receipt, on the website from which you download the Software, as inferred from any time-limited evaluation license keys that you are provided with to activate the Software, or otherwise, then You may use the Software only for internal evaluation purposes ("Evaluation Use") for a maximum of 30 days or such other duration as may specified by Radware in writing at its sole discretion (the "Evaluation Period"). The evaluation copy of the Software contains a feature that will automatically disable it after expiration of the Evaluation Period. You agree not to disable, destroy, or remove this feature of the Software, and any attempt to do so will be a material breach of this License Agreement. During or at the end of the evaluation period, you may contact Radware sales team to purchase a Commercial License to continue using the Software pursuant to the terms of this License Agreement. If you elect not to purchase a Commercial License, you agree to stop using the Software and to delete the evaluation copy received hereunder from all computers under your possession or control at the end of the Evaluation Period. In any event, your continued use of the Software beyond the Evaluation Period (if possible) shall be deemed your acceptance of a Commercial License to the Software pursuant to the terms of this License Agreement, and you agree to pay Radware any amounts due for any applicable license fees at Radware's then-current list prices.
3. **Subscription Software**. If you licensed the Software on a subscription basis, your rights to use the Software are limited to the subscription period. You have the option to extend your subscription. If you extend your subscription, you may continue using the Software until the end of your extended subscription period. If you do not extend your subscription, after the expiration of your subscription, you are legally obligated to discontinue your use of the Software and completely remove the Software from your system.
4. **Feedback**. Any feedback concerning the Software including, without limitation, identifying potential errors and improvements, recommended changes or suggestions ("Feedback"), provided by you to Radware will be owned exclusively by Radware and considered Radware's confidential information. By providing Feedback to Radware, you hereby assign to Radware all of your right, title and interest in any such Feedback, including all intellectual property rights therein. With regard to any rights in such Feedback that cannot, under applicable law, be assigned to Radware, you hereby irrevocably waives such rights in favor of Radware and grants Radware under such rights in the Feedback, a worldwide, perpetual royalty-free, irrevocable, sublicensable and non-exclusive license, to use, reproduce, disclose, sublicense, modify, make, have made, distribute, sell, offer for sale, display, perform, create derivative works of and otherwise exploit the Feedback without restriction. The provisions of this Section 4 will survive the termination or expiration of this Agreement.
5. **Limitations on Use**. You agree that you will not: (a) copy, modify, translate, adapt or create any derivative works based on the Software; or (b) sublicense or transfer the Software, or include the Software or any portion thereof in any product; or (b) reverse assemble, decompile, reverse engineer or otherwise attempt to derive source code (or the underlying ideas, algorithms, structure or organization) from the Software; or (c) remove any copyright notices, identification or any other proprietary notices from the Software (including any notices of Third Party Software (as defined below); or (d) copy the Software onto any public or distributed network or use the Software to operate in or as a time-sharing, outsourcing, service bureau, application service provider, or managed service provider environment. Notwithstanding Section 5(d), if you provide hosting or cloud computing services to your customers, you are entitled to use and include the Software in your IT infrastructure on which you provide your services. It is hereby clarified that the prohibitions on modifying, or creating derivative works based on, any Software provided by Radware, apply whether the Software is provided in a machine or in a human readable form. Human readable Software to which this prohibition applies includes (without limitation) "Radware AppShape++ Script Files" that contain "Special License Terms". It is acknowledged that examples provided in a human readable form may be modified by a user.

6. **Intellectual Property Rights**. You acknowledge and agree that this License Agreement does not convey to you any interest in the Software except for the limited right to use the Software, and that all right, title, and interest in and to the Software, including any and all associated intellectual property rights, are and shall remain with Radware or its third party licensors. You further acknowledge and agree that the Software is a proprietary product of Radware and/or its licensors and is protected under applicable copyright law.

7. **No Warranty**. The Software, and any and all accompanying software, files, libraries, data and materials, are distributed and provided “AS IS” by Radware or by its third party licensors (as applicable) and with no warranty of any kind, whether express or implied, including, without limitation, any non-infringement warranty or warranty of merchantability or fitness for a particular purpose. Neither Radware nor any of its affiliates or licensors warrants, guarantees, or makes any representation regarding the title in the Software, the use of, or the results of the use of the Software. Neither Radware nor any of its affiliates or licensors warrants that the operation of the Software will be uninterrupted or error-free, or that the use of any passwords, license keys and/or encryption features will be effective in preventing the unintentional disclosure of information contained in any file. You acknowledge that good data processing procedure dictates that any program, including the Software, must be thoroughly tested with non-critical data before there is any reliance on it, and you hereby assume the entire risk of all use of the copies of the Software covered by this License. Radware does not make any representation or warranty, nor does Radware assume any responsibility or liability or provide any license or technical maintenance and support for any operating systems, databases, migration tools or any other software component provided by a third party supplier and with which the Software is meant to interoperate.

