Access Operating System (AOS) Configuration Guide

Release 5.0
Part Number 220-0166-01
Order Number TPCG-5.0

Corporate Headquarters
Redback Networks Inc.
250 Holger Way
San Jose, CA 95134-1362
USA
http://www.redback.com
Tel: 408-571-5000

Rights and Restrictions

All statements, specifications, recommendations, and technical information contained are current or planned as of the date of publication of this document. They are reliable as of the time of this writing and are presented without warranty of any kind, expressed or implied. In an effort to continuously improve the product and add features, Redback Networks Inc. ("Redback") reserves the right to change any specifications contained in this document without prior notice of any kind.

Redback shall not be liable for technical or editorial errors or omissions which may occur in this document. Redback shall not be liable for any indirect, special, incidental or consequential damages resulting from the furnishing, performance, or use of this document.

Limited Hardware Warranty and Disclaimer

Limited Warranty. Redback warrants to the original purchaser of the product ("Purchaser") only that the hardware sold hereunder shall be free of defects in material and workmanship and shall perform, under normal use and circumstances, in accordance with Redback's published specifications for a period of ninety (90) days from the shipment date. In the event that Redback receives notice from Purchaser during the warranty period that any hardware does not conform to its warranty, Redback shall, at its sole option (and as Purchaser's sole remedy), either repair or replace the non-conforming hardware, or refund the purchase price of such unit. Hardware replaced under the terms of any such warranty may be refurbished or new equipment substituted at Redback's option. This warranty is the only warranty made by Redback with respect to the hardware delivered hereunder and may be modified, amended or supplemented only by a written instrument signed by a duly authorized officer of Redback and accepted by Purchaser.

Procedures. A hardware item may only be returned with the prior written approval of Redback. Any such approval shall reference a return material authorization number issued by authorized Redback service personnel. Transportation costs, if any, incurred in connection with the return of a defective item to Redback shall be borne by Purchaser. Any transportation costs incurred in connection with the re-delivery of a repaired or replaced item to Purchaser shall be borne by Redback; provided that, such costs shall be borne by Purchaser if Redback reasonably determines that the item is not defective. If Redback determines, in its sole discretion, that the allegedly defective item is not covered by the terms of the warranty provided hereunder or that a warranty claim is made after the warranty period, the cost of repair by Redback, including all shipping expenses, shall be reimbursed by Purchaser.

Exclusions. The foregoing warranties and remedies are for Purchaser's exclusive benefit and are non-transferable. The foregoing warranties do not apply to any hardware which (1) has been altered, except as authorized by Redback, (2) has not been installed, operated, repaired, or maintained in accordance with any installation, handling, maintenance, or operating instructions supplied by Redback, (3) has been subjected to unusual physical or electrical stress, misuse, negligence, or accident (4) is used in ultrahazardous activities, (5) has been damaged or rendered unserviceable by installation or use outside of environmental specifications, or (6) has been exported from the original country of destination. In no event does Redback warrant that Purchaser will be able to operate its networks without problems or interruptions.

Third Party Products. Where a product not manufactured by Redback is sold by Redback hereunder to complete an order, the warranty coverage on that product is limited to its original manufacturer's warranty to the Purchaser, if any.

THE LIMITED WARRANTIES SET FORTH ABOVE ARE IN LIEU OF ALL OTHER WARRANTIES, WHETHER EXPRESSED, IMPLIED, STATUTORY OR OTHERWISE, AND REDBACK SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. REDBACK DOES NOT WARRANT THAT THE PRODUCTS WILL MEET PURCHASER'S REQUIREMENTS OR THAT THE OPERATION OF THE PRODUCTS WILL BE UNINTERRUPTED OR ERROR FREE.

Software License Agreement

CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS. BY INSTALLING AND USING REDBACK SOFTWARE, YOU ARE AGREEING TO BE BOUND BY THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE TO THESE TERMS AND CONDITIONS, DO NOT USE THE REDBACK SOFTWARE PRODUCTS.

Software. The software covered by this license agreement includes all Redback proprietary software, whether provided on magnetic media, loaded on the product or embedded in the product as firmware, and any third party software licensed to Redback included therein (jointly, the "Software").

License. Subject to the terms and conditions of this Agreement, Redback grants to the original end user of the products ("Licensee") a personal, nonexclusive and nontransferable license to use the specific Redback base program, supplement(s) or feature set(s) Software and related product documentation (the "Documentation") for which Licensee has paid the required license fees, in object code form only, in accordance with the terms and conditions of this agreement solely in connection with the use of Redback equipment, on a single hardware chassis, or on a single central processing unit, as applicable, owned or leased by Licensee. If Licensee has purchased a multi-user license, then, subject to the terms and conditions of this Agreement, Licensee is granted a nonexclusive and nontransferable license to allow the number of simultaneous users authorized under such license and for which Licensee has paid the required license fee to use the Software.

Copies. Licensee agrees not to make any copies of the Software or the Documentation, in whole or in part, other than one copy for archival purposes only. Licensee agrees not to modify, translate, reverse engineer, de-compile, disassemble, or create derivative works based on the Software, except to the extent that the such limitation is prohibited by applicable law. Licensee agrees to take reasonable steps to safeguard copies of the Software against disclosure, copying or use by unauthorized persons, and to take reasonable steps to ensure that the provisions of this license are not violated by Licensee's employees or agents.

Proprietary Information. Licensee agrees that aspects of the Software and Documentation constitute trade secrets and/or copyrighted material of Redback or its suppliers. Licensee shall not disclose, provide, or otherwise make available such trade secrets or copyrighted material to any third party without the written consent of Redback.

Title. All right, title and interest in and to the Software and Documentation, including all intellectual property rights therein, shall remain the property of Redback or its suppliers, subject only to the limited license granted to Licensee. This license is not a sale and does not transfer to Licensee any title or ownership in or to the Software or the Documentation or any patent, copyright, trade secret, trade name, trademark or other proprietary or intellectual property rights related thereto.

Limited Warranty. Redback warrants to Licensee only that the media on which the Software is recorded shall be free from defects in materials and workmanship under normal use for a period of 90 days from the date of shipment by Redback. Licensee's exclusive remedy, and Redback's exclusive liability, shall be replacement of the media in accordance with this limited warranty. THE SOFTWARE IS PROVIDED "AS IS." REDBACK EXPRESSLY DISCLAIMS AND NEGATES ALL WARRANTIES FOR THE SOFTWARE, WHETHER EXPRESSED, IMPLIED, STATUTORY OR OTHERWISE, AND REDBACK SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT OF INTELLECTUAL PROPERTY OR OTHER VIOLATION OF RIGHTS. Redback does not warrant that the Software will meet Licensee's requirements or that the operation of the Software will be uninterrupted or error free. This warranty gives Licensee specific legal rights. Licensee may also have other rights, which vary from state to state or country to country.

Limitation of Liability. IN NO EVENT WILL REDBACK OR ITS SUPPLIERS BE LIABLE FOR ANY LOST REVENUE, PROFIT, OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL, OR PUNITIVE DAMAGES HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE EVEN IF REDBACK OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In no event shall Redback or its suppliers' liability to Licensee, whether in contract, tort (including negligence), or otherwise, exceed the license fees paid by Licensee. Some states or countries do not allow exclusion or limitation of incidental or consequential damages or limitation on how long an implied warranty lasts, so the above limitations or exclusions may not apply to Licensee.

Termination. This agreement shall continue in effect until terminated hereunder. This agreement shall terminate automatically on Licensee's failure to comply with any of the provisions herein, including any attempt to transfer this license or the Software or Documentation. Upon any termination, Licensee shall promptly destroy or return to Redback all copies of the Software and Documentation, including all original and archival copies. No refunds shall be given for such returned materials. Notwithstanding any termination of this License, the rights and obligations relating to title, warranty, termination and limitation of liability, as well as any other provisions which survive by their terms, shall survive termination:

Restricted Rights. The Software and Documentation are provided with Restricted Rights. Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c) (1) and (2) of the Commercial Computer Software—Restricted Rights at 48 CFR 52.227-19, as applicable. Manufacturer is Redback Networks Inc., 1195 Borregas Avenue, Sunnyvale, California 94089.

Miscellaneous. Licensee may not assign or transfer any of its rights or delegate any of its obligations under this agreement. No delay, failure or waiver by either party to exercise any right or remedy under this agreement shall operate to waive any exercise of such right or remedy or any other right or remedy. This agreement shall be governed by and construed in accordance with the laws of the State of California without regard to conflict of laws principles and without regard to the 1980 U.N. Convention on Contracts for the International Sale of Goods. If any provision in this agreement shall be found or be held to be invalid or unenforceable, then the meaning of said provision shall be construed, to the extent feasible, so as to render the provision enforceable, and the remainder of this agreement shall remain in full force and effect. This agreement constitutes the entire agreement between Licensee and Redback with respect to the subject matter of this agreement.

Proprietary Notices. Licensee shall maintain and reproduce all copyright and other proprietary notices on all copies of the Software in the same form and manner that such notices are included on the Software. The following third party Software may be included with your product and is subject to this software license agreement. All rights in copyright are reserved to the copyright owner:

SNMP Monolithic Agent. © 1992–1998 SNMP Research International, Inc.

VxWorks. © 1984–1998 Wind River Systems, Inc.

Redback adaptation and implementation of the UDP and TCP protocols developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. © 1982, 1986, 1988, 1990, 1993, 1995 The Regents of the University of California. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the University of California, Berkeley and its contributors."

Point-to-Point Protocol (PPP). © 1989 Carnegie-Mellon University.

Dynamic Host Configuration Protocol (DHCP) © 1997, 1998 The Internet Software Consortium.

Portions of the Redback Access Operating System (AOS) use cryptographic software written by Eric Young (eay@cryptsoft.com).

SSH IPSEC Technology (pat.pending). © 1995–2000 SSH Communications Security Ltd. (www.ssh.fi).

Neither the name of any third party Software developer nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission of such third party.

Limitation of Liability and Damages

THE FOLLOWING LIMITATION OF LIABILITY AND DAMAGES APPLIES TO ALL HARDWARE, SOFTWARE AND DOCUMENTATION SOLD, LICENSED OR OTHERWISE DISTRIBUTED BY REDBACK OR ITS RESELLERS.

IN NO EVENT SHALL REDBACK, ITS SUPPLIERS OR ITS DISTRIBUTORS BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGE, INCLUDING WITHOUT LIMITATION LOSS OF DATA, LOST PROFITS OR COST OF COVER, ARISING FROM THE USE OF THE HARDWARE, SOFTWARE OR DOCUMENTATION OR ANY DEFECT IN THE HARDWARE, SOFTWARE OR DOCUMENTATION, HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY. THIS LIMITATION SHALL APPLY EVEN IF REDBACK, ITS SUPPLIERS OR ITS DISTRIBUTOR SHALL HAVE BEEN ADVISED OF THE POSSIBILITY OF ANY SUCH DAMAGE. IN PARTICULAR, BUT WITHOUT LIMITATION, REDBACK, ITS SUPPLIERS AND ITS DISTRIBUTORS SHALL HAVE NO LIABILITY FOR THE LOSS OF ANY INFORMATION STORED OR COMMUNICATED OR ATTEMPTED TO BE STORED OR COMMUNICATED WITHIN ANY REDBACK SYSTEM USING THE HARDWARE OR SOFTWARE.

THE MAXIMUM AGGREGATE LIABILITY OF REDBACK AND ITS SUPPLIERS FOR ANY CLAIM ARISING OUT OF USE OF THE HARDWARE, SOFTWARE OR DOCUMENTATION OR ANY DEFECT IN THE HARDWARE, SOFTWARE OR DOCUMENTATION, ON ANY AND ALL THEORIES OF LIABILITY, INCLUDING WITHOUT LIMITATION NEGLIGENCE BY REDBACK, SHALL IN ALL EVENTS BE LIMITED TO RETURN OF THE AMOUNTS ACTUALLY PAID TO REDBACK FOR THE DEFECTIVE HARDWARE OR SOFTWARE, LESS DEPRECIATION OF SUCH AMOUNTS LINEARLY OVER A THREE-YEAR PERIOD, WHICH THE PARTIES AGREE CONSTITUTES A REASONABLE RATE OF DEPRECIATION.

FCC Notice

The following information is for FCC compliance of Class A devices: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC rules. 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 users will be required to correct the interference at their own expense.

1. MODIFICATIONS

The FCC requires the user to be notified that any changes or modifications made to this device that are not expressly approved by Redback could void the user's authority to operate the equipment.

2. CABLES

Connection to this device must be made with shielded cables with metallic RFI/EMI connector hoods to maintain compliance with FCC Rules and Regulations.

3. POWER CORD SET REQUIREMENTS

The power cord set used with the System must meet the requirements of the country, whether it is 100-120 or 220-264 VAC. For the U.S. and Canada, the cord set must be UL Listed and CSA Certified and suitable for the input current of the system.

For DC-powered systems, the installation instructions need to be followed.

VCCI Class A Statement

この装置は、情報処理装置等電波障害自主規制協議会（ＶＣＣＩ）の基準に基づくクラス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。

Safety Notices

1. Laser Equipment:

CAUTION! USE OF CONTROLS OR ADJUSTMENTS OF PERFORMANCE OR PROCEDURES OTHER THAN THOSE SPECIFIED HEREIN MAY RESULT IN HAZARDOUS RADIATION EXPOSURE.

Class 1 Laser Product
Product is certified by the manufacturer to comply with DHHS Rule 21 Subchapter J.

CAUTION! INVISIBLE LASER RADIATION WHEN AN OPTICAL INTERFACE IS OPEN.

2. Lithium Battery Warnings:

It is recommended that, when required, Redback replace the lithium battery.

WARNING! Do not mutilate, puncture, or dispose of batteries in fire. The batteries can burst or explode, releasing hazardous chemicals. Discard used batteries according to the manufacturer's instructions and in accordance with your local regulations.

Danger of explosion if battery is incorrectly replaced. Replace only with the same or equivalent type as recommended by the manufacturer's instructions.

VARNING Eksplosionsfara vid felaktigt batteribyte. Använd samma batterityp eller en ekvivalent typ som rekommenderas av apparattillverkaren. Kassera använt batteri enligt fabrikantens instruktion.

ADVARSEL! Lithiumbatteri—Eksplosionsfare ved fejlagtig håndtering. Udskiftning må kun ske med batteri af samme fabrikat og type. Levér det brugte batteri tilbage tilleverandøren.

VARIOTUS Paristo voi räjähtää, jos se on virheellisesti asennettu. Vaihda paristo ainoastaan valmistajan suosittelemaan tyyppiin. Hävitä käytetty paristo valmistajan ohjeiden mikaisesti.

ADVARSEL Eksplosjonsfare ved feilaktig skifte av batteri. Benytt samme batteritype eller en tilsvarende type anbefait av apparatfabrikanten. Brukte batterier kasseres i henhold til fabrikantens instruksjoner.

WAARSCHUWING! Bij dit produkt zijn batterijen geleverd. Wanneer deze leeg zijn, moet u ze niet weggooien maar inleveren als KCA.

Table of Contents

Part 6: Tunnels

Part 7: Security

Part 8: IP Services

Part 10: Access Control Lists

Part 14: Appendixes

Part 15: Indexes

About This Guide

Objectives

This guide describes the user tasks required to configure, operate, and maintain all Subscriber Management System (SMS) products. A technical overview, a set of configuration procedures, and configuration examples are provided for each Redback Access Operating System (AOS) feature set.

Note Each SMS platform supports a distinct set of modules and ports. Some configuration tasks described in this guide are specific to a particular Redback platform. For example, the SMS 10000 contains Connection Manager (CM) and System Manager (SM) modules, where the SMS 500, SMS 1000, and SMS 1800 have Forwarding Engine (FE) and Control Engine (CE) modules. This guide describes user tasks for all Redback products, including tasks specific to a particular platform, such as displaying FE module information using the **show fe stats** command, and displaying CM module information using the **show cm stats** command.

Related Publications

Use this guide in conjunction with the *Access Operating System (AOS) Command Reference* publication, which provides the syntax description and usage guidelines for all Redback AOS commands.

Intended Audience

This publication is intended for system and network administrators experienced in access and internetwork administration.

Organization

This guide is organized as follows:

- Part 1. Getting Started

  Describes SMS functions and applications and provides an overview of the command modes that comprise the Redback user interface. Provides information on how to access the Redback AOS, load system images and configuration files, and configure basic system parameters.

- Part 2. Setting Up Contexts with Interfaces and Subscribers

  Explains how to configure the local context and multiple contexts, and how to set up accounts for the operators and administrators who will maintain one or more contexts. Provides information on how to configure interfaces and subscribers, which are associated with a specific context.

- Part 3. Ports, Circuits, Channels, and Bindings

  Describes the tasks needed to configure common port, circuit, channel elements, and port-specific information. Provides encapsulation information and describes Asynchronous Transfer Mode (ATM) and Frame Relay protocol-specific tasks, including the creation of profiles, which can be easily applied to multiple ports. Describes how to bind circuits, channels, or ports to interfaces, subscribers, Layer 2 Tunneling Protocol (L2TP) or L2F (Layer 2 Forwarding) peers, L2TP groups, or bypasses.

- Part 4. Bridges and Bypasses

  Describes how to configure media access control (MAC)-based (transparent) and IEEE 802.1D Spanning-Tree Protocol bridges. Explains how to configure bypasses, which bind two circuits, channels, or ports together without protocol translation.

- Part 5. Point-to-Point Protocol

  Provides information on how to configure Point-to-Point Protocol (PPP) and PPP over Ethernet (PPPoE).

- Part 6. Tunnels

  Provides the tasks needed to configure GRE, L2TP, or L2F tunnels and peers.

- Part 7. Security

  Describes how to use the AOS to implement IP Security (IPSec) on an SMS device. You must have an IPSec/Compression Transform Engine (TE) module installed in your SMS device to implement IPSec.

- Part 8. IP Services

  Describes how the AOS interacts with Dynamic Host Configuration Protocol (DHCP) servers, Domain Naming System (DNS) servers, and Network Time Protocol (NTP) servers.

- Part 9. Routing

  Explains how to configure static IP, Routing Information Protocol (RIP), Open Shortest Path First (OSPF), and Border Gateway Protocol (BGP) routing. Describes how to configure routing policies. Provides the tasks needed to configure Internet Group Management Protocol (IGMP) proxy, which allows the SMS device to forward IP multicast traffic without running a multicast routing protocol.

- Part 10. Access Control Lists

  Provides information on how to configure IP and bridging access control lists, and service access lists, which restrict subscriber access to contexts and domains on a per-circuit basis.

- Part 11. AAA and RADIUS

  Describes the tasks needed to configure global or context-specific Authentication, Authorization, and Accounting (AAA) and Remote Authentication Dial-In User Service (RADIUS) features. Subscriber AAA can be accomplished through local configuration of subscriber records or through a remote RADIUS server. Circuits can be created on-demand via RADIUS for a specific context. The AAA configuration of the specified context is used to configure the profile, encapsulation, and binding of each circuit.

- Part 12. System Management

  Explains how to monitor and test system-wide parameters, including administrators, hardware, memory, processes, and so on. Provides the configuration tasks that enable the AOS to collect system statistics (bulkstats), and to transfer and store the collected data on remote servers. Describes how to configure system event logging.

- Part 13. Network Management

  Explains how to configure Simple Network Management Protocol (SNMP) and Remote Monitoring (RMON) features, how to access the AOS from a web browser, and how to configure the Netop server port on the SMS device that is used to communicate with the NetOp Network Manager product.

- Part 14. Appendixes

  Provide a sample Redback AOS system configuration file, a list of supported Management Information Base (MIB) objects, RADIUS attributes, and L2TP attribute pair values.

- Part 15. Index

  Provides an alphabetical index.

Conventions

Commands

Command descriptions use the following conventions:

- Commands and keywords are indicated in **boldface**.
- Arguments where you must supply the value are indicated in *italics*.
- Optional constructs within commands, and constructs that are not used in all cases are shown in square brackets ([ ]).
- Alternative parameters within commands are separated by vertical bars (|).
- Alternative, but required parameters, are shown within grouped braces ({ }), and are separated by vertical bars (|).

Examples

Examples use the following conventions:

- System prompts are of the form `[context]hostname(mode)#` for interactive sessions. Here the context is the current context in which operator and administrator commands are applied, the hostname is the configured name of the Redback system, and the mode is a string indicating the current configuration mode, if applicable. For example, the prompt in context configuration mode is `[local]RedBack(config-ctx)#`.
- Information displayed by the system is in `Courier` font.
- Information that you should enter is in **`boldface Courier`** font.

Part 1

Getting Started

System Overview

This chapter describes the Access Operating System (AOS) software, including where our products fit in today's high-speed access networks, an overview of AOS concepts and terminology, and an overview of the AOS configuration process.

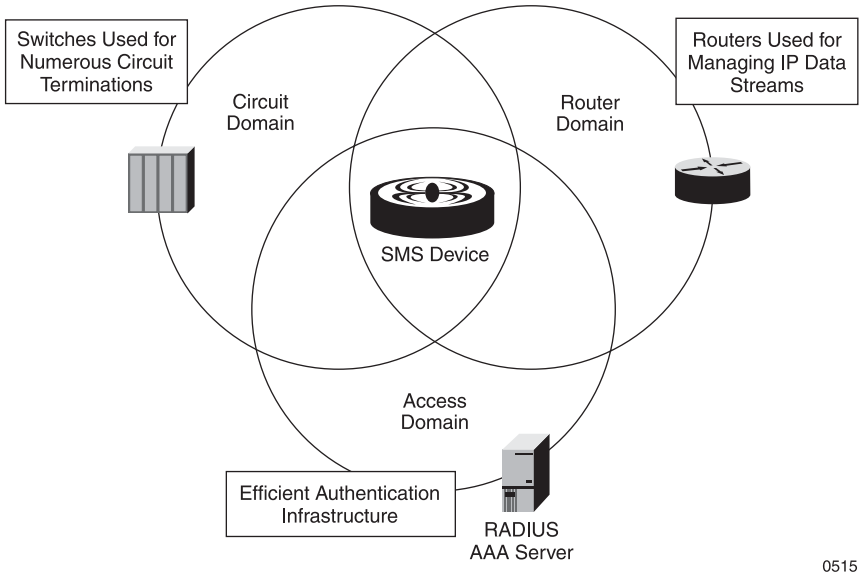
The Redback Solution

Modern networks have grown to the point where general-purpose devices can no longer handle all the complex functions necessary to deliver emerging high-speed services. Increasingly, service providers have partitioned their networks into access functions and backbone functions. Our products, including the Subscriber Management System (SMS) hardware and the AOS software, bridge the gap between high-speed access methods, such as digital subscriber line (DSL), cable, and wireless, and the Internet backbone.