   This disclaimer of warranty constitutes an essential and material part of this License.

   In the event that, notwithstanding the disclaimer of warranty above, Radware is held liable under any warranty provision, Radware shall be released from all such obligations in the event that the Software shall have been subject to misuse, neglect, accident or improper installation, or if repairs or modifications were made by persons other than by Radware's authorized service personnel.

8. **Limitation of Liability**. Except to the extent expressly prohibited by applicable statutes, in no event shall Radware, or its principals, shareholders, officers, employees, affiliates, licensors, contractors, subsidiaries, or parent organizations (together, the “Radware Parties”), be liable for any direct, indirect, incidental, consequential, special, or punitive damages whatsoever relating to the use of, or the inability to use, the Software, or to your relationship with, Radware or any of the Radware Parties (including, without limitation, loss or disclosure of data or information, and/or loss of profit, revenue, business opportunity or business advantage, and/or business interruption), whether based upon a claim or action of contract, warranty, negligence, strict liability, contribution, indemnity, or any other legal theory or cause of action, even if advised of the possibility of such damages. If any Radware Party is found to be liable to You or to any third-party under any applicable law despite the explicit disclaimers and limitations under these terms, then any liability of such Radware Party, will be limited exclusively to refund of any license or registration or subscription fees paid by you to Radware.

9. **Third Party Software**. The Software includes software portions developed and owned by third parties (the “Third Party Software”). Third Party Software shall be deemed part of the Software for all intents and purposes of this License Agreement; provided, however, that in the event that a Third Party Software is a software for which the source code is made available under an open source software license agreement, then, to the extent there is any discrepancy or inconsistency between the terms of this License Agreement and the terms of any such open source license agreement (including, for example, license rights in the open source license agreement that are broader than the license rights set forth in Section 1 above and/or no limitation in the open source license agreement on the actions set forth in Section 5 above), the terms of any such open source license agreement will govern and prevail. The terms of open source license agreements and copyright notices under which Third Party Software is being licensed to Radware or a link thereto, are included with the Software documentation or in the header or readme files of the Software. Third Party licensors and suppliers retain all right, title and interest in and to the Third Party Software and all copies thereof, including all copyright and other

intellectual property associated therewith. In addition to the use limitations applicable to Third Party Software pursuant to Section 5 above, you agree and undertake not to use the Third Party Software as a general SQL server, as a stand-alone application or with applications other than the Software under this License Agreement.

10. **Term and Termination**. This License Agreement is effective upon the first to occur of your opening the package of the Product, purchasing, downloading, installing, copying or using the Software or any portion thereof, and shall continue until terminated. However, sections 4-13 shall survive any termination of this License Agreement. The License under this License Agreement is not transferable and will terminate upon transfer of the Software. If the Software is licensed on subscription basis, this Agreement will automatically terminate upon the termination of your subscription period if it is not extended.
11. **Export**. The Software or any part thereof may be subject to export or import controls under the laws and regulations of the United States and/or Israel. You agree to comply with such laws and regulations, and, agree not to knowingly export, re-export, import or re-import, or transfer products without first obtaining all required Government authorizations or licenses therefor.
12. **Governing Law**. This License Agreement shall be construed and governed in accordance with the laws of the State of Israel.
13. **Miscellaneous**. If a judicial determination is made that any of the provisions contained in this License Agreement is unreasonable, illegal or otherwise unenforceable, such provision or provisions shall be rendered void or invalid only to the extent that such judicial determination finds such provisions to be unreasonable, illegal or otherwise unenforceable, and the remainder of this License Agreement shall remain operative and in full force and effect. In any event a party breaches or threatens to commit a breach of this License Agreement, the other party will, in addition to any other remedies available to, be entitled to injunction relief. This License Agreement constitutes the entire agreement between the parties hereto and supersedes all prior agreements between the parties hereto with respect to the subject matter hereof. The failure of any party hereto to require the performance of any provisions of this License Agreement shall in no manner affect the right to enforce the same. No waiver by any party hereto of any provisions or of any breach of any provisions of this License Agreement shall be deemed or construed either as a further or continuing waiver of any such provisions or breach waiver or as a waiver of any other provision or breach of any other provision of this License Agreement.

IF YOU DO NOT AGREE WITH THE TERMS OF THIS LICENSE YOU MUST REMOVE THE SOFTWARE FROM ANY DEVICE OWNED BY YOU AND IMMIDIATELY CEASE USING THE SOFTWARE.