AOS Functions

The AOS software provides effective circuit termination, access functions, and routing in a single system. Figure 1-1 shows the functional areas of the our products.

Figure 1-1 AOS Functions



The AOS software provides the following functions:

- Circuit termination—The AOS provides effective circuit termination for all major high-speed access methods. Whether deployed by carriers, cable operators, or service providers, the AOS accepts a large concentration of high-speed data traffic from such devices as DSL access multiplexors (DSLAMs), cable modem termination systems, and wireless termination systems. The AOS offloads the circuit termination function from routers connecting to the Internet backbone, reducing the processing requirements for these routers and providing a scalable solution. The AOS supports up to 10,000 subscribers.
- Access—The AOS provides access functionality that traditional routers were not designed to provide, such as subscriber management, provisioning, authentication, and accounting. AOS supports service providers' existing accounting and management software systems, enabling service providers to quickly deploy new high-speed access services.
- Routing—The AOS provides routing of subscriber traffic based on layer 3 addressing. The AOS performs all translations necessary to convert subscriber traffic to IP, relieving the service provider backbone routers of frame translations that can cause congestion on high-volume routers. The AOS grooms individual subscriber data streams into simplified IP flows for routers connecting to the Internet backbone.

AOS Features

The AOS is an advanced operating system designed to optimize subscriber management and routing functions. Some of the key features that the AOS software supports include:

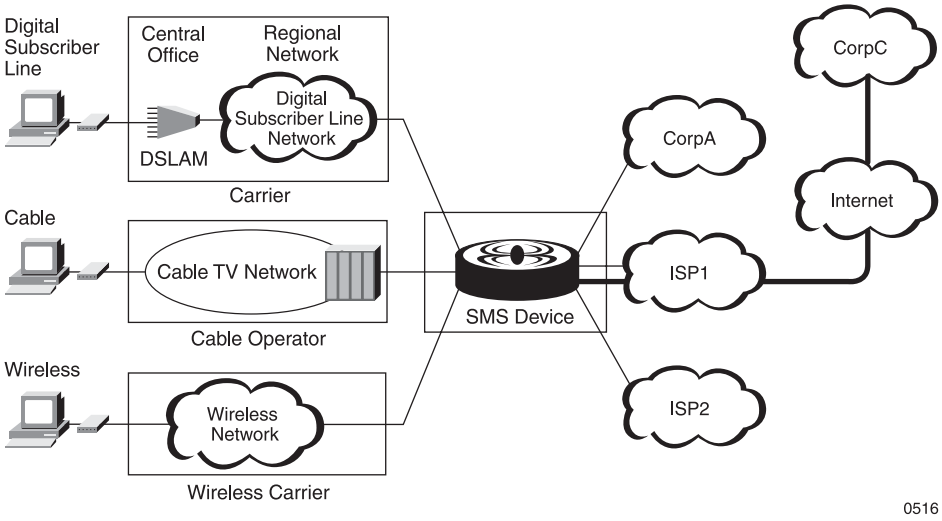
- Dynamic service selection—The AOS supports the unique capability to dynamically bind subscriber sessions to services. This capability enables dynamic service selection to be deployed by carriers and service providers alike.

- Layer 2 Tunneling Protocol (L2TP)—The AOS supports L2TP, the standard method of building a Virtual Private Network (VPN) that enables fixed and mobile users to simulate a private network using a shared infrastructure, such as the Internet. VPNs also enable mobile users to make secure connections to their corporate intranets or extranets over the public Internet.
- Traffic management—The AOS supports traffic management features, including policing and rate-limiting, to support the creation of different service classes and provide service providers with predictable traffic behavior for better management of their networks.
- Routing protocol support—The AOS includes support for various popular routing protocols.
- IP multicast—The AOS supports Internet Group Multicast Protocol (IGMP) proxy functionality.
- Web-based management—The web-based management capabilities in the AOS allow service providers to streamline operations and simplify troubleshooting through a common, easy-to-use browser interface.
- Bulk statistics—The bulk statistics capabilities in the AOS allow service providers access to information that enables them to provide efficient storage and transfer of high-volume accounting data.

AOS Applications

Figure 1-2 shows how the Subscriber Management System (SMS) device provides access services for different types of high-speed access methods, including DSL, cable, and wireless. It also shows the SMS device being used to provide access to multiple networks, including two corporate networks (`CorpA` and `CorpC`), and two service provider networks (`ISP1` and `ISP2`). In this example, `ISP1` provides a VPN service for `CorpC` using a tunnel. All of these features can be implemented using a single system.

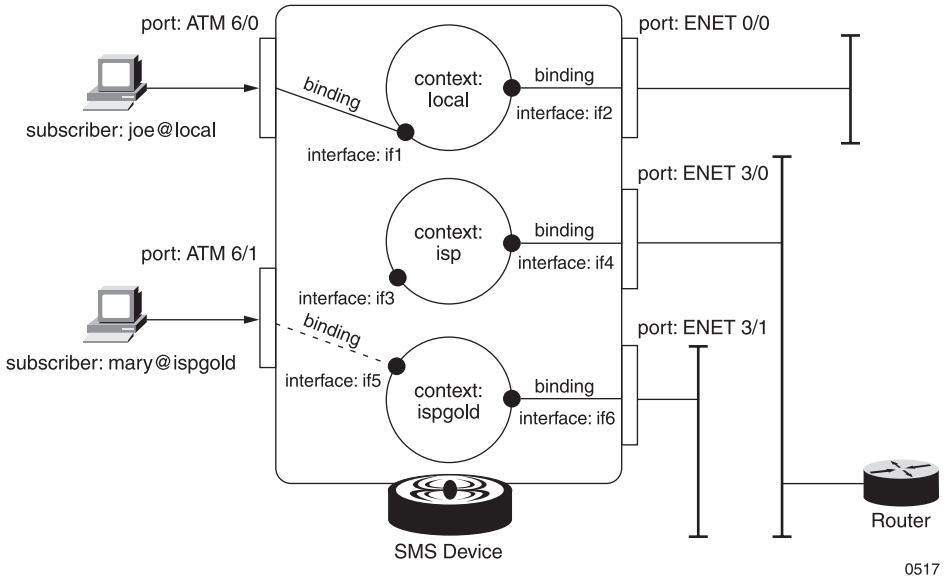
Figure 1-2 AOS Application



AOS Concepts

Figure 1-3 shows the relationships between different AOS software components. The rest of this section describes these components in greater detail.

Figure 1-3 AOS Software Component Interrelationships



Contexts

Most networking products are designed so that the entire set of ports, circuits, and protocols operate together as one global instance. The AOS supports an advanced feature called multiple contexts. Each AOS context is a virtual SMS device instance running within a single physical device. A context has its own IP routing table, its own Remote Authentication Dial-In User Service (RADIUS) client, and so on, and does not share this information with other contexts. By separating the address and name spaces in this way, service providers can use multiple contexts to manage subscribers and provide access for customers of different providers, or to provide different classes of services for customers. Service providers use a single physical device to implement this, with one or more contexts being assigned to each service provider or service class.

Figure 1-3 shows three contexts configured within a single SMS device: `local`, `isp`, and `ispgold`. The `isp` and `ispgold` contexts show the use of multiple contexts to provide different classes of service. For example, the `isp` context could provide a standard level of service, while the `ispgold` context could support a premium level of service, such as higher-speed access. Each of the three contexts functions independently of the others.

An SMS device with a single context configured is similar to traditional networking products. This is referred to as a single-context configuration.

Every configuration includes a special context named local that cannot be deleted. In single-context configurations, this is the only context. The local context allows you to do the following:

- Configure and examine other contexts.
- Configure global resources such as ports, the Simple Network Management Protocol (SNMP), and system logging.

Each context provides a separate security, management, and operating environment on behalf of a given network. You configure interface and subscriber information as part of a context.

Interfaces

The concept of an interface in the AOS differs from that in traditional networking devices. The term, interface, is often used synonymously with port or circuit, which are physical entities. In the AOS, an interface is a logical construct that provides higher-layer protocol and service information, such as layer 3 addressing. Interfaces are configured as part of a context, and are independent of physical ports and circuits. The decoupling of the interface from the physical layer entities enables many of the advanced features offered by the AOS.

For the higher-layer protocols to become active, you must associate an interface with a physical port or circuit. This association is referred to as a binding in the AOS. See the “Bindings” section later in this chapter for more information.

In Figure 1-3, each context shows two interfaces. These interfaces are configured within each context. The bindings shown in the figure are not present when the interfaces are configured; they are configured later.

Subscribers

Subscribers are the end users of high-speed access services. You configure subscriber records as part of a context, either locally on the SMS device or on a RADIUS server. Subscriber records contain the information necessary to bind a subscriber to the correct interface, and therefore, to the correct network context and services. Subscriber records can also contain other configuration information, such as authentication, access control, rate-limiting, and policing information.

Ports and Circuits

Ports and circuits in the AOS represent the physical connectors and channels on the SMS hardware I/O modules. Physical port and circuit configuration includes traffic profiles and data encapsulation information. Traffic profiles provide a configuration shortcut. A single traffic profile with traffic shaping, counter, and statistics configuration information can be applied to multiple ports. All circuits must have a configured encapsulation.

For configuration purposes, Ethernet ports are treated as a single circuit. This means that many of the generic circuit configuration commands are also available in Ethernet port configuration mode.

Before any higher-layer user data can flow through a physical port or circuit, you must associate that port or circuit with an interface, a bypass, or a tunnel within a context. This association is referred to as a binding in the AOS. The configuration for each port and circuit includes binding information.

Bindings

Bindings form the association in the AOS between the circuits or tunnels and the higher-layer bridging, routing, and switching protocols configured for a given context. No user data can flow on a circuit or Ethernet port until some higher-layer service is configured and associated with it. Bindings are either statically mapped during configuration or dynamically created based on subscriber characteristics as defined in the local database, or on a RADIUS server. Once bound, traffic flows through the context as it would through any IP router.

Static binding occurs when you bind a circuit directly to an interface. In this case, the circuit is hard-wired to the higher-layer protocols defined for the interface. This is the simplest form of binding available in the AOS, providing functionality similar to that provided by traditional network devices, such as routers. You can use static bindings for any circuit with any encapsulation type. The bindings between the Ethernet ports and the interfaces within the contexts shown in Figure 1-3 are static bindings.

You can also statically bind a circuit to a particular subscriber in a given context. In this case, the binding between the circuit and the higher-layer protocols is determined indirectly, through the subscriber record. In Figure 1-3, subscriber `joe` is configured with an IP address that maps to the `if1` interface in the `local` context. When the virtual circuit on Asynchronous Transfer Mode (ATM) port `6/0` is bound to the subscriber named `joe`, the AOS determines the interface that the circuit will be bound to by looking at the subscriber information for `joe`.

Dynamic binding occurs when you bind a circuit to the higher-layer protocols based on session information. For example, a Point-to-Point Protocol (PPP)-encapsulated session could be bound to a particular context and interface by examining the authenticated structured username in the form *user@context*.

Dynamic binding is the key to enabling advanced features, such as dynamic service and provider selection. Dynamic binding also enables simultaneous access to multiple services on a single circuit.

Figure 1-3 shows a dynamic binding between the virtual channel on ATM port `6/1` and the `if5` interface in the `ispgold` context. When the subscriber initiates a PPP session using the structured username `mary@ispgold, the` AOS determines the context (`ispgold`) for the connection, and selects an interface (`if5`) to bind the circuit to. Successful dynamic binding depends on subscriber information for mary configured in context `ispgold`, and successful PPP authentication during PPP session establishment. The binding between this circuit and the `ispgold` context is removed when the PPP session is ended. Because the binding on the circuit is dynamic, this same circuit could be used by a different subscriber to select a different service.

Understanding the User Interface

The primary user interface to the Access Operating System (AOS) is the command-line interface (CLI). You can access the CLI from the console port or through a remote session to perform all configuration tasks and to monitor the AOS. All CLI commands are simple strings of keywords and user-specified arguments.

This chapter provides an overview of the user interface and the basic features that allow you to navigate the CLI effectively. The following topics are covered:

- Command Modes
- Auto-More
- Command-Line History and Command Editing

For detailed information on syntax and usage guidelines for commands listed under "Command Modes," see the "User Interface Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

Command Modes

The CLI is comprised of modes. There are two groups of modes: exec and configuration. Within the exec mode grouping are two modes: operator exec and administrator exec. The configuration mode group contains all the remaining modes (see Figure 2-1).

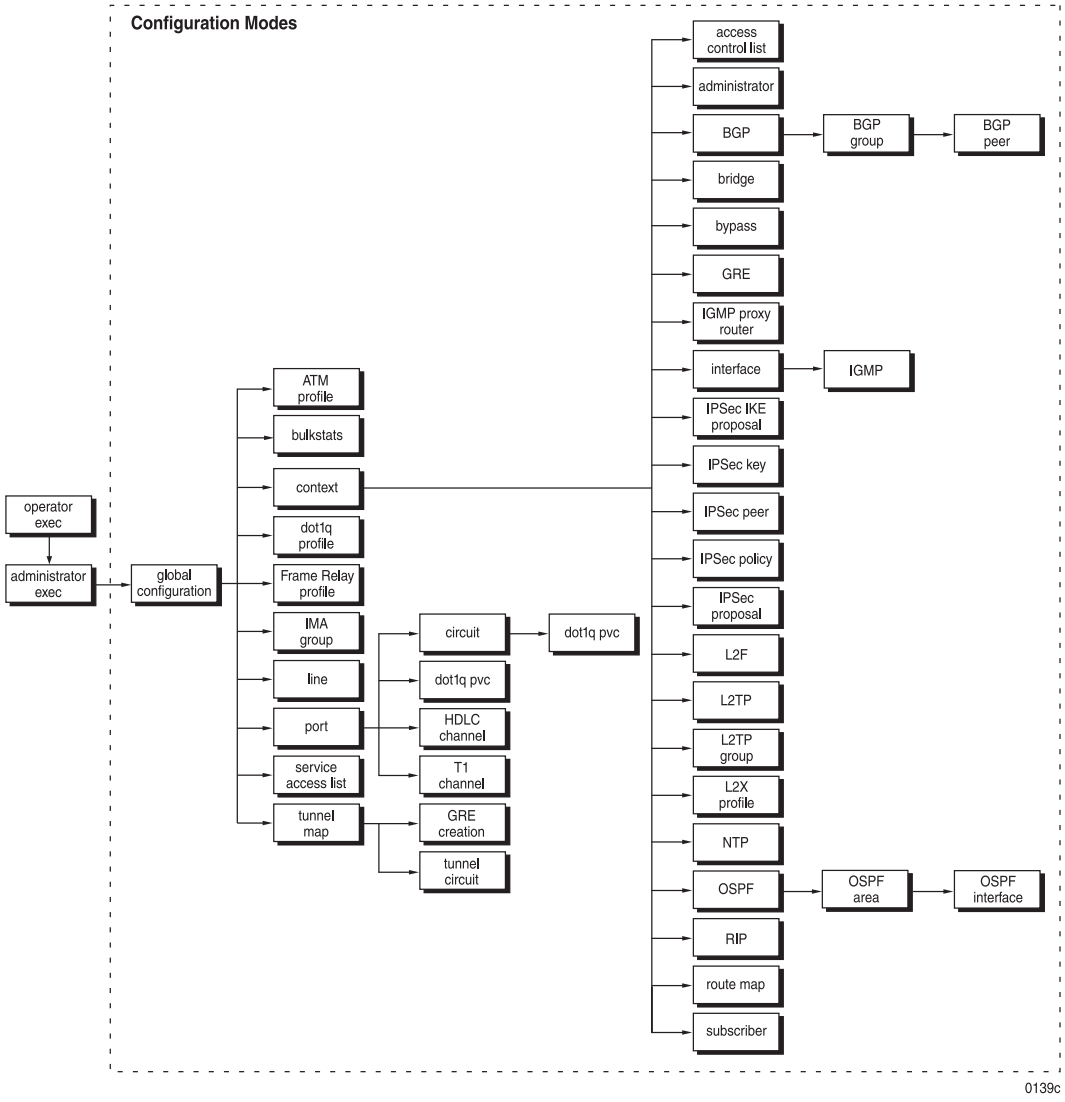
A mode is an environment in which a group of related commands is valid. All commands are mode-specific, and certain commands are valid in more than one mode. When initiating a session, the CLI is always set to the operator exec (nonprivileged exec) mode by default. This mode allows the user to examine the state of the system through a subset of the available CLI commands. To make any changes to the system, you must enter the administrator exec (privileged exec) mode. The **enable** command is used for this purpose. The **enable** command allows an operator or administrator to change the privilege level for the exec session, up to the maximum value configured for the account (see the "Configure Privilege Levels" section in Chapter 5, "Configuring Basic System Parameters").

Within any configuration mode, the **exit** command brings the user up one level in the mode hierarchy (shown in Figure 2-1). For example, when in subscriber configuration mode, typing **exit** takes you to context configuration mode. The **exit** command ends a CLI session when typed from the operator exec or administrator exec mode. The **end** command causes the CLI to immediately exit any configuration mode and return to the administrator exec mode.

Note Within any configuration mode, you can type commands at the next highest level of the hierarchy without first entering the **exit** command. For example, within the interface configuration mode, you can type any of the commands in that mode and any commands in the context configuration mode—the next highest mode in the hierarchy. This is a keyboard shortcut and its use in interactive sessions (console and remote), and in configuration files, is optional.

The *Access Operating System (AOS) Command Reference* publication describes the currently supported commands. Figure 2-1 shows the relationships of the various CLI modes to one another.

Figure 2-1 Relationship of CLI Modes



System prompts are of the form `[context]hostname(mode)#` for interactive sessions. In this example, *context* is the current context to which operator and administrator commands are applied, *hostname* is the currently configured hostname of the device, and *mode* is a string indicating the current configuration

mode, if applicable. For example, the prompt in global configuration mode, assuming the factory default *hostname* and *context*, is `[local]RedBack(config)#`. Table 2-1 shows the CLI commands used to enter each mode and the system prompt that is displayed when you are in each mode.

Table 2-1 Mode Access Commands and Prompts

Mode Name	Commands Used to Access	Command-Line Prompt
operator exec	(user logon)	>
administrator exec	**enable** command in operator exec mode	#
global configuration	**configure** command in administrator exec mode	(config)#
access control list configuration	**ip access-list** or **bridge access-list** command in context configuration mode	(config-acl)#
administrator configuration	**administrator** command in context configuration mode	(config-admin)#
ATM profile configuration	**atm profile** command in global configuration mode	(config-atmpro)#
BGP configuration	**router bgp** command in context configuration mode	(config-bgp)#
BGP group configuration	**group** command in BGP configuration mode	(config-group)#
BGP peer configuration	**neighbor** command in BGP group configuration mode	(config-peer)#
bridge configuration	**bridge** command in context configuration mode	(config-bridge)#
bulkstats configuration	**bulkstats mode** command in global configuration mode	(config-bulkstats)#
bypass configuration	**bypass** command in context configuration mode	(config-bypass)#
circuit configuration	**atm pvc** or **frame-relay pvc** command in port configuration mode or **frame-relay pvc** command in HDLC channel configuration mode	(config-pvc)#
context configuration	**context** command in global configuration mode	(config-ctx)#
dot1q profile configuration	**dot1q profile** command in global configuration mode	(config-dot1qpro)#
dot1q PVC configuration	**dotq1 pvc** command in port or circuit configuration mode	(config-dot1-pvc)#
Frame Relay profile configuration	**frame-relay profile** command in global configuration mode	(config-frpro)#
GRE configuration	**gre-peer** command in context configuration mode	(config-gre)#
GRE creation configuration	**gre-circuit** command in tunnel-map configuration mode	(config-gre-creation)#
HDLC channel configuration	**hdlc-channel** command in port configuration mode	(config-chan)#

Table 2-1 Mode Access Commands and Prompts

Mode Name	Commands Used to Access	Command-Line Prompt
IGMP configuration	**ip igmp mode** command in interface configuration mode	(config-igmp)#
IGMP proxy router configuration	**router igmp-proxy** command in context configuration mode	(config-router-igmp)#
IMA group configuration	**ima group** command in global configuration mode	(config-ima)#
interface configuration	**interface** command in context configuration mode	(config-if)#
IPSec IKE proposal configuration	**ipsec proposal ike name** command in context configuration mode	(config-ipsec-proposal_ike)#
IPSec key configuration	**ipsec key name** command in context configuration mode	(config-ipsec-key)#
IPSec peer configuration	**ipsec peer name** command in context configuration mode	(config-ipsec-peer)#
IPSec policy configuration	**ipsec policy name** command in context configuration mode	(config-ipsec-policy)#
IPSec proposal configuration	**ipsec proposal crypto name** command in context configuration mode	(config-ipsec-proposal)#
L2F configuration	**l2f-peer name** command in context configuration mode	(config-l2f)#
L2TP configuration	**l2tp-peer default**, **l2tp-peer name**, or **l2tp-peer unnamed** command in context configuration mode	(config-l2tp)#
L2TP group configuration	**l2tp-group name** command in context configuration mode	(config-l2tpgrp)#
L2X profile configuration	**l2x profile** command in context configuration mode	(config-l2xprof)#
line configuration	**line** command in global configuration mode	(config-line)#
NTP configuration	**ntp mode** command in context configuration mode	(config-ntp)#
OSPF area configuration	**area** command in OSPF configuration mode	(config-ospf-area)#
OSPF configuration	**router ospf** command in context configuration mode	(config-ospf)#
OSPF interface configuration	**ospf-interface** command in OSPF area configuration mode	(config-ospf-interface)#
port configuration	**port** command in global configuration mode	(config-port)#
RIP configuration	**router rip** command in context configuration mode	(config-rip)#
route map configuration	**route-map** command in context configuration mode	(config-route-map)#

Table 2-1 Mode Access Commands and Prompts

Mode Name	Commands Used to Access	Command-Line Prompt
service access list configuration	**service access-list** command in global configuration mode	(config-service)#
subscriber configuration	**subscriber** command in context configuration mode	(config-sub)#
T1 channel configuration	**t1** command in port configuration mode	(config-t1)#
tunnel circuit configuration	**tunnel-circuit** command in tunnel map configuration	(config-tun-circuit)#
tunnel map configuration	**tunnel map** command in global configuration mode	(config-tunnel)#

To see a list of all CLI commands available in any mode, type a question mark (?) at the system prompt in the mode of interest. You can also type the question mark at any time while entering a command. Doing so displays the list of valid choices for the next keyword in the command. Liberal use of the question mark functionality is an easy and effective way to explore the command syntax.

You can also use the Tab key in any mode to carry out command completion. Partially typing a command name and pressing the Tab key causes the command to be displayed in full to the point where a further choice has to be made.

In all modes, the system recognizes and accepts partially typed command keywords, provided a sufficient amount has been entered to uniquely recognize it. For example, rather than typing **configure**, typing **conf** causes the CLI to enter configuration mode. However, if you enter the string **co**, an error is returned, because insufficient characters have been entered to distinguish between the **configure** command and the **copy** command.

Keywords in commands are not case-sensitive. For example, the **show version** command would be accepted if entered in any of the following ways:

show version

SHOW VERSION

Show Version

However, values that you provide for arguments are case-sensitive. For example, if you supply "Customers" for the *name* argument in the **l2tp-group name** *group-name* command, the AOS software would not recognize the name "customers" as the same Layer 2 Tunneling Protocol (L2TP) group.

Almost every configuration command also supports the **no** keyword. Typing the **no** keyword in front of a command disables the function or removes a command from the configuration. For example, to enable the RIP routing protocol in a context, enter the **router rip** command in context configuration mode. To subsequently disable the Routing Information Protocol (RIP) process and remove the command from the configuration, enter the **no router rip** command.

Auto-More

Automatic pagination of output at the CLI for console, Telnet, and Secure Shell (SSH) sessions is supported. The AOS prints "--More--" to indicate the presence of more output. You can use a subset of the commands available in the UNIX **more(1)** command, including text searching functions; see Table 2-2 for a list of commands supported by the AOS software. The **terminal length** and **terminal width** commands in exec mode and the **length** and **width** commands in line configuration mode allow you to specify terminal size to correctly paginate the output.

Table 2-2 Auto-More Commands

Input	Function
q	Skips all remaining output and returns to the CLI prompt
Enter	Displays one additional line out output
Space	Displays the next page of output
/pattern	Skips output until the text matching the regular expression specified by the *pattern* argument is found
-*pattern*	Displays all output excluding lines that contain text that matches the regular expression specified by the *pattern* argument
+*pattern*	Displays only output lines that contains text that matches the regular expression specified by the *pattern* argument

Command-Line History and Command Editing

The AOS software maintains a list of previous commands that you can step through by pressing the up arrow and down arrow keys, and then pressing **Enter** (**Return**) to enter the command.

The AOS software also supports Emacs-style command editing. Some of the available commands are listed in Table 2-3. The syntax **Ctrl+p** means press the **p** key while holding down they keyboard's **Control** key (sometimes labeled **Ctl** or **Ctrl**, depending on the keyboard and operating system of your computer). Similarly, **Esc+f** means holding down the **Escape** key (often labeled **Esc** on many keyboards) and typing the **f** key.

Table 2-3 Emacs-Style Keyboard Command Shortcuts

Keyboard	Description
Ctrl+p or up arrow	Recalls previous command in the command history
Ctrl+n or down arrow	Recalls next command in the command history
Ctrl+f or right arrow	Moves cursor forward one character
Ctrl+b or left arrow	Moves cursor backward one character
Esc+f	Moves cursor forward one word
Esc+b	Moves cursor backward one word

Table 2-3 Emacs-Style Keyboard Command Shortcuts

Keyboard	Description
Ctrl+a	Moves cursor to beginning of line
Ctrl+e	Moves cursor to end of line
Ctrl+k	Deletes to end of line
Ctrl+u	Deletes to beginning of line
Ctrl+d	Deletes character
Esc+d	Deletes word
Ctrl+c	Quits editing the current line
Ctrl+l	Refreshes (redraws) the display
Ctrl+t	Transposes characters

For more information on Emacs key bindings, see the GNU Emacs documentation available at http://www.gnu.org.

Accessing the AOS

This chapter provides an overview of accessing the Access Operating System (AOS) software and describes the tasks involved in configuring the system for local access through the console port or remote access through Telnet or Secure Shell (SSH), how to establish a local or remote session, and how to configure terminal settings for local and remote sessions. For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "Terminal Settings and Telnet Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter contains the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

You can access the command line-interface (CLI) of the AOS via any of the following:

- System console port—This is a special port to which you can directly connect a terminal.
- Telnet—Telnet provides remote access to the AOS CLI.
- Secure Shell—Secure Shell (SSH) provides secured remote access to the AOS CLI.

The AOS supports up to eight concurrent remote (Telnet, SSH, and HTTP) sessions.

This chapter describes how to configure the system for local access through the console port, or remote access through Telnet or SSH, how to establish a local or remote session, and how to configure terminal settings for local and remote sessions.

Configuration Tasks

This section describes how to log on to the console port and configure console port parameters, how to configure Telnet, and how to establish a remote session and configure session parameters. The following tasks are described:

- Log On to the Console Port
- Configure the Console Port
- Enable Remote Access
- Configure SSH
- Establish a Remote Session
- Clear a Remote Session

Log On to the Console Port

On the Subscriber Management System (SMS) 10000, the console port is located on the System Manager (SM) module. The SMS 10000 can support redundant SM modules. In a redundant configuration, only one SM module is active, and you can only log on to the console port on the active SM module.

Before configuring the system, the console is not secured and a session can be initiated by simply pressing **Enter** (**Return**). See the hardware guide for your system for information on connecting and configuring a terminal for use with the console port.

To secure the console and allow for remote access via Telnet or SSH, you must configure one or more administrator accounts; see the "Configure Operators and Administrators" section in Chapter 6, "Configuring Contexts," for instructions on how to define operators and administrators.

After you have configured at least one operator or administrator, the next time a local or remote session is initiated, the system prompts for a username and password. The user must enter a username and password at the appropriate prompts to gain access. The username is of a structured form established by the **aaa username-format** and the **aaa default-domain** commands in global configuration mode (see Chapter 40, "Configuring AAA"). This instructs the system which domain to use for authentication. It can include a domain for a console logon, but the domain name is optional—if a domain name is not supplied, the local context is assumed.

When connecting to the system either via the console or a remote session, the password entered is not echoed. In addition, passwords are stored in the configuration file with strong encryption.

If a password is forgotten, the particular account configuration record must be deleted and a new one entered.

Note When you type the **enable** command to change from operator exec mode to administrator exec mode, the system prompts for a password. This password is the same as the administrator logon password.

Configure the Console Port

The AOS software provides default settings for console sessions. If you would like to customize these settings, perform the following optional tasks:

- Configure Terminal Settings for the Current Session
- Configure Default Terminal Settings for the Console Port

Configure Terminal Settings for the Current Session

You can configure the terminal length and width for the current session. To do so, enter the appropriate command in operator exec mode:

terminal length *length*

terminal width *width*

Configure Default Terminal Settings for the Console Port

You can also configure a default terminal length and width to be used for all console port sessions. Follow these steps to modify the default terminal settings:

1. To enter line configuration mode, enter the following command in global configuration mode:

   line console

2. To modify the default terminal length for console port sessions, enter the following command in line configuration mode:

   length *length*

3. To modify the default terminal width for console port sessions, enter the following command in line configuration mode:

   width *width*

Enable Remote Access

You can configure the AOS software to allow remote access (Telnet and SSH) to the system. The AOS software supports up to eight concurrent remote sessions. Remote access is disabled by default. To enable remote access on the system, perform the following tasks:

- Create Administrators and Operators
- Configure the Management Port
- Configure Default Terminal Settings
- Reserve Remote Sessions

Create Administrators and Operators

To enable remote access, you must configure operators and administrators on the system. For more information, see the “Configure Operators and Administrators” section in Chapter 6, “Configuring Contexts.”

Configure the Management Port

The management port is an Ethernet port on the system that is designated for system management. The location of this port varies depending on the hardware platform:

- The management port on an SMS 500, SMS 1000, or SMS 1800 is located on the Control Engine (CE) module in slot 0. The management port is specified as 0/0.
- The management port on an SMS 10000 is located on an Ethernet Management module that is associated with a System Management (SM) module. The SMS 10000 supports redundant SM modules (SM2 and SM3), as well as redundant Ethernet Management modules. The Ethernet Management module associated with SM2 is in slot 4; the Ethernet Management module associated with SM3 is in slot 6. In a redundant configuration, only one SM module and one Ethernet Management module is active at a time. The active management port on a system is port 0 on the active Ethernet Management module. Use the following guidelines when configuring the management port on an SMS 10000:
  - The management port is 4/0 when SM2 is active and 6/0 when SM3 is active.
  - The AOS accepts configuration commands for either port 4/0 or 6/0, regardless of which SM module is active. AOS always applies these commands to the active management port.

Note The configuration for 4/0 and 6/0 must be the same. If you edit a configuration file offline, ensure the configuration for these ports is the same. The port configuration commands for 4/0 and 6/0 will be executed in order on the active management port, and if the configuration for these ports is not consistent, you might not achieve the expected configuration.

  - The **show configuration** administrator exec command shows identical configuration information for port 4/0 and 6/0.
  - The **save configuration** administrator exec command saves identical configuration information for port 4/0 and 6/0.

To configure the management port, perform the following tasks:

1. Configure the Management Interface
2. Configure Management Port Settings
3. Bind the Management Port
4. Enable the Management Port

Configure the Management Interface

Interfaces are created as part of a context. Typically, the management interface is configured in the special context named “local”. Follow these steps to configure an interface to use for the management port:

1. To enter context configuration mode and configure the local context, enter the following command in global configuration mode:

   context local

   For additional information on configuring contexts, see Chapter 6, “Configuring Contexts.”

2. To create a management interface and enter interface configuration mode, enter the following command in context configuration mode:

   interface *if-name*

3. To assign an IP address to the interface, enter the following command in interface configuration mode:

   ip address *ip-address* [*netmask*]

4. To enable the standard Ethernet Address Resolution Protocol (ARP) on the interface, enter the following command in interface configuration mode:

   ip arp arpa

   For additional information on the commands listed in step 2 through step 4, see Chapter 7, "Configuring Interfaces."

Configure Management Port Settings

To begin configuring the management port, enter the following command in global configuration mode:

port ethernet *slot/port*

The Ethernet port is configured with default values that enable it to operate correctly. To modify Ethernet port parameters, see Chapter 10, "Configuring Ethernet Ports."

Bind the Management Port

The management port will not become operational until you bind it to an interface. To bind the management port to the interface created earlier, enter the following command in port configuration mode:

bind interface *if-name ctx-name*

For additional information on configuring bindings, see Chapter 20, "Configuring Bindings."

Enable the Management Port

You must now enable the management port. To do so, enter the following command in port configuration mode:

no shutdown

Configure Default Terminal Settings

You can configure a default terminal length and width to be used for remote sessions. Follow these steps to modify the default terminal settings:

1. Enter the following command in global configuration mode to enter line configuration mode:

   line tty *start-tty* [**through** *end-tty*]

   The AOS supports up to eight remote sessions. You can use the **through** keyword to configure more than one line at a time.

2. To modify the default terminal length for remote sessions, enter the following command in line configuration mode:

   length *length*

3. To modify the default terminal width for remote sessions, enter the following command in line configuration mode:

   width *width*

Reserve Remote Sessions

You can configure the AOS software to reserve remote sessions for a particular IP address or administrator. This guarantees that the specified number of remote (Telnet or SSH) sessions are available for use by the preferred IP address, or the preferred operator or administrator.

To reserve remote sessions for a particular IP address or username, enter the following command in global configuration mode:

administrator reserve {**ipaddress** *ip-address* | **name** *username*} **context** *ctx-name* **sessions** *sessions*

To display information on remote session reservations, enter the following command in operator exec mode:

show administrator reservations

Configure SSH

The AOS software supports SSH access to the CLI. Remote access to the CLI via SSH is similar to remote access via Telnet, in that operators and administrators use the same usernames and passwords stored in the AOS configuration file or in Remote Authentication Dial-In User Service (RADIUS). The difference is that with SSH, the interactive session is encrypted with the single DES encryption algorithm. This makes eavesdropping on usernames, passwords, and other data transmitted over the network very difficult.

Note This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

You must complete the tasks listed in the "Enable Remote Access" section, before you configure SSH. Configuring SSH consists of the following tasks:

- Enable SSH
- Disable SSH

Enable SSH

To create an encryption key and enable SSH on the system, enter the following command in administrator exec mode:

sshd keygen

This command generates a new SSH key and stores it in the file named /flash/sshd.key.

Disable SSH

To disable SSH and remove the key from the system, use the **delete** command in administrator exec mode to remove the key file from the system as follows:

delete /flash/sshd.key

Establish a Remote Session

After you have configured the management port and one or more operators or administrators, you can establish a Telnet or SSH session to the system. There are many tools that provide Telnet access to remote systems. These tools are beyond the scope of this document. In general, you must provide the following information to establish a remote session to the AOS:

- System name or IP address—Enter the hostname configured for the system, or the IP address configured for the system.
- Username—Enter the name of a configured operator or administrator in the following format: *username@ctx-name*
- Password—Enter the password for the specified operator or administrator.

Once you are logged on the system, you will have access to the CLI, based on whether you are logged on as an operator or administrator and to which context you are logged on. This section describes how to:

- Configure Terminal Settings for the Current Session
- Display System Events During a Remote Session
- End a Remote Session
- Enable Telnet Debugging
- Enable SSH Debugging

If you are unable to establish a Telnet session, log on to the console port, and see the "Enable Telnet Debugging" subsection.

If you are unable to establish an SSH session, log on to the console port, and see the "Enable SSH Debugging" subsection.

Configure Terminal Settings for the Current Session

To configure the terminal length and width for the current session, enter the following commands in operator exec mode:

terminal length *length*

terminal width *width*

These commands override the system default values, or the default values specified for the session.

Display System Events During a Remote Session

You can configure the AOS to enable logging of events to your remote session. Enter the following command in operator exec mode to view the event log output:

terminal monitor [**circuit** {*slot/port* [*vpi vci* | [**hdlc-channel** *name*] *dlci*] | **lac** *vcn* | **lns** *vcn* | **pppoe** [*cm-slot-*]*session-id* [**cm** *slot*]} [**only**]]

The optional keywords and arguments enable you to limit the output to particular circuits or sessions of interest.

End a Remote Session

To end your remote session, you must first be in exec mode. If you are in configuration mode, enter the following configuration mode command to return to exec mode:

end

To end the exec session, enter the following command in operator exec mode:

exit

Enable Telnet Debugging

If you experience problems with accessing the system using Telnet, you can log on to the console port and enable debugging of the Telnet protocol to help isolate the source of the problem. To enable Telnet debugging messages, enter the following command in administrator exec mode:

debug ip telnet

Enable SSH Debugging

If you experience problems with accessing the system using SSH, you can log on to the console port and enable SSH debugging messages to help isolate the source of the problem. To do so, enter the following command in administrator exec mode:

debug sshd [*debug-level*]

Clear a Remote Session

You can end a remote session on the system. To do so, enter the following command in operator exec mode:

clear tty *num*

Configuration Examples

The following example displays a sample configuration for the management port on an SMS 500. An administrator named `admin1` is also configured. Two remote sessions are reserved for `admin1`.

```
[local]RedBack>enable

Password:
[local]RedBack#configure
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#interface mgmt
[local]RedBack(config-if)#ip address 192.168.110.1 255.255.255.0
[local]RedBack(config-if)#ip arp arpa
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#ip route 0.0.0.0 0.0.0.0 192.168.110.254 mgmt
[local]RedBack(config-ctx)#administrator admin1 password SooperSecret
[local]RedBack(config-admin)#exit
[local]RedBack(config-ctx)#exit
[local]RedBack(config)#port ethernet 0/0
```

```
[local]RedBack(config-port)#bind interface mgmt
[local]RedBack(config-port)#no shutdown
[local]RedBack(config-port)#exit
[local]RedBack(config)#administrator reserve 2 admin1 local
```

The following example configures new default terminal settings for all console and remote sessions:

```
[local]RedBack(config)#line console
[local]RedBack(config-line)#length 40
[local]RedBack(config-line)#width 60
[local]RedBack(config-line)#line tty 1 through 8
[local]RedBack(config-line)#length 40
[local]RedBack(config-line)#width 60
[local]RedBack(config-line)#
```

There are many different tools that provide Telnet access to a system. The following example initiates a Telnet session to the system named `RedBack` from a UNIX system. The administrator `admin1` types the password `SooperSecret` to log on. The password is not echoed by the AOS.

```
unix>telnet RedBack

Connected to RedBack.
Escape character is '^]'.

Username:admin1@local
Password:
[local]RedBack>
.
.
.
[local]RedBack>exit
```

System Images and Configuration Files

This chapter describes how to upgrade the system image, view, modify, and save system configurations, and load the Access Operating System (AOS) across the network. For a complete description of the commands discussed in this chapter, see the "System Image and Configuration File Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

All Subscriber Management System (SMS) devices are shipped with both a default image and default configuration file. These files are stored on the device, on the local file system.

The system image file contains the software that runs the AOS. By default, the system automatically loads the system image named redback.bin from the local file system. This file is loaded on the file system at the factory.

Configuration files are a script of configuration commands that can be loaded into the system. Configuration files can contain partial configurations and more than one can be read at any time. This allows you to keep sequences of commands that may be required from time to time. By default, the system automatically loads the system configuration file called redback.cfg from the local file system. This file is loaded on the file system at the factory.

If the file does not exist, the system automatically generates a minimal configuration. This minimal configuration only contains a list of all available ports in the shutdown state, present in the system at the time it was powered up. You can then begin to modify the configuration. You can modify the active system configuration in one of two ways. First, you can change the system configuration interactively. Interactive configuration consists of beginning a command-line interface (CLI) session, using the **enable** command in operator exec mode to enter administrator exec mode, and then entering configuration mode by typing the **configure** command in administrator exec mode. Once in configuration mode, you can enter any number of configuration commands.

You can also create and modify configuration files offline. The AOS supports comment lines within configuration files. To add a comment to your configuration file, simply begin the line using the **!** key. When you load a configuration file, any line that begins with the **!** key is not processed as a command.

System images and configuration files can be stored on the SMS device local file system or stored on a remote server and accessed by either the Trivial File Transfer Protocol (TFTP) or File Transfer Protocol (FTP). Transferring files of any type between the file system and a remote machine requires either the TFTP or FTP. The AOS expects to get files from a server or put files to a server. The server must be reachable through one of the system interfaces.

Understanding the Local File System

All SMS hardware platforms have local flash storage available for system image files and configuration files.

Depending on your hardware configuration, the local file system can contain:

- The flash file system—The Control Engine (CE) module on the SMS 500, SMS 1000, and SMS 1800 contains sufficient flash memory to retain system images, configuration files, and system log files. The flash system device name is /flash.
- The PCMCIA file system—Some CE modules on the SMS 500, SMS 1000, and SMS 1800 and all System Manager (SM) modules on the SMS 10000 contain a PCMCIA module that can support two PCMCIA Type-I or Type-II cards. Each PCMCIA slot supports Type-I or Type-II PC cards and can be used for synchronous RAM (SRAM) or additional flash memory. The PCMCIA devices are named /pcmcia0 and /pcmcia1.

See the appropriate hardware guide to determine the types of local flash available on your system.

By default, the system automatically loads the system image called redback.bin from the CE module's flash file system on the SMS 500, SMS 1000, and SMS 1800, and from the SM module's PCMCIA flash file system on the SMS 10000. This file is loaded on the file system at the factory. The primary command used to transfer files to or from the local file system is the **copy** command in administrator exec mode. The **copy** command expects two arguments in URL form. If you are writing over an existing file, a confirmation prompt appears. The file being overwritten is deleted once the copy process begins, even if the copy fails.

Redundant File Systems

You can configure an SMS 10000 with two SM modules. Only one SM module is active; the other operates in standby as a backup. The local file system on an SMS 10000 resides on the SM modules. The file system on the active SM module is referred to as the active file system; the file system on the backup SM module is referred to as the backup file system. Some commands on the local file system can be performed on files that are on the active SM module, a particular SM module, or both SM modules. See the "URLs" subsection for additional details.

URLs

Most AOS commands use a URL to access a file. For details on a particular command, see the "Usage Guidelines" section for a particular command in the *Access Operating System (AOS) Command Reference* publication.

When referring to a file on the local file system, the URL takes the following form:

[**file:**][*/sm*]*/device*[*/directory*]*/filename.ext*

On an SMS 10000 that is configured with redundant SM modules, you can specify a file on a device that resides on the active SM module, on a particular SM module, or on both the active and backup SM modules using the optional */sm* argument as follows:

- Specify /**sm** to specify the file system on the active SM module.
- Specify /**sm2** or /**sm3** to specify the file system on a particular SM module.
- Omit the */sm* argument in a source file to specify a file on the active SM module; omit the */sm* argument for a destination file to specify the file system on both the active and backup SM module.

There are restrictions for specifying the SM module for some commands. For details on a particular command, see the "Usage Guidelines" section for a particular command in the *Access Operating System (AOS) Command Reference* publication.

The *device* argument can be **/flash**, **/pcmcia0**, or **/pcmcia1**, depending on your hardware configuration.

When referring to a file on a Trivial File Transfer Protocol (TFTP) server, the URL takes the following form, where the *ip-address* argument is the IP address, or the *hostname* argument is the hostname of the TFTP server:

tftp://{*ip-address* | *hostname*}[*/directory*]*/filename.ext*

When referring to a file on a File Transfer Protocol (FTP) server, the URL takes the following form, where the *username:passwd* construct specifies the user and an optional password, the *ip-address* argument is the IP address of the FTP server, and the *hostname* argument is the hostname of the FTP server. The **passive** keyword specifies a passive FTP transaction.

ftp://*username*:*passwd*@{*ip-address* | *hostname*}[*/directory*]*/filename.ext* **passive**

The *hostname* argument for TFTP and FTP can only be used if the Domain Name System (DNS) is enabled via the **ip domain-lookup**, **ip domain-name**, and **ip name-servers** commands in context configuration mode. See Chapter 28, "Configuring DNS."

Configuration Tasks

Managing and loading system images and configuration files typically consists of the following tasks:

- Configure the Boot System Image File
- Set the Boot Configuration File
- Load a Configuration File
- Save a Configuration to a File
- Display Configuration Information
- Reload the System
- Upgrade the System Image
- Manage Files and Directories
- Format PCMCIA SRAM Devices

- Configure Switch Fabric Modules
- Hot-Swap an I/O Module
- Display AOS Version Information
- Enable IP TFTP Debugging Messages

Configure the Boot System Image File

You can configure the AOS with a list of system images to use at the next system boot time. The system attempts to load the first image in the list and, if that image fails, tries to load the next image, and so on. The image files can be stored locally on the system, or on a remote TFTP or FTP server.

To specify the location of a system image to use at the next system boot, enter the following command in global configuration mode:

boot system {**bootp** | *url* **local** *local-ipaddr* [**gateway** *gw-ipaddr*]}

Note On systems configured with redundant SM modules, you must specify the active SM module in the URL for a local system image file. See the "URLs" section in this document for additional details for additional details on the *url* argument format.

To remove an image from the list, enter the following command in global configuration mode:

no boot system {**bootp** | *url* **local** *local-ipaddr* [**gateway** *gw-ipaddr*]}

The arguments you specify in the **no** form of the command must match those that you specified when adding the image.

To return the system to using the default system image, enter the following command in global configuration mode:

default boot system

This command removes any other previously specified boot system files from the configuration.

Set the Boot Configuration File

You can configure the AOS with a list of configuration files to be read at the next system boot time. The system reads each file successively until all the files and the commands they contain are executed in to the system's configuration. The configuration files can be stored locally on the system or on an TFTP server.

To specify a boot configuration file, enter the following command in global configuration mode:

boot configuration *url*

Note On systems configured with redundant SM module, you must specify the active SM module in the URL for a local boot configuration file. See the "URLs" section in this document for additional details for additional details on the *url* argument format.

To remove a boot configuration file from the list, enter the following command in global configuration mode:

no boot configuration *url*

The argument you supply in the **no** form of the command must match those that you specified when adding the configuration file.

To return the system to using the default system image, enter the following command in global configuration mode:

default boot configuration

This command removes any other previously specified boot configuration files from the configuration.

Load a Configuration File

You can configure the system by instructing the system to read one or more configuration files. A configuration file is simply a script file that contains a sequence of configuration commands. When reading a configuration file, the system parses and applies each command to the system sequentially, as if you had entered it interactively. If an error is encountered, the system displays a message and continues parsing the file with the next command. Configuration files can contain partial configurations and more than one can be read at any time. This allows you to keep sequences of commands that may be required from time to time.

If the system encounters a line in the configuration file that begins with the **!** key, the system considers that line to be a comment, and does not process the line.

The system does not reboot the system when loading a configuration file.

To configure the system from a preexisting configuration file, enter the following command in administrator exec mode:

configure *url* [**verbose**]

Note On systems configured with redundant SM modules, you must specify the active SM module in the URL for the **configure** command when specifying a local configuration file.

The **verbose** keyword causes the AOS to display each line and its line number as it loads a configuration file.

Save a Configuration to a File

To save the current configuration of the device to a file, enter the following command in administrator exec mode:

save configuration *url* [**verbose**] [**-noconfirm**]

Note If you overwrite an existing configuration file, you lose all existing information in that configuration file, including any comments. Also, the AOS may reorder the command sequence.

Display Configuration Information

You can display the current configuration of the device, or a previously saved configuration. To display configuration information, enter the following command in administrator exec mode:

show configuration [*url* | **verbose**] [**context** *ctx-name*] [**port** *slot/port*]

Note You cannot specify a file on the backup SM module as the *url* argument for the **show configuration** command.

The optional **context** and **port** keywords enable you to view only the configuration information for the specified context name or port number.

Reload the System

To reboot the system, enter the following command in administrator exec mode:

reload

The system prompts you to confirm the reload. Type the **y** key to proceed with the reload, or the **n** key to cancel the reload.

Note You should reload the system from the console port. All remote sessions to the system are disconnected during a reload.

If you proceed with the reload, the system uses the boot system images and boot configuration files specified in the current configuration of the device.

After entering the **reload** command at the system console, you have approximately three seconds to interrupt the reload process and modify system parameters by typing any key. The system displays the following message:

```
Press any key to stop auto-boot...
 3
```

If you type any character during the three seconds before the boot process begins, the boot process is interrupted, and the system displays a boot prompt:

```
[RedBack Boot]:
```

(If the keys are not typed sufficiently fast enough, the system attempts to boot normally and fails. Repeat this process if necessary.)

From this prompt, you can do the following:

- Print Boot Parameters
- Change Boot Parameters
- Proceed with Reboot

Print Boot Parameters

At the prompt, type the **p** key followed by pressing **Enter** (**Return**) to display the current boot parameters, including the boot device, processor number, filename, remote server IP address, management port IP address, boot flags, and startup script.

Change Boot Parameters

To modify one or more of the boot parameters, enter the character **c** followed by pressing **Enter (Return)**. The system prompts you for each boot parameter. The prompt includes the current value for the parameter. To keep the currently specified value, simply press **Enter (Return)**. Otherwise, enter the new value, followed by pressing **Enter (Return)**.

You can change the following parameters:

- boot device—Depending on your system, the available boot devices can be as follows:
  - **dc**—Indicates network loading
  - **flash**—Indicates the local flash file system
  - **pcmcia**—Indicates the local pcmcia flash file system
- host name—Indicates the host name of the remote server.
- file name—Indicates the name of the system image to be loaded.
- inet on ethernet (e)—Indicates the IP address on the system management port.
- host inet (h)—Indicates the IP address of the remote server.
- gateway inet (g)—Indicates the IP gateway address to use, if the IP address of the remote server is not on the same subnet as the system.
- user (u)—Specifies the username for FTP loading
- ftp password (pw)—Specifies the password for FTP loading
- flags (f)—Indicates flags to be used during boot. These include:
  - 0x04—Specifies that the system should not auto-boot
  - 0x08—Specifies that the system should perform a quick boot, with no countdown.
  - 0x10—Specifies that the system should not automatically load any startup scripts after booting.
  - 0x20—Specifies that the system should disable logon security.
  - 0x40—Specifies that the system should use BOOTP to get boot parameters.
  - 0x80—Specifies that the system should use the Trivial File Transfer Protocol (TFTP) to get the boot image.
  - 0x100—Specifies that the system should use proxy Address Resolution Protocol (ARP).

  A flag of **0x0** is the default and normally instructs the system to boot from flash. You can specify a combination of these flags by adding the hexadecimal values. For example, if you want to use BOOTP to get the boot parameters and TFTP to get the boot image, specify a boot flag value of **0xC0** (the sum of 0x40 and 0x80).
- startup script(s)—Specifies one or more configuration files to be read upon startup.

Note You must use the active Ethernet management port on the system (see Chapter 3, "Accessing the AOS," for information on configuring the system management port). The system cannot load from any other system port. The device name for this port is **dc**.

Note Any changes you make to the boot parameters are permanently recorded by the system. After the system has initialized, you must configure the system, and then either copy a valid system image to flash or verify that one exists. The boot parameters must be modified to return to the default boot device of flash.

Proceed with Reboot

After you have modified the boot parameters to suit your needs, you can proceed with the system reload by typing the @ key, and then pressing **Enter (Return)**.

Upgrade the System Image

There are several ways to upgrade the system:

- Copy a New System Image to the Local File System
- Network-Load a New System Image

Copy a New System Image to the Local File System

The primary command used to transfer files to or from the local file system is the **copy** command in administrator exec mode. The **copy** command expects two arguments in URL form. If you are writing over an existing file, the system prompts you for confirmation. The file being overwritten is deleted once the copy process begins, even if the copy fails.

To upgrade the system image, perform the following steps:

1. Verify that there is enough space to copy a file to the local file system. To view the contents of the local file system, enter the following command in administrator exec mode:

   directory *url*

   Specify one of the flash file system devices available on your hardware platform (/flash, /pcmcia0, or /pcmcia1) for the *url* argument. The last line of the output indicates the number of bytes available on the specified device. For example:

   ```
   [local]RedBack#dir flash

   size    date         time      name
   ----    ----         ------    ------
   2073    APR-12-2001  10:13:18  OLD.CFG
   4157792 APR-12-2001  09:04:22  REDBACK.BIN
   2085    APR-12-2001  10:13:50  REDBACK.CFG
   2139    APR-12-2001  08:55:20  TFTP.CFG
   total bytes: 7347712, used bytes:4197888, free bytes: 3149824
   ```

2. If there is sufficient space on the local file system, make a copy of the old image. The following example copies the factory default filename of `/flash/redback.bin` to a file named `/flash/old.bin`.

   ```
   [local]RedBack#copy /flash/redback.bin /flash/old.bin
   ```

Note If there is not sufficient space on the local file system, you must boot off the network as described in the "The following example shows a reload from the console that is interrupted. The example first prints out the current settings for the system, then modifies them to boot from the network using the `/tftpboot/redback.bin` filename. The example modifies the boot device, filename, local Ethernet address, gateway, and flags, and then reloads the system." section in this document. Ensure that the image works correctly before you copy it to your system.

3. Copy the new image from the server to the local file system. In the following example, the Internet Protocol (IP) address of the TFTP server is `10.1.1.1`, and the new image is stored in a file named `redback.bin` in the default directory on the TFTP server:

   ```
   [local]RedBack#copy tftp://10.1.1.1/redback.bin /flash/redback.bin
   ```

   Information about the connection to the server is shown; an exclamation point (**!**) is printed for each packet transmission to show the progress of the transfer.

   If the file transfer fails for any reason, copy the file you saved in step 2 to `/flash/redback.bin` to leave the system in a state where it can reload the original image, if necessary. Then troubleshoot the server and the network to locate the source of the problem.

4. Reload the system by typing the following command:

   ```
   [local]RedBack#reload
   ```

Note If you encounter problems transferring files to or from an SMS device, first verify that the server is reachable by pinging its IP address from the system. If this is OK, verify that the file and directory being accessed on the server has the appropriate read/write protections.

Network-Load a New System Image

You can load the AOS software over a network connection using BOOTP or TFTP to transfer the image directly to the dynamic RAM (DRAM) on the Subscriber Management System (SMS) device. This process may be required, for example, if the "redback.bin" system image was deleted by a system administrator, and the system was subsequently powered off before a new image could be transferred to the local file system. You may also want to use this as the normal load procedure when you expect frequent changes to the AOS.

Changing the default boot process consists of interrupting the normal system boot task, and subsequently instructing the SMS device to use BOOTP or TFTP to obtain its image, rather than to look for it in flash. See the "Reload the System"section in this document for complete instructions.

Before you attempt this procedure, verify that a BOOTP server and a TFTP server are reachable on the network from the active Ethernet management port on the system.

Note The process of setting up a BOOTP and TFTP server is beyond the scope of this guide. Consult the technical publications provided with your management platform for details.

Manage Files and Directories

The following tasks are typically used to manage files and directories on the local file system:

- Create Directories
- Remove Directories
- Copy Files
- Delete Files
- Rename Files
- Display Files

Create Directories

To create a new directory on the local file system, enter the following command in administrator exec mode:

mkdir *directory*

Remove Directories

To remove a directory from the local file system, enter the following command in administrator exec mode:

rmdir *directory*

Copy Files

You can copy files from either a TFTP or FTP server to the SMS device; from the SMS device to a TFTP or FTP server; or from one location to another on the local SMS file system. To copy a file, enter the following command in administrator exec mode:

copy *url1 url2* [**passive**] [**-noconfirm**]

On a system configured with redundant SM modules, you can also use the **copy** command in administrator exec mode to copy the entire contents of a device on the active SM module to a device on the backup SM module or from the backup SM module to the active SM module. To do this, include the appropriate /*sm* argument and the device in both the source file and destination file URLs. The following example copies the entire pcmcia0 device from SM2 (the active device) to pcmcia0 on SM3:

```
[local]RedBack#copy /sm2/pcmcia0 /sm3/pcmcia0
```

Delete Files

To remove a file from the local file system, enter the following command in administrator exec mode:

delete *url* [**-noconfirm**]

Rename Files

To rename a file or directory on the local file system, enter the following command in administrator exec mode:

rename *source target* [**-noconfirm**]

Display Files

To display a list of files on a local file system, enter the following command in administrator exec mode:

directory *url* [**-size** | **-time**] [**-reverse**]

Format PCMCIA SRAM Devices

You can configure the system to use a PCMCIA SRAM card for additional nonvolatile storage for Dynamic Host Control Protocol (DHCP) secured ARP information. To do so, enter the following command in administrator exec mode:

format [*/sm*]*device* **dhcp-secured-arp**

See Chapter 29, "Configuring DHCP," for additional information on DHCP.

Configure Switch Fabric Modules

The SMS 10000 hardware platform can be configured with up to four Switch Fabric modules: A, B, C, and D. When the system is functioning normally, the default Switch Fabric modules (A, B, and C) are operational and Switch Fabric module D is in standby mode. If one of the default Switch Fabric modules fails, fabric D becomes operational. For example, if Switch Fabric module B fails, the system uses Switch Fabric modules A, C, and D. The system does not automatically switch back to the default Switch Fabric modules A, B, and C. In the previous example, the system continues to use Switch Fabric modules A, C, and D, even if you replace Switch Fabric module B.

You can configure the system so that it reverts back to the default Switch Fabric modules (A, B, and C) when a failed Switch Fabric module becomes operational again. If you configured the system to revert to the default Switch Fabric modules, then in the previous example, when you replace Switch Fabric module B, the system automatically switches back to using Switch Fabric modules A, B, and C.

To configure the system so that it reverts to the default fabric, enter the following command in global configuration mode:

fabric revert

If you configure a system so that it does not revert to the default Switch Fabric modules, you can always override this setting and manually switch back to the default Switch Fabric modules. To manually switch to the default Switch Fabric modules, enter the following command in administrator exec mode:

fabric revert

The following example shows how to configure an SMS 10000 so that it automatically switches back to default Switch Fabric modules A, B, and C:

```
[local]RedBack#config
[local]RedBack(config)#fabric revert
[local]RedBack(config)#exit
```

With this configuration, if there is a problem with Switch Fabric module B that causes the system to switch to Switch Fabric modules A, C, and D, the system automatically switches back to Switch Fabric modules A, B, and C when Switch Fabric module B is replaced.

The following example shows how to manually switch the fabric back on a system that is not configured to automatically switch back:

```
[local]RedBack#fabric revert
```

Hot-Swap an I/O Module

The hot-swap feature in AOS enables you to replace an I/O module without interrupting operations on a running system. On systems and I/O modules that are equipped with ejector tabs, simply remove the module you want to replace, and insert a new module. If you insert the same module type into the slot, the system automatically configures the module; if you insert a different module type into the slot, you must manually configure the new module.

On systems or modules that are not equipped with ejector tabs, you must perform the following tasks to hot-swap an I/O module:

1. Prepare the Module for Extraction
2. Replace the Module
3. Verify the New Module Installation
4. Configure the New Module

Prepare the Module for Extraction

Follow these steps to prepare to extract an I/O module without ejector tabs from the system:

1. To shut down the ports on the module, remove all port, circuit, and binding information for the module from the current configuration of the device, and place the ports into the EXTRACT_READY state, enter the following command in administrator exec mode:

   module extract *slot*

Note The amount of time this process takes varies depending on the configuration.

Note You can undo this command by entering the **no module extract** command in administrator exec mode. Otherwise, you must either remove the module and install a new module or restart the system for the slot to become usable again.

2. To verify that the ports on the module are in the EXTRACT_READY state, enter the following command in operator exec mode:

   show port table

Note Never remove an I/O module unless all ports are in the EXTRACT_READY state.

Replace the Module

After you have issued the **module extract** command and all of the ports on the module are in the EXTRACT_READY state, you can remove the module and install a new module. Always follow proper ESD protection procedures to avoid damaging the module. Please see the hardware guide for your system for complete instructions on removing and installing modules.

When you insert the new module, the system automatically detects the new module and performs initialization. This initialization process may take up to 30 seconds. If the new module is the same type as the module you have removed, the system automatically configures the module with the port parameters stored in memory.

Verify the New Module Installation

Follow these steps to verify that the new module has been installed correctly:

1. To confirm that the new module and ports are recognized by the system, enter the following command in operator exec mode:

   show hardware

2. If you installed a module type that is the same as the module you removed earlier, enter the following command for each port on the module to verify that the ports have been configured correctly:

   show configuration port *slot/port*

   If you installed a module type that is different from the module you removed earlier, the AOS does not load any configuration for the slot, and places the ports in the SHUTDOWN/UNCONFIGURED state. Verify that the ports are in the SHUTDOWN/UNCONFIGURED state. To view the port states, enter the following command in operator exec mode:

   show port table

Configure the New Module

If you installed a module type that is the same as the module you removed, the AOS automatically configures the module from system memory. If the new module is not the same type as the module you removed, you must configure the ports on the module. To do so, enter the following command in administrator exec mode:

configure [*url* [**verbose**]]

Display AOS Version Information

To display the version of AOS running on the system, enter the following command in operator exec mode:

show version

The display also includes the following:

- System uptime
- System reload date and time
- Image file loaded
- Configuration files read at reload

Enable IP TFTP Debugging Messages

If you experience problems loading, copying, or otherwise accessing a remote file using TFTP, you can enable TFTP debugging messages to help determine the cause of the problem. To enable the logging of TFTP debugging messages, enter the following command in administrator exec mode:

debug ip tftp

Configuration Examples

The following example shows a reload from the console that is interrupted. The example first prints out the current settings for the system, then modifies them to boot from the network using the `/tftpboot/redback.bin` filename. The example modifies the boot device, filename, local Ethernet address, gateway, and flags, and then reloads the system.

```
[local]RedBack#reload

Proceed with reload? [confirm]y
```

The system displays loader version and copyright information before beginning the reload countdown:

```
Press any key to stop auto-boot...
 3

[RedBack Boot]: p

boot device          : flash
processor number     : 0
file name            : /flash/redback.bin
flags (f)            : 0x0
other (o)            : flash

[RedBack Boot]: c

'.' = clear field;  '-' = go to previous field;  ^D = quit

boot device          : flash dc
processor number     : 0
host name            :
file name            : /flash/redback.bin redback.bin
inet on ethernet (e) : 10.1.1.10
inet on backplane (b):
host inet (h)        :
gateway inet (g)     : 10.1.1.1
user (u)             :
ftp password (pw) (blank = use rsh):
flags (f)            : 0x0 0xc0
target name (tn)     :
startup script (s)   :
other (o)            :
```

```
[RedBack Boot]: p

boot device          : dc
processor number     : 0
file name            : redback.bin
inet on ethernet (e) : 10.1.1.10
gateway inet (g)     : 10.1.1.1
flags (f)            : 0xc0
other (o)            :

[RedBack Boot]:@
```

The system displays a similar message to the following if it is correctly loading:

```
Loading /tftpboot/redback.bin... 1554048 + 395560 + 486464
Image checksum verified.
Starting at 0x108000...
```

When the system has completed its initialization and is ready for use, it displays the following message:

```
Press <return> to connect...
```

Configuring Basic System Parameters

This chapter describes parameters that report basic system information to the operator or administrator.

This chapter contains the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

For detailed information on syntax and usage guidelines for commands listed in the "Configuration Tasks" section,m see the "Basic System Commands" chapter in the *Access Operating System (AOS) Command Reference*.

Overview

There are basic Access Operating System (AOS) parameters that need to be established when setting up a new system. The administrator needs to determine the system's hostname, set the location of the system, reference contact information, and set the clock. You can also customize privilege levels within the AOS software to provide different levels of access to the AOS command-line interface (CLI). This chapter describes these tasks along with how to enter the global configuration mode that allows these and many other parameters to be set.

Configuration Tasks

To configure basic system parameters, perform the tasks described in the following sections:

- Configure AOS Banner
- Configure System Contact Information
- Configure the System Hostname
- Configure the System Location
- Configure System Clock Settings
- Configure Privilege Levels

Enter Global Configuration Mode

The global configuration mode is the gateway to all other configuration modes that provide specific commands for the system.

To enter global configuration mode, enter the following command in administrator exec mode:

configure

Configure AOS Banner

You can create a message of the day (MOTD) that displays on all connected systems. It appears at logon and is useful for sending messages that affect administrators and operators, such as scheduled maintenance or system shutdowns. By default, no banner is present on logon.

To create an MOTD, enter the following command in global configuration mode:

banner motd *delimited-text*

where *delimited-text* is the message you want displayed.

Use the **no banner motd** command to delete the message.

Configure System Contact Information

System contact information tells the user how to contact the Information Service (IS) helpline. The contact information can be any alphanumeric string, including spaces, that is no longer than one line. By default, no system contact information is defined.

To configure contact information, enter the following command in global configuration mode:

system contact *text*

where *text* is the contact information.

Configure the System Hostname

The purpose of the system hostname is to identify a specific device. The default hostname for an SMS device is RedBack.

Hostnames can be no more than 63 characters and must comply with the guidelines established in RFC 1035, *Domain-Names—Implementation and Specification* and RFC 1178, *Choosing a Name for your Computer.*

To configure a hostname, enter the following command in global configuration mode:

system hostname *name*

Configure the System Location

The system location lets the administrator or operator know the physical location of a device. By default, no system location is set. When setting the location, the text can be any alphanumeric string including spaces that is no longer than one line.

To configure the system location, enter the following command in global configuration mode:

system location *text*

Configure System Clock Settings

This section describes the various system clock settings that are available. To configure clock settings, perform the tasks described in the following sections:

- Set the Clock
- Enable Summer Time Mode
- Set the Clock Time Zone
- Display Clock Information

Set the Clock

The administrator can set and preserve the time across system reloads. The clock specifies the year, month, day, hour, minutes, and seconds. The hour is in a 24-hour format.

To set the clock, enter the following command in administrator exec configuration mode:

clock set *yyyy:mm:dd:hh:mm*[*:ss*]

Specifying the seconds is optional.

Enable Summer Time Mode

Use the **clock summer-time** command to configure the system to automatically update to daylight savings time (summer time) and then to revert automatically to standard time.

To set the system to automatically switch between daylight savings and standard times, enter the following command in global configuration mode:

clock summer-time *zone1 zone2* **recurring** *week day month hh week day month hh* |
date *yyyy:mm:dd:hh:mm yyyy:mm:dd:hh:mm*

The keywords and arguments define the exact switchover times. For a detailed explanation of the syntax of this command, see Chapter 5, “Basic System Commands,” in the *Access Operating System (AOS) Command Reference* publication.

Set the Clock Time Zone

The **clock timezone** command allows the administrator to set one or more time zones and their distances from Universal Coordinated Time (UTC) for display purposes. The default time zone is UTC. If no time zone is configured with the **local** keyword, the system uses UTC when displaying time. To display one or more time zones, enter the following command in global configuration mode:

clock timezone *zone hours* [*minutes*] [**local**]

Display Clock Information

To display current time of day in local time, enter the following command in operator exec configuration mode:

show clock

To display the current time in UTC, add the **universal** keyword:

show clock universal

Configure Privilege Levels

The AOS supports up to 16 different privilege levels that can be used to provide operators and administrators with different levels of access to the AOS command-line interface (CLI).

Each command in the AOS CLI is assigned to a particular privilege level. By default, all operator exec commands are assigned to privilege level 3, and all administrator exec and configuration commands are assigned to privilege level 10. You can configure a different privilege level for any command available through the AOS CLI.

All AOS exec sessions run at a particular privilege level, determined by the configuration for each operator or administrator account. The operator or administrator has access to all commands that are assigned to the current privilege level or a lower privilege level. When an operator or administrator logs on to the system, the privilege level for the exec session is set to the initial privilege level configured for the operator or administrator. The operator or administrator can change the current privilege level, up to the maximum level configured for that operator or administrator, to access commands assigned to a higher privilege level than the starting privilege level.

There are two main tasks involved in configuring privilege levels:

- Assign a Privilege Level to a Command
- Configure Privilege Levels for an Operator or Administrator

From an exec session, an operator or administrator can:

- Change Current Privilege Level
- Display Current Privilege Level

Assign a Privilege Level to a Command

By default, all operator exec commands are assigned a privilege level of 3 and all administrator exec and configuration commands are assigned a privilege level of 10.

To assign a different privilege level to a particular command, enter the following command in global configuration mode:

privilege *mode* [**inherit**] **level** *level command*

Configure Privilege Levels for an Operator or Administrator

Operator and administrator accounts are created with a default starting privilege and maximum privilege level that you can modify to customize access to the AOS commands. See the "Configure Operators and Administrators" section in Chapter 6, "Configuring Contexts," for information on configuring administrator and operator accounts.

Change Current Privilege Level

To change the current privilege level for an exec session, enter the following command in operator exec configuration mode:

enable *level*

Display Current Privilege Level

To display the current privilege level for an exec session, enter the following command in operator exec configuration mode:

show privilege

Configuration Examples

The following example configures the AOS banner, contact information, hostname, and location settings:

```
[local]RedBack#configure
[local]RedBack(config)#banner motd /Welcome to Redback SMS/
[local]RedBack(config)#system contact IS Hotline 1-800-555-1567
[local]RedBack(config)#system hostname freebird
[local]freebird(config)#system location Building 3, 2nd Floor, Lab 3
```

The following example raises the privilege level for all **debug** exec commands to 8, the **configure** exec command to 12, the **context** global configuration command to 14, and the **reload** exec command to 14:

```
[local]RedBack(config)#privilege exec inherit level 8 debug
[local]RedBack(config)#privilege global inherit level 12 configure
[local]RedBack(config)#privilege configure level 14 context
[local]RedBack(config)#privilege exec level 14 reload
```

The following example shows an exec session for the adm-plus administrator:

```
Username:adm-plus@local
Password:
[local]RedBack>show privilege

Current privilege level is 3
[local]RedBack>enable 10
Password:
```

```
[local]RedBack#show privilege

Current privilege level is 10
[local]RedBack#exit
```

Part 2

Setting Up Contexts with Interfaces and Subscribers

Configuring Contexts

This chapter provides an overview of the Access Operating System (AOS) contexts and describes the tasks involved in performing basic context configuration through the AOS. For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "Context Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

One of the most advanced features of the AOS is the ability to provide dynamic service and provider selection to subscribers using high-speed access technologies.

Local Context

The Subscriber Management System (SMS) with a single context configured is similar to traditional networking products. This is referred to as a single-context configuration. Every configuration includes a special context named `local` that cannot be deleted. In single-context configurations, this is the only context. The local context allows you to do the following:

- Configure and examine other contexts.
- Configure global resources such as ports, the Simple Network Management Protocol (SNMP), and system logging.

Multiple Contexts

One of the most advanced features of the AOS is the ability to provide dynamic service and provider selection to subscribers using high-speed access technologies. The AOS provides this functionality by supporting multiple contexts.

Each AOS context is a virtual SMS instance running within a single physical device, providing a separate security, management, and operating environment on behalf of a given network. A context has its own IP routing table, its own Remote Authentication Dial-In User Service (RADIUS) client, and so on, and does not share this information with other contexts. By separating the address and name spaces in this way, service providers can use multiple contexts to manage subscribers and provide access for customers of different providers, or to provide different classes of services for customers. Service providers use a single physical SMS device to implement this, with one or more contexts being assigned to each service provider or service class.

Configuration Tasks

The basic tasks involved in configuring contexts are:

- Create a Context
- Configure Domain Names
- Configure Operators and Administrators
- Display Contexts
- Display IP Hosts
- Display IP Traffic
- Clear IP Counters
- Restrict System Access

After you have created a context and performed the basic configuration tasks, you can configure subscribers and networking protocols (such as bridging, routing and tunneling) for that context. See the appropriate chapters in this book for additional configuration information.

Create a Context

To create a new context and enter context configuration mode, enter the following command in global configuration mode:

context *name*

When you enter this command on a system configured with a Forwarding Engine (FE) module, the AOS determines the amount of memory available on the FE. If the context to be configured will consume most of the available memory, the AOS displays a warning message and then creates the context. If there is not enough memory to create the context, the AOS displays an error message and does not allow you to create the new context.

Use this command also to enter context configuration mode to modify an existing context.

Configure Domain Names

You can configure domain names that can be used as an alias for a context. Subscribers can then use this alias to log on to a particular context. To configure a domain name alias for a context, enter the following command in context configuration mode:

domain *alias* [**advertise**]

Configure Operators and Administrators

To secure the system and enable remote access to the system, you must create operator and administrator logon accounts. These accounts are created within a context, and are valid only for the specified context. Operators and administrators are allowed to log on directly to the console and through Telnet.

Configuring an operator or administrator account consists of the following tasks:

- Create an Operator or Administrator Account
- Configure the Session Timeout for an Account
- Configure Privilege Levels for an Account

Create an Operator or Administrator Account

Operators do not have privileges to run the **enable** command and, therefore, cannot view or modify the system configuration. Also, an operator's view of the system is limited to the context in which the operator account is defined.

To create or modify an operator account, enter the following command in context configuration mode:

operator *name* **password** *password*

Administrators can run the **enable** command and access additional information within the context that the administrator is defined. Administrator accounts created in the local context can modify the system configuration and view all system information.

To create or modify an administrator account, enter the following command in context configuration mode:

administrator *name* **password** *password*

Both the **operator** and **administrator** global configuration commands enter administrator configuration mode. In this mode, you can configure the operator or administrator account.

Configure the Session Timeout for an Account

You can configure an idle or absolute timeout for the operator's or administrator's remote and console sessions by entering the following command in administrator configuration mode:

timeout {**absolute** | **idle**} *minutes*

Configure Privilege Levels for an Account

When an operator or administrator logs on to the system, the exec session runs at the initial privilege level configured for the account. This allows the operator or administrator access to the AOS commands that are assigned a privilege level less than or equal to the initial privilege level. The operator or administrator can change the privilege level of an exec session up to the maximum privilege level configured for the account, allowing for greater access to the AOS command set.

See the "Configure Privilege Levels" section in Chapter 5, "Configuring Basic System Parameters," for information on how to assign a privilege level to a command.

By default, the initial privilege level for operators is 6 and the initial privilege level for administrators is 15.

To modify the initial privilege level for an operator or administrator, enter the following command in administrator configuration mode:

privilege start *level*

To display the privilege level for the current exec session, enter the following command in operator exec configuration mode:

show privilege

The maximum privilege level specifies the highest privilege level that the operator or administrator can run. By default, the maximum privilege level for operators is 6 and the maximum privilege level for administrators is 15.

To modify the maximum privilege level for an operator or administrator, enter the following command in administrator configuration mode:

privilege max *level*

Display Contexts

To display configured context names, enter the following command in operator exec configuration mode:

show context [*ctx-name* | **all**]

If you do not specify any optional arguments, the current context name is displayed.

Display IP Hosts

To display information about statically configured IP hosts in the current context, enter the following command in operator exec configuration mode:

show ip host [*ip-address*]

If you do not specify any optional arguments, all IP host table entries are displayed.

Display IP Traffic

To display IP traffic information for the current context, enter the following command in operator exec configuration mode:

show ip traffic

Clear IP Counters

To clear the IP traffic statistics associated with the **show ip traffic** command in the current context, enter the following command in administrator exec configuration mode:

clear ip counter

Restrict System Access

To apply an access control list to a context, restricting administrative access to the system, enter the following configuration command:

ip access-group *name* {**in** | **out**}

Configuration Examples

The first example shows the creation of an administrator account with the `super` username, and the `icandoanything` password. When the administrator logs on to the system, the initial privilege level is `6`. The administrator can modify the privilege level up to the maximum of `15`. Because this account is created in the local context, this administrator is able to view and modify the entire system configuration, and view all running information on the system.

```
[local]RedBack#configure
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#administrator super password icandoanything
[local]RedBack(config-admin)#privilege start 6
[local]RedBack(config-admin)#privilege max 15
[local]RedBack(config-admin)#exit
```

The second example shows a sample configuration for a context named `isp.net`. The example configures two domain name aliases. With this configuration, operators, administrators, and subscribers can log on to this context using the `isp` and `ispx` domain names, and the context name `isp.net`. The example configures an operator account named `isp-operator`. Using this logon account, an operator can have a 10-minute session and is able to view system information only in the `isp.net` context. The example also configures an administrator account named `isp-admin`. This administrator has access to additional information in the `isp.net` context. If the administrator logs on to the system using Telnet, the session is ended after 10 minutes of idle time.

```
[local]RedBack(config-ctx)#context isp.net
[local]RedBack(config-ctx)#domain isp advertise
[local]RedBack(config-ctx)#domain ispx advertise
[local]RedBack(config-ctx)#operator isp-operator password oper!secret
[local]RedBack(config-admin)#timeout absolute 10
[local]RedBack(config-admin)#exit
[local]RedBack(config-ctx)#administrator isp-admin password admin!secret
[local]RedBack(config-admin)#timeout idle 10
[local]RedBack(config-admin)#end
```

The following example shows sample output for the **show context** command given the configuration in the previous examples:

```
[local]RedBack#show context all

local(0)
isp.net (1)
```

The following example shows sample output for the **show ip traffic** command. The output displays IP statistics for the local context, because that is the context in which the command is executed. The example then clears the IP counters for the local context.

```
[local]RedBack#show ip traffic

IP statistics:
   Rcvd:   712605 total, 662371 local destination
           0 format errors, 0 checksum errors
   Frags:  0 reassembled, 0 timeouts
           0 couldn't reassemble
           0 fragmented, 0 couldn't fragment
   Sent:   567659 generated, 0 forwarded
           0 no route 0 other errors

ICMP statistics:
   Rcvd:   0 format errors, 0 checksum errors
           0 unreachables, 36 echo, 0 echo replies
           2 mask requests 0 mask replies, 0 quench
           0 parameter, 0 timestamp, 0 info request
           0 redirects
   Sent:   0 redirects, 0 unreachables, 0 echo
           36 echo replies, 0 timestamp, 0 info reply
           0 quench, 0 mask requests, 0 mask replies
           0 time exceeded, 0 parameter problem

ARP statistics:
   Rcvd:   50292 requests, 343 replies, 0 other
   Sent:   598 requests, 0 replies, 0 bad
   InvArp: 0 req-rcvd, 0 rep-sent

UDP statistics:
   Rcvd:   82457 total, 0 bad format
           0 checksum errors, 82452 no port
           0 full socket 0 pcb lookup failure
     Sent: 0 total

TCP statistics:
   Rcvd:  570809 total, 0 bad format
          24 checksum errors 104 no port
   Sent:  9791 total 8994 data packet
          24 control packet 102 retransmitted
```

```
IGMP statistics:
   Rcvd:   0 format errors 0 too short 0 checksum
           0 bad queries 0 bad reports 0 no router alert
           0 queries 0 reports 0 leaves 0 total
   Sent:   0 queries 0 reports 0 leaves

[local]RedBack#clear ip counters
[local]RedBack#
```

Configuring Interfaces

This chapter provides an overview of interfaces and describes the basic tasks involved in configuring interfaces through the Access Operating System (AOS). Tasks include configuring, displaying, and debugging IP addresses, IP address pools, and Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), and maximum transmission unit (MTU) parameters.

For protocol-specific, or feature-specific, interface configuration mode commands, see the appropriate chapter in this guide. For example, to enable interfaces to originate Internet Group Management Protocol (IGMP) queries and use IGMP responses from hosts, see Chapter 36, "Configuring IGMP Proxy."

For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "Interface Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

Within the AOS, an interface is a logical entity that provides higher-layer protocol and service information, such as Layer 3 addressing. Interfaces are configured as part of a context and are independent of physical ports and circuits. The separation of the interface from the physical layer allows for many of the advanced features offered by the AOS. For higher-layer protocols to become active, an interface must be bound to a physical port or circuit.

Configuration Tasks

To configure interfaces, perform the tasks in the following sections:

- Configure an Interface
- Assign a Primary and, Optionally, Secondary IP Addresses
- Assign a Range of Addresses from an IP Pool
- Provide a Description of the Interface
- Bind the Interface to a Circuit
- Configure Interface Parameters
- Display Interface Information
- Enable the Logging of Debug Messages

Configure an Interface

Interface names can be any arbitrary string. For ease of identification, the slot and port is typically used as part of the name.

To configure an interface, use the following command in context configuration mode. This command also places you in interface configuration mode:

interface *if-name* [**loopback** | **ppp default**]

A loopback interface is an interface that has no association with any circuit in the system. This is useful in applications that require an IP address, but not a physical connection. For instance, a loopback interface can be useful for routing protocols, because the interface is not associated with a physical port that can go down. Up to 16 loopback interfaces can be configured per context.

A Point-to-Point Protocol (PPP) default interface is an interface that acts as a fall back for incoming PPP connections. Ordinarily, PPP sessions that attempt to come up and cannot bind to a valid interface simply fail. A PPP default interface acts as a fall back for those incoming PPP connections. If a PPP session is established, and there is no valid interface to which it can bind, the session binds to the default interface. The default interface is a virtual interface; there is no actual outgoing circuit. Therefore, a proxy is necessary.

One or more interfaces that are not the default interface are set up as proxies using the **ip ppp-proxy-arp** command in interface configuration mode. The outgoing circuits from these proxies can then be used to handle the traffic on the virtual default interface.

Once an interface has been configured, it requires a primary IP address. In addition, all nonloopback interfaces must be bound to a circuit.

Assign a Primary and, Optionally, Secondary IP Addresses

You can statically assign a primary IP address to an interface using the **ip address** command as described here, or by applying a pool of IP addresses to an interface as described in the next section, "Assign a Range of Addresses from an IP Pool."

To statically configure a primary IP address and, optionally, one or more secondary IP addresses, enter the following command in interface configuration mode:

ip address *ip-address* [*netmask*] [**secondary**]

You cannot configure a netmask for a PPP default interface. Use the **secondary** keyword to designate an IP address as a secondary address to an interface. Up to 15 secondary addresses can be configured per-primary interface.

You cannot configure secondary IP addresses for loopback or PPP default interfaces.

To assign an IP address to a subscriber, use the **ip address** command in subscriber configuration mode. See Chapter 8, "Configuring Subscribers."

Assign a Range of Addresses from an IP Pool

To assign a range of IP addresses from a locally defined pool to an interface, enter the following command in interface configuration mode:

ip pool *ip-address netmask*

For the **ip pool** command to take effect, a Remote Authentication Dial-In User Service (RADIUS) server must be configured to return the Framed-IP-Address attribute with a value of 255.255.255.254. This RADIUS attribute informs the Subscriber Management System (SMS) device that the interface's IP address is assigned from a pool.

Note This command does not apply to loopback interfaces.

Provide a Description of the Interface

To associate descriptive information with an interface, enter the following command in interface configuration mode:

description *text*

The description appears in the output of the **show interface** and **show configuration** commands.

Bind the Interface to a Circuit

To bind an interface to a circuit, enter the following command:

bind interface *if-name ctx-name*

Specify the interface name and the context in which the interface resides.

This command is available in several modes:

- To bind an interface to an Ethernet or Cisco High-level Data Link Control (HDLC)-encapsulated port, use the **bind interface** command in port mode. (These port types are treated as a single circuit.)
- To bind an interface to a channel on a Cisco HDLC-encapsulated channelized DS-3 port, use the **bind interface** command in HDLC channel configuration mode. (The channel is treated as a single circuit.)
- To bind an interface to a virtual circuit, use the **bind interface** command in circuit configuration mode.

This command does not apply to loopback interfaces.

For detailed information on bindings, see Chapter 20, "Configuring Bindings."

Configure Interface Parameters

You can configure a variety of interface parameters, including enabling the ARP, the IGMP, the Routing Information Protocol (RIP), and so on.

Note All of the commands described in the following sections are found in interface configuration mode.

Enable ARP

By default, ARP is disabled on all interfaces.

To indicate that the address resolution type to be used on an interface is standard Ethernet ARP, enter the following command:

ip arp arpa

This command does not apply to loopback interfaces or to PPP default interfaces.

Enable Secured ARP

Secured-ARP enables the SMS device to resolve only those Media Access Control (MAC) addresses that correspond to configured subscriber IP addresses.

To enable secured-ARP on an interface, enter the following command:

ip secured-arp

The **ip arp arpa** command must be enabled on an interface before any ARP processing for that interface can take place.

This command does not apply to loopback interfaces or to PPP default interfaces.

Modify the Amount of Time an ARP Entry Remains in the System Cache

To set the number of seconds an idle ARP cache entry remains in the system's cache, enter the following command:

ip arp timeout

This command does not apply to loopback interfaces or to PPP default interfaces.

Enable ICMP Mask Replies

To enable an interface to send ICMP mask replies on receipt of an ICMP mask request, enter the following command:

ip mask-reply

Allow IP Packet Fragmentation for Forwarding

By default, when fragmentation is required to forward an IP packet and the packet's "don't fragment" bit is set, the outgoing interface discards the packet.

To allow a forwarded IP packet to be fragmented when its length exceeds the MTU size associated with the outgoing interface, regardless of the packet's "don't fragment" setting, enter the following command:

ip ignore-df-bit

Modify the MTU Size

To modify the MTU size for IP packets sent on an interface, enter the following command:

ip mtu *bytes*

If an IP packet exceeds the MTU size, the system fragments that packet.

Configure the Next-Hop Lookup Method Used in Routing

The SMS device keeps two forwarding tables: the host table and the routing table. By default, when a packet is received by an interface, the SMS device selects the next-hop interface by first examining the routing table and then by looking at the host table.

To set the SMS device to look at the host table first when selecting the next-hop interface, enter the following command:

ip lookup host

Configure the Interface IP Address as the Source for SNMP and RADIUS Packets

To configures the interface's primary IP address as the source address for all Simple Network Management Protocol (SNMP) trap packets and RADIUS packets that are sent from the context, enter the following command:

ip source-address {**snmp** [**radius**] | **radius** [**snmp**]}

Note This command is also described in Chapter 45, "Configuring SNMP and RMON" and Chapter 41, "Configuring RADIUS."

Display Interface Information

You can display information about the IP ARP table, interfaces, and IP address pools.

Note The commands described in the following sections are found in operator exec mode.

Show IP Interface

To display information about IP interfaces configured in the current context, enter the following command:

show ip interface [**brief** | *if-name* [**access-statistics**]]

You can use this command to display information (detailed or brief) about all interfaces or information specific to a named interface, including a list of all circuits or ports currently bound to the interface and their status. The **access-statistics** keyword displays the number of inbound and outbound packets filtered by any access control list configured for the named interface.

Show IP Pool

To display all IP address pools for the current context, enter the following command:

show ip pool

A list of IP addresses from pools assigned to interfaces are displayed, as are the number of addresses in use, available, or unusable. Unusable addresses include those used by an interface or the interface's all ones or all zeros address.

Show IP ARP

To display the IP ARP table for the current context, enter the following command:

show ip arp [*ip-address*]

This command displays host address, next-hop count, MAC address, address resolution status, and time-to-live value information.

Show IP Secured ARP

To display IP hosts residing on network segments associated with interfaces in the current context for which secured ARP is enabled, enter the following command:

show ip secured-arp [*ip-address*]

Enable the Logging of Debug Messages

IP Interfaces

To enable the logging of debug messages for all interfaces, enter the following command in administrator exec mode:

debug ip interface

IP ARP

To enable the logging of IP ARP debug messages, enter the following command in administrator exec mode:

debug ip arp

IP Secured-ARP

To enable the logging of IP secured ARP debug messages, enter the following command in administrator exec mode:

debug ip secured-arp

Configuration Examples

The following commands configure two interfaces and enable ARP:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#interface enet20
[local]RedBack(config-if)#ip address 10.1.2.1 255.255.255.0
[local]RedBack(config-if)#ip arp arpa
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#interface enet21
[local]RedBack(config-if)#ip address 10.1.1.1 255.255.255.0
[local]RedBack(config-if)#ip arp arpa
```

Two noncontiguous Classless InterDomain Routing (CIDR) blocks are configured for the interface `Downstream`:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#interface Downstream
[local]RedBack(config-if)#ip address 10.0.0.1 255.255.255.0
[local]RedBack(config-if)#ip address 11.0.0.1 255.255.255.0 secondary
```

The following commands enable subscriber `fred`'s circuit to bind to the interface `Downstream` using either IP address:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#subscriber name fred
[local]RedBack(config-sub)#ip address 10.0.0.2 255.255.255.240
[local]RedBack(config-sub)#ip address 11.0.0.2 255.255.255.240
...
[local]RedBack(config)#port atm 3/0
[local]RedBack(config-port)#atm pvc 0 1 profile UBR encapsulation bridge1483
[local]RedBack(config-pvc)#bind subscriber fred@local
```

Configuring Subscribers

This chapter provides an overview of subscribers and describes the basic tasks involved in configuring, maintaining, and troubleshooting subscribers and subscriber sessions through the Access Operating System (AOS).

Note For protocol-specific, or feature-specific, subscriber configuration mode commands, see the appropriate chapter in this guide. For example, to enable subscribers to transmit or receive IP multicast traffic, see Chapter 36, "Configuring IGMP Proxy."

For detailed information on syntax and usage guidelines for the commands listed in the the "Configuration Tasks" section, see the "Subscriber Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

Subscribers are end users of high-speed access services. Subscriber records are used to define a set of attributes, such as username, password, authentication, access control, rate-limiting, and policing information. A record is specific to the context in which the subscriber is configured.

You can configure a default subscriber record to define attributes that are applied to all subscribers in the context. With a default subscriber record, you can configure attributes that are shared by many subscribers in a single configuration, rather than applying the same attributes separately to each subscriber record. Default attributes are overridden when identical attributes with different values are configured in a specific subscriber record.

Subscribers use hosts connected to circuits that are configured for RFC 1483 bridged or RFC 1490 bridged encapsulation, or are configured for Point-to-Point Protocol (PPP) or PPP over Ethernet (PPPoE) connections.

Subscriber records can be configured in two ways:

- Locally, through the Access Operating System (AOS) command-line interface (CLI) commands.

  You can use subscriber records to provide local authentication and authorization information whenever a remote authentication and authorization server, such as Remote Authentication Dial-In User Service (RADIUS), is not available nor desired.

- Via attributes stored on a RADIUS server that the Subscriber Management System (SMS) device is configured to access.

  If the RADIUS server is configured within the local context of the AOS, authentication and/or accounting is applied globally to all subscribers. If the RADIUS server is configured within any other context, authentication and/or accounting applies only subscribers configured in that particular context.

  See Chapter 41, "Configuring RADIUS," for details on how to configure RADIUS servers, authentication, authorization, and accounting (AAA), circuit creation through AAA, and more. See Appendix C, "RADIUS Attributes," for a description of all standard and Redback vendor-specific attributes supported by the AOS.

  For detailed information on the commands described in this chapter, see Chapter 8, "Configuring Subscribers," in the *Access Operating System (AOS) Command Reference* publication.

Configuration Tasks

To configure subscribers, perform the tasks in the following sections:

- Configure Subscribers
- Configure Authentication
- Configure Session Timeout for Subscribers
- Assign IP Addresses
- Enable IP Source Address Validation
- Create an Entry in the ARP Cache
- Set the Type of Service Bit
- Limit the Number of Concurrent Sessions Allowed
- Modify the Traffic Rate and Burst Tolerance
- Display Subscriber Information
- Clear Subscribers
- Clear Host Addresses from the ARP Cache
- Enable IP ARP Debugging Messages

Configure Subscribers

To configure an individual or default subscriber record, enter the following command in context configuration mode. This command also places you in subscriber configuration mode.

subscriber {**default** | **name** *sub-name*}

Use the **name** keyword and specify the subscriber name to create an individual subscriber record. Use the **default** keyword to configure a default subscriber record.

Attributes configured for the default subscriber record apply to all subscribers within the context. With a default subscriber record, you can configure attributes that are shared by many subscribers in a single configuration, rather than applying the same attributes separately to each subscriber record. Default attributes are overridden when identical attributes with different values are configured in a specific subscriber record.

Note The subscriber configuration mode commands, **ip arp** and **password**, are available for individual subscriber records, but *not* for a default subscriber record.

Configure Authentication

To configure the authentication password that the subscriber enters when initiating a Point-to-Point Protocol (PPP) session, enter the following command in subscriber configuration mode:

password *password*

To configure the password supplied by the AOS to the subscriber's host to authenticate the subscriber for a PPP session, enter the following command in subscriber configuration mode:

outbound password *password*

Configure Session Timeout for Subscribers

To set an idle or absolute Point-to-Point Protocol (PPP) or PPP over Ethernet (PPPoE) timeout for a subscriber, enter the following command in subscriber configuration mode:

timeout {**absolute** | **idle**} *minutes*

Note Keepalive messages are considered traffic for purposes of measuring idle time.

Assign IP Addresses

An IP address is required for each subscriber.

To assign an IP address to a subscriber, enter the following command in subscriber configuration mode:

ip address {*address* [*network-mask*] | **pool** [**name** *if-name*]}

To specify a range of contiguous IP addresses, use the optional *network-mask* argument. Use the **pool** keyword to assign an address from a pool of IP addresses. See Chapter 7, "Configuring Interfaces," for information on how to configure a pool via the **ip pool** command in interface configuration mode.

You can specify either an IP address or an IP pool for a subscriber record, but not both. Use the optional **name** *interface-name* construct if you want the IP address to be assigned from a pool configured specifically for that interface.

There must be an interface whose IP address and netmask range includes the IP address assigned to each subscriber configured. To configure an interface, use the **interface** command in interface configuration mode. To assign an IP address to an interface, use the **ip address** command in interface configuration mode; See Chapter 7, "Configuring Interfaces."

A subscriber record is read-only when the subscriber is bound to a circuit. See Chapter 20, "Configuring Bindings," for information on static and dynamic binding options.

Enable IP Source Address Validation

Use IP source-address validation to prevent address spoofing. To deny all IP packets from address sources that are not reachable through a subscriber's associated circuit, enter the following command in subscriber configuration mode:

ip source-validation

Without the **unsolicit** keyword, the sending of unsolicited traffic is set to **deny** by default. Use the **unsolicit** keyword to permit the subscriber to send unsolicited multicast traffic. By default, subscribers can join an unlimited number of multicast groups.

Create an Entry in the ARP Cache

If a subscriber's host is not capable of (or not configured to) responding to ARP requests, manually create an entry in the ARP cache by entering the following command in subscriber configuration mode:

ip arp *ip-address mac-address*

For example, hosts connected to RFC 1483 bridged or RFC 1490 bridged encapsulated circuits are not capable of responding to ARP requests. This command is available for individual subscriber records, but *not* for a default subscriber record.

Set the Type of Service Bit

The type of service (ToS) bit may already be set in the headers of incoming IP packets. To reset the ToS bit, enter the following command in interface configuration mode:

ip tos-field {**normal** | **min-cost** | **max-reliability** | **max-throughput** | **min-delay** | **raw** *value*}

Limit the Number of Concurrent Sessions Allowed

To limit the number of sessions a subscriber can access simultaneously, enter the following command in subscriber configuration mode:

port-limit *max-sessions*

Modify the Traffic Rate and Burst Tolerance

You can limit the aggregate packet stream received or sent a subscriber's circuit by rate and burst tolerance. To limit the stream received, enter the following command in subscriber configuration mode:

police *rate* **burst** *size*

To limit the stream sent, enter the following command in subscriber configuration mode:

rate-limit rate *rate* **burst** *size*

Packets exceeding the specified rate and burst tolerance are dropped.

Display Subscriber Information

You can display a variety of subscriber information about inbound and outbound packets, and also information about the IP ARP table for the current context.

Show Subscribers

To display subscriber information, enter the following command in operator exec mode:

show subscribers [**access-statistics** [*sub-name*] | **active** [*sub-name*] | **address** *sub-name* | [**summary**] [**all**]]

Show IP ARP

To display the IP ARP table for the current context, enter the following command in operator exec mode:

show ip arp

Host address, next-hop count, Media Access Control (MAC) address, address resolution status, and time-to-live value information is displayed.

Clear Subscribers

To clear a subscriber, thus terminating any PPP or PPPoE session or dropping any RFC 1483 bridged or RFC 1490 bridged encapsulated circuit connection, enter the following command in operator exec configuration mode:

clear subscriber *sub-name*

You can also use this command to modify a subscriber record for a subscriber that is already bound for the changes to take effect. The subscriber session is terminated and restarted with the new parameters.

Clear Host Addresses from the ARP Cache

To clear one or all host addresses from the dynamic ARP cache in the current context, enter the following command in operator exec mode:

clear arp-cache [*host-address*]

If the optional *host-address* argument is not specified, all entries are cleared; otherwise, only the host with the matching IP address is cleared.

Enable IP ARP Debugging Messages

To enable the logging of IP ARP debugging messages, enter the following command in administrator exec mode:

debug ip arp

Configuration Examples

The following example configures an inbound password, an outbound password, and an IP address in the subscriber record named `pppuser` in the `local` context:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#subscriber name pppuser
[local]RedBack(config-sub)#password in-test
[local]RedBack(config-sub)#outbound password out-test
[local]RedBack(config-sub)#ip address 10.1.3.30
```

The following commands create two subscriber records, `sub1` and `sub2`. The circuit associated with `sub1` will be implicitly bound to the `downstream1` interface. The circuit associated with `sub2` may be implicitly bound to either the `downstream1` or `downstream2` interface and is expected to consume four IP host addresses. The interface chosen will be whichever interface has remaining capacity from its pool of 200 expected dynamic addresses.

```
[local]RedBack(config-ctx)#subscriber name sub1
[local]RedBack(config-sub)#ip address 10.1.1.1
[local]RedBack(config-sub)#subscriber name sub2
[local]RedBack(config-sub)#dhcp max-addr 4
```

For subscriber `joe@local`, the following example would cause a PPPoE Active Discovery Message packet (PADM) containing the URL http://www.cust1.com/members/joe@local to be sent to the PPPoE client when the PPP session is established:

```
[local]RedBack(config-ctx)#subscriber name joe
[local]RedBack(config-sub)#pppoe url http://www.cust1.com/members/%U
```

The next example uses the **pppoe url** command to configure the subscriber default. For every subscriber to which the subscriber default is applied, a PADM containing http://www.aol.com/members/*name* is sent to the PPPoE client when the PPP session is established:

```
[local]RedBack(config-ctx)#subscriber default
[local]RedBack(config-sub)#pppoe url http://www.aol.com/members/%u
```

The following example configures a subscriber named `roger` in the `corp.com` context to have a maximum PPP session time of `120` minutes (2 hours):

```
[local]RedBack(config)#context corp.com
[local]RedBack(config-ctx)#subscriber name roger
[local]RedBack(config-admin)#timeout absolute 120
```

The following example configures the system to supply a primary Domain Name System (DNS) address to every PPP subscriber in the current context. See RFC 1877, *PPP Internet Protocol Control Protocol Extensions for Name Server Addresses*.

```
[local]RedBack(config-ctx)#subscriber default
[local]RedBack(config-sub)#dns primary 10.10.1.1
```

The following example creates a message of the minute (MOTM):

```
[local]RedBack(config-sub)#pppoe motm System coming down at 0400 today for scheduled
maintenance
```

The following example replaces the first MOTM with a new one:

```
[local]RedBack(config-sub)#pppoe motm Scheduled system maintenance canceled for 08/29.
```

The following example removes the existing MOTM so that no message is sent to subscribers:

```
[local]Redback(config-sub)#no pppoe motm
```

Part 3

Ports, Circuits, Channels, and Bindings

Configuring Common Port, Circuit, and Channel Parameters

This chapter provides an overview of ports, circuits, and channels and describes the tasks involved in configuring common port, circuit, and channel parameters through the Access Operating System (AOS). For detailed information on the syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "Common Port, Circuit, and Channel Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

For information on configuring parameters that are specific to a particular port, circuit, or channel type, see the appropriate chapter in this manual.

Overview

Ports are the physical interfaces on the Subscriber Management System (SMS) I/O modules. The SMS supports many different types of I/O modules. Configuration tasks vary for the ports, depending on the type of I/O module. This chapter covers common configuration tasks that apply to all SMS port types.

The following types of ports transfer data over virtual circuits:

- Asynchronous Transfer Mode (ATM) ports—See Chapter 17, "Configuring ATM," for overview information and configuration tasks related to ATM virtual circuits.
- Frame Relay Ports—See Chapter 18, "Configuring Frame Relay," for overview information and configuration tasks related to Frame Relay virtual circuits.

On channelized DS-3 ports, you can configure T1 and High-Level Data Link Control (HDLC) channels. For overview information and configuration tasks related to T1 and HDLC channels, see Chapter 12, "Configuring Channelized DS-3 Ports."

Configuration Tasks

AOS provides default values for all port, circuit, and channel parameters. The following are optional tasks that you can perform to customize your configuration:

- Provide a Description
- Configure Buffers
- Configure Bulk Statistics Schemas
- Modify Police Rate Parameters
- Modify Rate-Limit Parameters
- Enable HDLC Debugging Messages
- Display Port Information
- Clear Sessions on Circuits

Provide a Description

You can provide a textual description for any port, circuit, or channel. This allows you to associate additional information with the port, circuit, or channel. To configure a description, enter the following command in port, circuit, or channel configuration mode:

description *text*

Configure Buffers

You can configure AOS to limit the total number of packet buffers that can be consumed by a port, both on the transmit side and the receive side.

To modify the number of transmit or receive packet buffers, enter the following command in port configuration mode:

buffers {**transmit** *value1* | **receive** *value2*}

Note This command should be used with caution. Improperly setting this value can severely impact overall system performance. Consult with your technical support representative before you modify the default settings.

Note This command does not apply to Gigabit Ethernet ports.

Configure Bulk Statistics Schemas

To define the statistics schema for the contents of the bulkstats collection file for a port, enter the following command in port configuration mode:

bulkstats schema *name* **format** *format-string* [*AOS-variable* [*AOS-variable*...]]

See Chapter 43, “Configuring Bulk Statistics,” for additional information on configuring bulk statistics.

Modify Police Rate Parameters

You can limit the aggregate packet stream received from a port to a certain rate (in kilobits per second) and burst tolerance (in bytes). A reasonable rule-of-thumb for burst tolerance is ten times the link MTU, or around 15,000 to 20,000 bytes for subscriber circuits. A larger burst tolerance is generally appropriate for backhaul circuits. Packets exceeding the specified rate and tolerance parameters are dropped.

By default, policing is disabled. To enable policing on a port, and configure the police rate parameters, enter the following command in port configuration mode:

police rate *rate* **burst** *size*

Modify Rate-Limit Parameters

You can limit the aggregate packet stream transmitted on a port to the specified rate and burst tolerance. A reasonable rule-of-thumb for burst tolerance is ten times the link MTU, or around 15,000 to 20,000 bytes for subscriber circuits. A larger burst tolerance is generally appropriate for backhaul circuits. Packets exceeding the specified rate and tolerance are dropped.

By default, rate-limiting is disabled. To enable rate-limiting on a port, and configure the rate-limit parameters, enter the following command in port configuration mode:

rate-limit rate *rate* **burst** *size*

Enable HDLC Debugging Messages

You can configure the system to display debugging messages related to High-Level Data Link Control (HDLC). These debugging messages apply only to ports, circuits, or channels configured for Cisco HDLC encapsulation. To enable HDLC debugging, enter the following command in administrator exec mode:

debug hdlc [*slot/port* [{**all** | **hdlc-channel** *name*}]]

Display Port Information

Administrators and operators can display various types of information about ports. This section describes how to:

- Display System Ports
- Display Port Hardware Configuration
- Display Port Configuration Information
- Display Port Statistics
- Clear Port Statistics

Display System Ports

To display a table of the ports in your system, enter the following command in operator exec mode:

show port table

This command displays the following information for all ports in the system:

- slot and port number
- port type
- port state
- driver type

The following example shows sample output from the **show port table** command on a system that is configured with the following I/O modules:

- Two-port Ethernet
- Two-port ATM DS-3
- Two-port ATM OC-3c
- Two-port clear-channel DS-3
- Eight-port packet T1

```
[local]RedBack>show port table

I/O Port Table contents are:

   Port 0/0 is UP driver type is ENET port type is 100BT
   Port 2/0 is SHUTDOWN driver type is ENET port type is 100BT
   Port 2/1 is SHUTDOWN driver type is ENET port type is 100BT
   Port 3/0 is SHUTDOWN driver type is ATM port type is DS3
   Port 3/1 is SHUTDOWN driver type is ATM port type is DS3
   Port 4/0 is SHUTDOWN driver type is FRAME port type is DS3
   Port 4/1 is SHUTDOWN driver type is FRAME port type is DS3
   Port 5/0 is SHUTDOWN driver type is ATM port type is OC3
   Port 5/1 is SHUTDOWN driver type is ATM port type is OC3
   Port 6/0 is SHUTDOWN/UNCONFIGURED driver type is FRAME port type is CT3
   Port 6/1 is SHUTDOWN/UNCONFIGURED driver type is FRAME port type is CT3
   Port 7/0 is SHUTDOWN/UNCONFIGURED driver type is FRAME port type is DS1
   Port 7/1 is SHUTDOWN/UNCONFIGURED driver type is FRAME port type is DS1
   Port 7/2 is SHUTDOWN/UNCONFIGURED driver type is FRAME port type is DS1
   Port 7/3 is SHUTDOWN/UNCONFIGURED driver type is FRAME port type is DS1
   Port 7/4 is SHUTDOWN/UNCONFIGURED driver type is FRAME port type is DS1
   Port 7/5 is SHUTDOWN/UNCONFIGURED driver type is FRAME port type is DS1
   Port 7/6 is SHUTDOWN/UNCONFIGURED driver type is FRAME port type is DS1
   Port 7/7 is SHUTDOWN/UNCONFIGURED driver type is FRAME port type is DS1
```

Display Port Hardware Configuration

To display hardware configuration information a port, enter the following command in operator exec mode:

show port diagnostics *slot*/*port*

The information in the display varies depending on the port type. The following example shows sample diagnostics information for an ATM DS-3 port:

```
[local]RedBack>show port diag 3/1

Slot/Port number 3/1
        STATE_PRESENT
        Description "Brooktree 8233"
        Vendor ID = 0x109e
        Device ID = 0x8233
        Sub Vendor ID = 0x0000
        Sub System ID = 0x0000
        Class = 02 Network Controller
        Sub Class = 0x03 ATM
        Base 0 = 0x62000000     size = 0x01000000
        Interrupt line = 0x0b
        Command = 0x0346
        Status = 0x0080
        Lat Timer = 0x10
        Special Status = 0x01
        Max Burst Len = 0x0d
        Curr Mstr Rd Addr = 0xff77fff0
        Curr Mstr Wr Addr = 0xbf7ffffc
```

Display Port Configuration Information

To display configuration information for a port, enter the following command in operator exec mode:

show port info [*slot*/*port*]

The following example shows sample output for an ATM DS-3 port:

```
[local]RedBack>show port info 3/1

Port 3/1,  state is DOWN, driver type is ATM
Description                 = To DSLAM in Rack 5, shelf 4
MAC Address                 = 00:10:67:00:22:be
Rate limit rate             = Disabled
Rate limit burst            = Disabled
Police rate                 = 100000
Police burst                = 9984
Physical layer interface    = DS3
Loopback                    = none
Cell-delineation            = hcs
Payload scrambling          = enabled
Clock-source                = internal
Idle cell header            = 0x00000000
Idle cell data              = 0x5a
Cable length                = short ( <= 225 ft )
External 8KHz Timing        = disabled
Transmit Buffers            = 256
Receive Buffers             = 64
Circuit Creation Mode       = Explicit
```

To display the contents of the system configuration that relate to a particular port, enter the following command in administrator exec mode:

show configuration port *slot*/*port* **verbose**

The following example shows sample output for the same port as in the previous example. The **verbose** option includes defaulted configuration values in the output:

```
[local]RedBack#show configuration port 3/1 verbose
port atm 3/1
 description To DSLAM in Rack 5, shelf 4
 no buffers transmit
 no buffers receive
 no 8khztiming
 no stuffing
 clock-source internal
 default scramble
 no loopback
 circuit creation explicit
 police rate 100000 burst 9984
```

Display Port Statistics

To display statistics for a port, enter the following command in operator exec mode:

show port counters *slot/port*

This command displays general counters as well as counters that are specific to the port type. For additional information including descriptions of the counters, see the "Common Port, Circuit, and Channel Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

Clear Port Statistics

To clear statistics for a port, enter the following command in administrator exec mode:

clear port counters *slot/port* [**hdlc-channel** *chan-name*] [**pvc** {**all** | *vpi* [*vci* [**through** *end-vci*]] | *dlci* [**through** *end-dlci*]} [**dot1q-pvc** {**all** | *vlan-id* | **untagged**}]] [**-noconfirm**]

Clear Sessions on Circuits

To tear down active subscriber sessions on a particular circuit or circuits, enter the following command in operator exec mode:

clear circuit {*slot/port* {*vpi vci* [**through** *end-vci*] | [*hdlc-channel*] *dlci* [**through** *end-dlci*] | **all**} | **pppoe** {[*cm-index*-]*session-id* [**through** *end-session-id*] | **all**}}

Configuration Examples

The following example shows the configuration of the common port parameters for a clear-channel DS-3 port:

```
[local]RedBack(config)#port ds3 5/0
[local]RedBack(config-port)#description To DSLAM Rack 1, shelf 3
[local]RedBack(config)#buffers receive 100
[local]RedBack(config)#buffers transmit 100
[local]RedBack(config)#bulkstats schema sample format “global: %u, %u, %u, host: %s”,
sysuptime date timeofday hostname
[local]RedBack(config)#police-rate 100000 burst 20000
[local]RedBack(config)#rate-limit 50000 burst 10000
```

Configuring Ethernet Ports

This chapter provides an overview of Ethernet ports and describes the tasks involved in configuring Ethernet ports through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "Ethernet Port Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

The information in this chapter applies to all Ethernet ports on the system, including the Ethernet management port. For additional information on configuring the management port, see the "Create Administrators and Operators" section in Chapter 3, "Accessing the AOS."

Configuration Tasks

Configuring Ethernet ports typically consists of the following tasks:

- Define Ethernet Port Characteristics
- Configure Common Port Parameters
- Set the Encapsulation Type
- Enable the Port

To enter port configuration mode to begin to configure an Ethernet port, enter the following command in global configuration mode:

port ethernet *slot/port*

Define Ethernet Port Characteristics

Configuring the Ethernet port characteristics consists of the following optional tasks:

- Set the Port Speed and Duplex Mode
- Configure RADIUS Attributes
- Configure a Static Host Entry
- Configure a Loopback

Set the Port Speed and Duplex Mode

By default, all Ethernet ports are configured to auto-sense both the port speed and the duplex mode. This is the recommended configuration.

You can also set an Ethernet port to use a specified speed and duplex mode. To do so, enter the following command in port configuration mode:

medium speed {**10** | **100** | **1000**} **duplex** {**half** | **full**}

Note The port does not come up if the medium speed or the duplex mode is configured incorrectly.

To configure the port back to the default (auto-sense), use one of the following commands in port configuration mode:

medium auto

default medium

Configure RADIUS Attributes

To configure the value of the Medium-Type Remote Access Dial-In User Service (RADIUS) attribute for any Point-to-Point Protocol (PPP) over Ethernet (PPPoE) sessions that arrive at the Subscriber Management System (SMS) device over the port, enter the following command in port configuration mode:

radius attribute medium-type {**cable** | **dsl** | **satellite** | **wireless**}

If you do not configure this parameter, the attribute is not sent.

See Chapter 41, “Configuring RADIUS,” for overview information and configuration tasks related to RADIUS.

Configure a Static Host Entry

You can create a static entry in the system host table if dynamic address resolution through the Address Resolution Protocol (ARP) is not possible or not wanted. To do so, enter the following command in port configuration mode:

ip host *ip-address mac-address*

Configure a Loopback

You can create a loopback on an Ethernet port to test the port. To do so, enter the following command in port configuration mode:

loopback

Configure Common Port Parameters

Ethernet ports support many of the common port, circuit, and channel parameters supported by the AOS; see Chapter 9, "Configuring Common Port, Circuit, and Channel Parameters," for information on configuring common port parameters.

Set the Encapsulation Type

Ethernet ports use IP over Ethernet as the default encapsulation. You can configure the port to specify 802.1Q encapsulation, PPPoE encapsulation, or a combination of PPPoE and IP over Ethernet. To configure the encapsulation type, enter the following command in port configuration mode:

encapsulation {**dot1q** | **ppp over-ethernet** | **multi**}

Note You cannot enter this command for the Ethernet management port; the management port only supports IP over Ethernet encapsulation.

When you select 802.1Q encapsulation using the **dot1q** keyword, you enter dot1q encapsulation configuration mode. For additional information on configuring 802.1Q encapsulation, see Chapter 19, "Configuring 802.1Q."

Enable the Port

By default, all ports on the system are configured to be shut down. To enable a port, enter the following command in port configuration mode:

no shutdown

Configuration Examples

The following example shows a complete configuration for an Ethernet port, including common port parameters:

```
[local]RedBack(config)#port ethernet 2/0
[local]RedBack(config-port)#description To DSLAM Rack 1, shelf 3
[local]RedBack(config)#buffers receive 350
[local]RedBack(config)#buffers transmit 350
[local]RedBack(config)#medium auto
[local]RedBack(config)#encapsulation multi
[local]RedBack(config)#no shutdown
```

Configuring ATM Ports

This chapter provides an overview of Asynchronous Transfer Mode (ATM) ports and describes the tasks to configure ATM ports through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "ATM Port Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter contains the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

This chapter describes how to configure ATM ports. For data to flow over an ATM port, you must also configure virtual circuits. After you have completed configuring ATM ports, see Chapter 17, "Configuring ATM," for information on configuring virtual circuits and other ATM software features.

Configuration Tasks

The configuration tasks related to configuring ATM ports depend on the hardware type of the port. The following sections describe the configuration tasks for various ATM port types:

- Configure ATM OC-3 and OC-12 Ports
- Configure ATM DS-3 and ATM E3 Ports
- Configure ATM T1 and ATM E1 Ports

To enter port configuration mode to configure an ATM port, enter the following command in global configuration mode:

port atm *slot/port*

Configure ATM OC-3 and OC-12 Ports

The AOS provides default values for all ATM OC-3c and ATM OC-12 port parameters. The tasks described in this section describe how to modify the default port parameters to suit your application.

To configure an ATM OC-3 or OC-12 port, perform the following tasks:

- Set the Transmit Data Clock Source
- Configure Framing for the Port
- Change the Idle Cell Header and Payload
- Enable Scrambling
- Configure Common Port Parameters
- Enable the Port

Set the Transmit Data Clock Source

The transmit data clock on an ATM OC-3 or ATM OC-12 port can be generated internally by the port (the default), or derived from the received clock. To configure the clock source, enter the following command in port configuration mode:

clock-source {**internal** | **line**}

Configure Framing for the Port

By default, ATM OC-3 ports are configured to use SONET/STS-3c framing. To configure the port to use SDH/STM-1 framing, enter the following command in port configuration mode:

framing stm1

ATM OC-12 ports are configured by default to use SONET/STS-12c framing. To configure the port to use SDH/STM-4 framing, enter the following command in port configuration mode:

framing stm4

Change the Idle Cell Header and Payload

By default, idle cells transmitted by an ATM port use a header value of 0x00000000 and a payload value of 0x5A. To modify these values, enter the following command in port configuration mode:

idle-cell {**header** *header-value* | **payload** *payload-value*}

Enable Scrambling

To enable payload scrambling on an ATM port, enter the following command in port configuration mode:

scramble

Configure Common Port Parameters

See Chapter 9, “Configuring Common Port, Circuit, and Channel Parameters,” for information on configuring common port parameters.

Enable the Port

By default, all ports are configured to be shut down. To begin operations on the port, enter the following command in port configuration mode:

no shutdown

Configure ATM DS-3 and ATM E3 Ports

The AOS provides default values for all ATM DS-3 and ATM E3 port parameters. This section describes how to modify the port parameters to suit your application.

To configure ATM DS-3 and ATM E3 ports perform the following tasks:

- Configure 8kHz Timing
- Set the Cell Delineation
- Set the Transmit Data Clock Source
- Change the Idle Cell Header and Payload
- Configure the Framing Type (ATM E3 only)
- Set the Cable Length (ATM DS-3 only)
- Enable Scrambling
- Configure Common Port Parameters
- Enable the Port

Configure 8kHz Timing

By default, the transmit Physical Layer Convergence Protocol (PLCP) synchronizes to the received PLCP reference. If preferred, you can configure the ATM DS-3 or ATM E3 port to force the transmit PLCP to use an external 8kHz timing reference. To change the timing reference, enter the following command in port configuration mode:

8khztiming

Set the Cell Delineation

ATM DS-3 and ATM E3 ports can be configured to use header check sequence (HCS) framing (the default) or framing based on PLCP for cell delineation. To modify the cell delineation, enter the following command in port configuration mode:

cell-delineation {**hcs** | **plcp**}

Set the Transmit Data Clock Source

The transmit data clock on an ATM DS-3 or ATM E3 port can be generated internally by the port (the default), or derived from the received clock. To configure the clock source, enter the following command in port configuration mode:

clock-source {**internal** | **line**}

Change the Idle Cell Header and Payload

By default, idle cells transmitted by an ATM port use a header value of 0x00000000 and a payload value of 0x5A. To modify these values, enter the following command in port configuration mode:

idle-cell {**header** *header-value* | **payload** *payload-value*}

Configure the Framing Type

By default, ATM E3 ports use G.751 (PLCP) framing. To configure the framing for an ATM E3 port, enter the following command in port configuration mode:

framing {**g751** | **g832**}

Note This command does not apply to ATM DS-3 ports.

Set the Cable Length

ATM DS-3 ports must be configured to work with the cable length attached to the port. By default, the port is configured to work with a short cable (less than 225 ft). To modify the cable length, enter the following command in port configuration mode:

length {**short** | **long**}

Note This command does not apply to ATM E3 ports.

Enable Scrambling

To enable payload scrambling on an ATM port, enter the following command in port configuration mode:

scramble

Configure Common Port Parameters

See Chapter 9, "Configuring Common Port, Circuit, and Channel Parameters," for information on configuring common port parameters.

Enable the Port

By default, all ports are configured to be shut down. To begin operations on the port, enter the following command in port configuration mode:

no shutdown

Configure ATM T1 and ATM E1 Ports

The AOS provides default values for all ATM T1 and ATM E1 port parameters. This section describes the how to modify the port parameters to suit your application.

To configure ATM T1 and ATM E1 ports, perform the following tasks:

- Set the Cable Length (ATM T1 only)
- Set the Transmit Data Clock Source
- Configure Framing
- Enable FDL Performance Report Transmission (ATM T1 only)
- Configure Linecode (ATM T1 only)
- Change the Idle Cell Header and Payload
- Enable Scrambling
- Configure Yellow Alarms
- Configure Common Port Parameters
- Enable the Port

Set the Cable Length

ATM T1 ports must be configured to work with the cable length attached to the port. By default, the port is configured to work with a short cable (less than 660 ft).To modify the cable length, enter the following command in port configuration mode:

cablelength {**long** {**10db** | **-7.5db** | **-15db** | **-22db**} | **short** {**110** | **220** | **330** | **440** | **550** | **660**}}

Note For longer cable lengths (greater than 660 ft), you must specify a transmit power level, in decibels.

Note This command does not apply to ATM E1 ports.

Set the Transmit Data Clock Source

The transmit data clock on an ATM T1 or ATM E1 port can be generated internally by the port (the default), or derived from the received clock. To configure the clock source, enter the following command in port configuration mode:

clock-source {**internal** | **line**}

Configure Framing

By default, ATM T1 ports use Extended Superframe Format (ESF) framing. You can optionally configure the port to use Superframe Format (SF), also known as D4 framing. To modify the framing for an ATM T1 port, enter the following command in port configuration mode:

framing {**esf** | **sf**}

By default, ATM E1 ports use CRC4 framing and you can optionally configure the port to no framing, per the ITU G.704 specification. To modify the framing for an ATM E1 port, enter the following command in port configuration mode:

framing {**crc4** | **no-crc4**}

Enable FDL Performance Report Transmission

You can optionally enable the transmission of performance reports for the T1 port using the Facility Data Link (FDL) per ANSI T1.403. To enable this transmission, enter the following command in port configuration mode:

fdl {**ansi** | **att**}

The **ansi** keyword enables a one-second transmission of the performance report. The **att** keyword enables a 15-minute transmission of the performance report.

Note To enable performance report transmissions, the port must be configured to use ESF framing. See the "Configure Framing" section.

Note This command does not apply to ATM E1 ports.

Configure Linecode

ATM T1 ports use B8ZS line coding by default. You can optionally configure the port to use alternate mark inversion (AMI) as the line coding. To change the line coding for an ATM T1 port, enter the following command in port configuration mode:

linecode {**ami** | **b8sz**}

Note This command does not apply to ATM E1 ports.

Change the Idle Cell Header and Payload

By default, idle cells transmitted by an ATM port use a header value of 0x00000000 and a payload value of 0x5A. To modify these values, enter the following command in port configuration mode:

idle-cell {**header** | **payload**} *hex-value*

Enable Scrambling

To enable payload scrambling on an ATM port, enter the following command in port configuration mode:

scramble

Configure Yellow Alarms

By default, ATM T1 and ATM E1 ports detect and generate yellow alarms. To configure a port to detect or generate yellow alarms, enter the following command in port configuration mode:

yellow-alarm {**detection** | **generation**}

Configure Common Port Parameters

See Chapter 9, "Configuring Common Port, Circuit, and Channel Parameters," for information on configuring common port parameters.

Enable the Port

By default, all ports are configured to be shut down. To begin operations on the port, enter the following command in port configuration mode:

no shutdown

Configuration Examples

The following example provides a complete configuration for an ATM OC-3 port:

```
[local]RedBack(config)#port atm 2/0
[local]RedBack(config-port)#description To DSLAM 1 (working port)
[local]RedBack(config-port)#rate-limit
[local]RedBack(config-port)#clock-source line
[local]RedBack(config-port)#framing stm1
[local]RedBack(config-port)#idle-cell header 0x00000000
[local]RedBack(config-port)#idle-cell payload 0x5a
[local]RedBack(config-port)#scramble
[local]RedBack(config-port)#circuit-creation explicit
[local]RedBack(config-port)#no shutdown
```

The following example provides a complete configuration for an ATM DS-3 port:

```
[local]RedBack(config)#port atm 3/0
[local]RedBack(config-port)#description To DSLAM 1
[local]RedBack(config-port)#rate-limit
[local]RedBack(config-port)#8khztiming
[local]RedBack(config-port)#cell-delineation hcs
[local]RedBack(config-port)#idle-cell header 0x00000000
[local]RedBack(config-port)#idle-cell payload 0x5a
[local]RedBack(config-port)#length long
[local]RedBack(config-port)#scramble
[local]RedBack(config-port)#no shutdown
```

The following example provides a complete configuration for an ATM T1 port:

```
[local]RedBack(config)#port atm 4/0
[local]RedBack(config-port)#description To DSLAM 1
[local]RedBack(config-port)#cablelength long 10db
[local]RedBack(config-port)#clock-source line
[local]RedBack(config-port)#framing esf
[local]RedBack(config-port)#fdl ansi
[local]RedBack(config-port)#linecode b8sz
[local]RedBack(config-port)#idle-cell header 0x00000000
[local]RedBack(config-port)#idle-cell payload 0x5a
[local]RedBack(config-port)#scramble
[local]RedBack(config-port)#no yellow-alarm detection
[local]RedBack(config-port)#no yellow-alarm generation
[local]RedBack(config-port)#circuit-creation explicit
[local]RedBack(config-port)#no shutdown
```

Configuring Channelized DS-3 Ports

This chapter provides an overview of channelized DS-3 ports and describes the tasks involved in configuring channelized DS-3 ports through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "Channelized DS-3 Port Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

The configuration model for the channelized DS-3 ports varies from that of other modules, because many logical channels can be created on each physical port and, in fact, many logical channels can be created within each T1 tributary of the channelized DS-3 port.

Configuration Tasks

Configuring channelized DS-3 ports typically consists of the following tasks:

- Configure a Channelized DS-3 Port
- Configure T1 Channels
- Configure HDLC Channels

Configure a Channelized DS-3 Port

Configuring a channelized DS-3 port typically consists of the following tasks:

- Define Channelized DS-3 Port Characteristics
- Configure a Loopback on the Port
- Configure Common Port Parameters
- Enable the Port

To enter port configuration mode and begin configuring a channelized DS-3 port, enter the following command in global configuration mode:

port channelized-ds3 *slot/port*

Define Channelized DS-3 Port Characteristics

AOS provides default values for DS-3 port characteristics. Perform the following tasks to modify the default parameters for a DS-3 port:

- Change the Source of the Transmit Data Clock
- Change the Framing
- Specify the Length of the Attached Cable

Change the Source of the Transmit Data Clock

By default, the transmit data clock for a DS-3 port is generated internally. You can configure the port to derive the transmit clock from the receive clock.

To select the clock source for a DS-3 port, enter the following command in port configuration mode:

clock-source {**internal** | **line**}

Change the Framing

By default, the framing on a DS-3 port is set to C-bit framing. You can configure the port to use M23 framing. To modify the framing for the channelized DS-3 port, enter the following command in port configuration mode:

framing {**c-bit** | **m23**}

Specify the Length of the Attached Cable

For the DS-3 port to operate correctly, you must configure AOS with the length of the cable attached to the DS-3 port. By default, AOS is configured to operate with a short cable (less than 225 feet). If the cable length is over 225 feet, you must configure the port to indicate that a long cable is connected.

To configure the length of the cable attached to a DS-3 port, enter the following command in port configuration mode:

length {**short** | **long**}

Configure a Loopback on the Port

You can create a loopback on a channelized DS-3 port to test port operation. The different loopback options can help to isolate the source of a port problem. The following loopback options are supported:

- The **line** option loops received data back to the transmit line for the channelized DS-3 port. All T1 channels in the DS-3 are looped.
- The **local** option loops locally generated frames back to the receiver for the channelized DS-3 port or T1 channel. All T1 channels in the DS-3 are looped.
- The **remote ds3** option verifies remote link connectivity and quality at the DS-3 signal level.
- The **remote t1** option performs remote link verification on a single DS-1 signal, or on all 28 individual DS-1 signals.

To configure a loopback on a channelized DS-3 port, enter the following command in port configuration mode:

loopback {**line** | **local** | **remote** [**ds3** | **t1** {**all** | *t1-channel*}]}

Configure Common Port Parameters

See Chapter 9, "Configuring Common Port, Circuit, and Channel Parameters" for information on configuring common port parameters.

Enable the Port

By default, all ports are configured to be shut down. To begin operations on the port, enter the following command in port configuration mode:

no shutdown

Configure T1 Channels

After you have configured a channelized DS-3 port, you can configure the hardware characteristics of the constituent T1 channels. Configuring the T1 channels typically consists of the following tasks:

- Change the Source of the Transmit Data Clock for T1 Channels
- Configure T1 Framing
- Configure Yellow Alarm
- Enable FDL Performance Report Transmission
- Display T1 Channel Configuration Information

To enter T1 channel configuration mode and begin configuring a T1 channel, enter the following command in port configuration mode:

t1 *t1-channel*

Change the Source of the Transmit Data Clock for T1 Channels

By default, the transmit data clock for a DS-3 port is generated internally. You can configure the port to derive the transmit clock from the receive clock.

To select the clock source for a DS-3 port, enter the following command in port configuration mode:

clock-source {**internal** | **line**}

Configure T1 Framing

By default, the T1 channels in a channelized DS-3 port use Extended Superframe Format (ESF) framing. You can optionally configure the port to use Superframe Format (SF), also known as D4 framing. To modify the framing for a T1 channel, enter the following command in T1 channel configuration mode:

framing {**esf** | **sf**}

Enable FDL Performance Report Transmission

You can optionally enable the transmission of performance reports for the T1 channel using the Facility Data Link (FDL) per American National Standards Institute (ANSI) T1.403. To enable this transmission, enter the following command in T1 channel configuration mode:

fdl {**ansi** | **att**}

The **ansi** option enables a one-second transmission of the performance report. The **att** option enables the enables a 15-minute transmission of the performance report. Use the **no** form of this command to disable the transmissions.

Note To enable performance report transmissions, the port must be configured to use **esf** framing. See the "Configure T1 Framing" section.

Configure Yellow Alarm

To configure T1 channels to detect and/or generate yellow alarms (the default), enter the following command in T1 channel configuration mode:

yellow-alarm {**detection** | **generation**}

Use the **no** form of this command to disable the detection or generation of yellow-alarms for a T1 channel.

Display T1 Channel Configuration Information

To display T1 channel configuration information for one or all T1 channels on a channelized DS-3 port, enter the following command in administrator exec mode:

show t1 info *slot/port* [*t1-channel*]

Configure HDLC Channels

An HDLC channel on a channelized DS-3 port is a logical channel. Configuring HDLC channels consists of the following tasks:

- Create an HDLC Channel
- Set the CRC Length
- Invert the Polarity of Bits
- Set the DS-0 speed
- Set the Encapsulation Type
- Enable the HDLC Channel
- Configuration Examples

Create an HDLC Channel

The following rules apply to HDLC channels:

- An HDLC channel can comprise as little as a single DS-0, up to a single constituent T1.
- HDLC channels cannot span T1 boundaries, but can contain any number of DS-0s within an individual T1.
- There can be multiple HDLC channels per T1, provided they consist of mutually exclusive DS-0s.

To create or modify an HDLC channel on a channelized DS-3 port, enter the following command in port configuration mode:

hdlc-channel *name* **t1** *t1-channel* **timeslot** *range*

Set the CRC Length

You can choose between 16-bit (the default) and 32-bit cyclic redundancy checking (CRC). To modify the CRC, enter the following command in HDLC channel configuration mode:

crc {**16** | **32**}

Invert the Polarity of Bits

You can configure an HDLC channel to invert the polarity of all bits in the HDLC-encoded stream. By default, the polarity is not inverted. To invert the polarity, enter the following command in HDLC channel configuration mode:

invert-data

Set the DS-0 speed

By default, the DS-0s that comprise an HDLC channel run at a speed of 64 kbps. To modify the speed of the constituent DS-0s in an HDLC channel, enter the following command in HDLC channel configuration mode:

speed {**56** | **64**}

Set the Encapsulation Type

You can select an encapsulation type for HDLC channels. The supported encapsulation types are as follows:

- Frame Relay encapsulation—This is the default encapsulation for all DS-3 ports. You can optionally configure additional parameters for Frame Relay-encapsulated ports. See Chapter 18, "Configuring Frame Relay" for additional information on configuring Frame Relay.
- Cisco High-Level Data Link Control (HDLC) encapsulation—Cisco's proprietary encapsulation of IP.
- Point-to-Point Protocol (PPP) encapsulation—Internet Engineering Task Force (IETF-)standard PPP over HDLC (RFC 1662, *PPP in HDLC-like Framing*).

To configure the encapsulation type for a channelized DS-3 port, enter the following command in port configuration mode:

encapsulation {**cisco-hdlc** | **frame-relay** | **ppp**}

If you select Cisco HDLC encapsulation, you can optionally configure the keepalive timer. By default, keepalives are configured to be sent every 10 seconds. To configure a different value, enter the following command in port configuration mode:

keepalive *seconds*

Enable the HDLC Channel

By default, all HDLC channels are configured to be shut down. To begin operations on the channel, enter the following command in HDLC channel configuration mode:

no shutdown

Configuration Examples

The following example configures two HDLC channels on the first constituent T1 of the channelized DS-3 in slot 4, port 0 of an SMS 1000. The name of the HDLC channel is in the form *t1:timeslots*, but this naming convention is arbitrary. You can use integers, proper nouns, or any other convenient naming convention because these are arbitrary character sequences. Note how the same DLCI value (18) is configured in each of the two HDLC channels. This is possible because the HDLC channels are distinct links from the point of Frame Relay:

```
[local]RedBack(config)#port channelized-ds3 4/0
[local]RedBack(config-port)#length long
[local]RedBack(config-port)#framing m23
[local]RedBack(config-port)#no shutdown
[local]RedBack(config-port)#t1 1
[local]RedBack(config-t1)#clock-source line
[local]RedBack(config-t1)#framing esf
[local]RedBack(config-t1)#fdl ansi
[local]RedBack(config-t1)#exit
[local]RedBack(config-port)#hdlc-channel 1:1-14 t1 1 timeslot 1-14
[local]RedBack(config-t1)#speed 56
[local]RedBack(config-t1)#invert-data
```

```
[local]RedBack(config-chan)#frame-relay pvc 18 profile dslam1 encapsulation ppp
[local]RedBack(config-pvc)#bind authentication chap pap
[local]RedBack(config-pvc)#exit
[local]RedBack(config-chan)#hdlc-channel 1:15-24 t1 1 teimeslot 15-24
[local]RedBack(config-chan)#frame-relay pvc 18 profile dslam2 encapsulation bridge1490
[local]RedBack(config-pvc)#bind subscriber fred@local
[local]RedBack(config)#end
[local]RedBack#
```

Configuring Clear-Channel DS-3 and Clear-Channel E3 Ports

This chapter provides an overview of clear-channel DS-3 and clear-channel E3 ports and describes the tasks involved in configuring clear-channel DS-3 and clear-channel E3 ports through the Access Operating System. For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "Clear-Channel DS-3 and Clear-Channel E3 Port Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

Clear-channel DS-3 and clear-channel E3 ports operate as a single channel.

Configuration Tasks

AOS provides default values for clear-channel DS-3 and clear-channel E3 port parameters. Typical configuration tasks you may need to perform to customize port settings to match your application include:

- Set the Cable Length (DS-3 only)
- Configure the Framing Type (DS-3 only)
- Set the Encapsulation Type for the Port
- Configure the Cisco HDLC Keepalive Timer
- Configure a Loopback on the Port
- Configure Common Port Parameters
- Enable the Port

To enter port configuration mode and begin configuring a clear-channel DS-3 port, enter the following command in global configuration mode:

port ds3 *slot/port*

To enter port configuration mode and begin configuring a clear-channel E3 port, enter the following command in global configuration mode:

port e3 *slot/port*

After you have configured the port, if you have selected Frame Relay encapsulation for the port (see the "Set the Encapsulation Type for the Port" subsection), see Chapter 18, "Configuring Frame Relay," for information on configuring Frame Relay parameters on the port.

Set the Cable Length (DS-3 only)

For the DS-3 port to operate correctly, you must configure AOS with the length of the cable attached to the DS-3 port. By default, AOS is configured to operate with a short cable (less than 225 feet). If the cable length is over 225 feet, you must use the configure the port to operate with a long cable.

To configure the length of the cable attached to a DS-3 port, enter the following command in port configuration mode:

length {**short** | **long**}

This command does not apply to clear-channel E3 ports.

Configure the Framing Type (DS-3 only)

By default, the framing on a DS-3 port is set to C-bit framing. You can configure the port to use M23 framing. To modify the framing for the channelized DS-3 port, enter the following command in port configuration mode:

framing {**c-bit** | **m23**}

This command does not apply to clear-channel E3 ports.

Configure the Transmit Data Clock Source

By default, the transmit data clock for a DS-3 or clear-channel E3 port is generated internally. You can configure the port to derive the transmit clock from the receive clock.

To select the clock source for a port, enter the following command in port configuration mode:

clock-source {**internal** | **line**}

Set the Encapsulation Type for the Port

You can select an encapsulation type for clear-channel DS-3 and clear-channel E3 ports. The supported encapsulation types are as follows:

- Frame Relay encapsulation—The default encapsulation for all DS-3 and clear-channel E3 ports. You can optionally configure additional parameters for Frame Relay-encapsulated ports. See Chapter 18, "Configuring Frame Relay" for additional information on configuring Frame Relay.

- Cisco High-Level Data Link Control (HDLC) encapsulation—Cisco's proprietary encapsulation of IP.
- Point-to-Point Protocol (PPP) encapsulation—Internet Engineering Task Force (IETF-)standard PPP over HDLC (RFC 1662, *PPP in HDLC-like Framing*).

To configure the encapsulation type for a clear-channel DS-3 or clear-channel E3 port, enter the following command in port configuration mode:

encapsulation {**cisco-hdlc** | **frame-relay** | **ppp**}

Configure the Cisco HDLC Keepalive Timer

If you select Cisco HDLC encapsulation, you can optionally configure the keepalive timer. By default, keepalives are configured to be sent every 10 seconds. To configure a different value, enter the following command in port configuration mode:

keepalive *seconds*

Configure a Loopback on the Port

You can create a loopback on a DS-3 or clear-channel E3 port to test port operation. The different loopback options can help to isolate the source of a port problem. The following loopback options are supported:

- Line loopback (**line**)—Loops all frames coming in on the receive line back to the sender.
- Local loopback (**local**)—Loops transmitted data back to the receiver internally through the framer.
- Remote loopback (**remote**)—Sends the Far End Alarm Condition (FEAC) loopback command to the remote end of the line to put the remote end in loopback. This option applies only to clear-channel DS-3 ports.

To configure a loopback of the specified type on a port, enter the following command in port configuration mode:

loopback {**line** | **local** | **remote**}

Note The **remote** loopback option does not apply to clear-channel E3 ports.

Configure Common Port Parameters

See Chapter 9, "Configuring Common Port, Circuit, and Channel Parameters," for information on configuring common port parameters.

Enable the Port

By default, all ports are configured to be shut down. To begin operations on the port, enter the following command in port configuration mode:

no shutdown

Configuration Examples

The following example displays a sample configuration for a clear-channel DS-3 port:

```
[local]RedBack(config)#port ds3 5/0
[local]RedBack(config-port)#length long
[local]RedBack(config-port)#framing c-bit
[local]RedBack(config-port)#clock-source line
[local]RedBack(config-port)#encapsulation ppp
[local]RedBack(config-port)#no shutdown
```

Configuring HSSI Ports

This chapter provides an overview of High-Speed Serial Interface (HSSI) ports and describes the tasks involved in configuring HSSI ports through the Access Operating System. For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "HSSI Port Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Configuration Tasks
- Configuration Examples

Configuration Tasks

To configure a HSSI port, perform the following tasks:

- Configure the Hardware Interface Type
- Configure the Encapsulation Type for the Port
- Configure the Cisco HDLC Keepalive Timer
- Configure a Loopback on the Port
- Configure Common Port Parameters
- Enable the Port

To enter port configuration mode and begin configuring a HSSI port, enter the following command in global configuration mode:

port hssi *slot/port*

After you have configured the port, if you have selected Frame Relay encapsulation for the port (see the "Configure the Encapsulation Type for the Port" subsection), see Chapter 18, "Configuring Frame Relay," for information on configuring Frame Relay parameters on the port.

Configure the Hardware Interface Type

By default, a HSSI port presents a data terminal equipment (DTE) interface to the remote end. To configure the hardware interface type (DTE or data communications equipment [DCE]) for a HSSI port, enter the following command in port configuration mode:

hardware-interface {**dce** | **dte**}

Note This command has no dependency on the **frame-relay intf-type** command. This command is used to configure the interface at the hardware level, while the **frame-relay intf-type** command defines the LMI interface at a software level.

Configure the Encapsulation Type for the Port

You can select an encapsulation type for HSSI ports. The supported encapsulation types are as follows:

- Frame Relay encapsulation—This is the default encapsulation for all HSSI ports. You can optionally configure additional parameters for Frame Relay-encapsulated ports. See Chapter 18, “Configuring Frame Relay” for additional information on configuring Frame Relay.
- Cisco High-level Data Link Control (HDLC) encapsulation—This is Cisco’s proprietary encapsulation of IP.
- Point-to-Point Protocol (PPP) encapsulation—IETF-standard PPP over HDLC (RFC 1662).

To configure the encapsulation type for HSSI ports, enter the following command in port configuration mode:

encapsulation {**cisco-hdlc** | **frame-relay** | **ppp**}

Configure the Cisco HDLC Keepalive Timer

If you select Cisco HDLC encapsulation, you can optionally configure the keepalive timer. By default, keepalives are configured to be sent every 10 seconds. To configure a different value, enter the following command in port configuration mode:

keepalive *seconds*

Configure a Loopback on the Port

You can create a loopback on a HSSI port to test port operation. The different loopback options can help to isolate the source of a port problem. The following loopback options are supported:

- Internal loopback (**internal**)—Loops locally generated frames back to the receiver internally in the DSCC4.
- Local loopback (**local**)—Loops transmitted data back to the receiver after it goes through the DSCC4, at the edge of the card.

To configure a loopback of the specified type on a HSSI port, enter the following command in port configuration mode:

loopback {**internal** | **local**}

Configure Common Port Parameters

See Chapter 9, “Configuring Common Port, Circuit, and Channel Parameters,” for information on configuring common port parameters.

Enable the Port

By default, all ports are configured to be shut down. To begin operations on the port, enter the following command in port configuration mode:

no shutdown

Configuration Examples

The following example displays a sample configuration for a HSSI port:

```
[local]RedBack(config)#port hssi 3/0
[local]RedBack(config-port)#description HSSI port
[local]RedBack(config-port)#hardware-interface dce
[local]RedBack(config-port)#encapsulation cisco-hdlc
[local]RedBack(config-port)#keepalive 20
[local]RedBack(config-port)#no shutdown
```

Configuring Packet T1 and E1

This chapter provides an overview of packet T1 and E1 ports and describes the tasks involved in configuring packet T1 and E1 ports through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed under "Configuration Tasks," see the "Packet T1 and E1 Commands" chapter in the *Access Operating System (AOS) Command Reference*.

This chapter contains the following sections:

- Configuration Tasks
- Configuration Examples

Configuration Tasks

Configuring packet T1 and E1 ports typically consists of the following tasks:

- Define Packet T1 and E1 Port Characteristics
- Configure Common Port Parameters
- Set the Encapsulation Type
- Enable the Port
- Display T1/E1 Configuration Information
- Display Performance Monitoring Statistics
- Clear Performance Monitoring Statistics

To begin configuring a packet T1 port, enter the following command in global configuration mode:

port ds1 *slot/port*

To begin configuring a packet E1 port, enter the following command in global configuration mode:

port e1 *slot/port*

After you have configured the port, if you have selected Frame Relay encapsulation for the port (see "Set the Encapsulation Type"), see Chapter 18, "Configuring Frame Relay" for information on configuring Frame Relay parameters on the port.

Define Packet T1 and E1 Port Characteristics

AOS provides defaults for packet T1 and E1 port parameters. The following tasks can be performed to modify the default configuration for a packet T1 or E1 port:

- Change the Source of the Transmit Data Clock
- Specify the Length of the Attached Cable (Packet T1 only)
- Change the Framing Type
- Enable FDL Performance Report Transmission (Packet T1 only)
- Modify the Line Coding (Packet T1 only)
- Invert the Polarity of Bits
- Modify the DS-0 or E0 Speed
- Configure Yellow Alarm
- Define the Timeslots

Change the Source of the Transmit Data Clock

By default, the transmit data clock for a packet T1 or E1 port is generated internally. You can configure the port to derive the transmit clock from the receive clock.

To select the clock source for a packet T1 or E1 port, enter the following command in port configuration mode:

clock-source {**internal** | **line**}

Specify the Length of the Attached Cable (Packet T1 only)

By default, a packet T1 port is configured to operate with a short cable (less than 110 feet long). If the cable length is longer than 110 feet, but less than 660 feet, you must configure the port with the **cablelength short** command, and specify the maximum cable length. If the cable length is longer than 660 feet, you must configure the port with the **cablelength long** command, and specify the transmit power level, in decibels. To configure the packet T1 cable length, enter the following command in port configuration mode:

cablelength {**long** {**10db** | **-7.5db** | **-15db** | **-22db**} | **short** {**110** | **220** | **330** | **440** | **550** | **660**}}

Change the Framing Type

By default, a packet T1 port is configured with Extended Superframe Format (ESF) framing. You can also configure a packet T1 port to use Superframe Format (or D4). To modify the framing for a packet T1 port, enter the following command in port configuration mode:

framing {**esf** | **sf**}

By default, a packet E1 port is configured to use CRC4 framing, per the International Telecommunication Union (ITU) G.704 specification. You can also configure a packet E2 port to use no CRC4 framing, per the ITU G.704 specification. To modify the framing for a packet E1 port, enter the following command in port configuration mode:

framing {**crc4** | **no-crc4**}

Enable FDL Performance Report Transmission (Packet T1 only)

You can optionally enable the transmission of performance reports for the T1 port using the Facility Data Link (FDL) per American National Standards Institute (ANSI) T1.403. To enable this transmission, enter the following command:

fdl {**ansi** | **att**}

The **ansi** option enables a one-second transmission of the performance report. The **att** option enables a 15-minute transmission of the performance report. Use the **no** form of this command to disable the transmissions.

Note To enable performance report transmissions, the port must be configured to use ESF framing. See the "Change the Framing Type" section.

Modify the Line Coding (Packet T1 only)

Packet T1 ports can be configured to use B8ZS (the default) or alternate mark inversion (AMI) line coding. To modify the line coding for a packet T1 port, enter the following command in port configuration mode:

linecode {**ami** | **b8zs**}

Invert the Polarity of Bits

You can configure a packet T1 or E1 port to invert the polarity of all bits in the High-Level Data Link Control (HDLC-) encoded data stream. To invert the polarity, enter the following command in port configuration mode:

invert-data

Modify the DS-0 or E0 Speed

By default, the DS-0s that comprise a T1 channel run at a speed of 64 kbps. To modify the speed of the constituent DS-0s in a T1 channel, enter the following command in port configuration mode:

speed {**56** | **64**}

Configure Yellow Alarm

To configure T1 channels to detect and/or generate yellow alarms (the default), enter the following command in port configuration mode:

yellow-alarm {**detection** | **generation**}

Use the **no** form of this command to disable the detection or generation of yellow alarms for a T1 channel.

Define the Timeslots

You can configure the timeslots that will comprise the HDLC channel within a T1 or E1 port. To specify the timeslots, enter the following command in port configuration mode:

timeslot *range*

For packet E1 ports only, timeslot 16 must be explicitly included by entering the following command in port configuration mode:

ts16

Configure Common Port Parameters

See Chapter 9, “Configuring Common Port, Circuit, and Channel Parameters” for information on how to configure common port parameters.

Set the Encapsulation Type

You can select an encapsulation type for a packet E1 or packet T1 port. The supported encapsulation types are as follows:

- Frame Relay encapsulation—This is the default encapsulation for all packet E1 and packet T1 ports. You can optionally configure additional parameters for Frame Relay-encapsulated ports. See Chapter 18, “Configuring Frame Relay” for additional information on configuring Frame Relay.
- Cisco High-level Data Link Control (HDLC) encapsulation—This is Cisco’s proprietary encapsulation of IP.
- PPP encapsulation—Internet Engineering Task Force (IETF) standard RFC 1662, *PPP in HDLC-like Framing*.

To configure the encapsulation type for a packet E1 or packet T1 port, enter the following command in port configuration mode:

encapsulation {**cisco-hdlc** | **frame-relay** | **ppp**}

If you select Cisco HDLC encapsulation, you can optionally configure the keepalive timer. By default, keepalives are every 10 seconds. To configure a different value, enter the following command in port configuration mode:

keepalive *seconds*

Enable the Port

By default, all ports are configured to be shut down. To begin operations on the port, enter the following command in port configuration mode:

no shutdown

Display T1/E1 Configuration Information

To display configuration information for a packet T1 or E1 port, enter the following command in administrator exec mode:

show t1 info *slot/port*

Display Performance Monitoring Statistics

To display performance monitoring statistics for a packet T1 or E1 port, enter the following command in operator exec mode:

show pmon [*slot/port*] [**pm** [**tabular**] [*interval*]]

Clear Performance Monitoring Statistics

To clear the performance monitoring information for a T1 port, enter the following command in administrator exec mode:

clear pmon *slot/port* [**-noconfirm**]

Configuration Examples

The following example shows a sample configuration for both a packet T1 and packet E1 port:

```
[local]RedBack(config)#port t1 7/1
[local]RedBack(config-port)#cablelength long 10db
[local]RedBack(config-port)#clock-source internal
[local]RedBack(config-port)#framing esf
[local]RedBack(config-port)#fdl ansi
[local]RedBack(config-port)#encapsulation cisco-hdlc
[local]RedBack(config-port)#keepalive 30
[local]RedBack(config-port)#timeslot 1-10
[local]RedBack(config-port)#port e1 3/0
[local]RedBack(config-port)#framing no-crc4
[local]RedBack(config-port)#invert-data
[local]RedBack(config-port)#speed 56
[local]RedBack(config-port)#timeslot 1-20
[local]RedBack(config-port)#ts16
```

Configuring Packet over SONET Ports

This chapter provides an overview of Packet over Synchronous Optical Network (SONET)/Synchronous Digital Hierarchy (SDH) and describes the tasks involved in configuring Packet Over SONET (POS) ports through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "Packet Over SONET Port Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

The AOS supports various POS I/O modules, including OC-3c and OC-12c.

Configuration Tasks

Configuring a POS port typically consists of the following tasks:

- Define Port Characteristics
- Set the Encapsulation Type
- Configure Common Port Parameters
- Enable the Port

After you have configured the port, you must define a binding. See Chapter 20, "Configuring Bindings" for information on configuring bindings.

Define Port Characteristics

This section discusses how to configure port parameters specific to POS ports. You can:

- Change the Source of the Transmit Data Clock
- Enable Payload Scrambling
- Modify the Path Signal Label (C2) Byte
- Change the Framing
- Specify a 16-Bit CRC
- Specify the Packet Length
- Create a Loopback

To enter port configuration mode and begin configuring a POS port, enter the following command in global configuration mode:

port pos *slot/port*

After you have configured the port, if you have selected Frame Relay encapsulation for the port (see "Set the Encapsulation Type"), see Chapter 18, "Configuring Frame Relay" for information on configuring Frame Relay parameters on the port.

Change the Source of the Transmit Data Clock

By default, the transmit data clock for a POS port is generated internally. You can configure the port to derive the transmit clock from the receive clock.

To select the clock source for a POS port, enter the following command in port configuration mode:

clock-source {**internal** | **line**}

Enable Payload Scrambling

Payload scrambling is enabled by default. To enable "X^43 + 1" payload scrambling, enter the following command in port configuration mode:

scramble

To disable payload scrambling, enter the following command in port configuration mode:

no scramble

Note When you modify scrambling on the port using the **scramble** or **no scramble** command, the AOS resets the value of the Path Signal Label (C2) byte to a default value, as specified in RFC 1615, *PPP over SONET/SDH*; see the "Modify the Path Signal Label (C2) Byte" section.

Modify the Path Signal Label (C2) Byte

RFC 1615, *PPP over SONET/SDH*, specifies two values for the Path Signal Label (C2) byte:

- PPP with scrambling—22 (hexadecimal 0x16)
- PPP with no scrambling—207 (hexadecimal 0xCF)

The AOS automatically configures the C2 byte to match these values, depending on whether scrambling is enabled on the port.

If you need to configure a different C2 byte value to interoperate with another vendor's equipment, first configure scrambling (see "Enable Payload Scrambling"), then override the C2 byte value. To override the C2 byte value, enter the following command in port configuration mode:

c2byte *value*

Change the Framing

The default framing on a POS port is SONET. To configure a port to use SDH framing, enter the following command in port configuration mode:

framing sdh

Specify a 16-Bit CRC

The AOS uses a 32-bit cyclic redundancy check (CRC), as specified in *RFC 1615, PPP over SONET/SDH* by default. Although the 32-bit Frame Check Sequence (FCS) is always recommended, you can modify an OC-3 POS port with SONET or SDH framing to use a 16-bit CRC. To select the 16-bit CRC, enter the following command in port configuration mode:

crc16

Specify the Packet Length

To configure the maximum HDLC frame length, in bytes, for a port, enter the following command in port configuration mode:

packet-length *value*

By default, the AOS configures POS ports to use 16,384 bytes as the maximum frame length.

Create a Loopback

You can create a loopback on a POS port to test port operation. The various loopback options can help to isolate the source of a port problem. The following loopback options are supported:

- Diagnostic loopback (**diag**)—Connects the transmit to the receive lines on the serializer chip to test operation on the serializer.
- Line loopback (**line**)—Routes retimed serial data from the receive section to the transmitter outputs on the serializer to test operation between the serializer and the SONET framer.
- Local loopback (**local**)—Connects the transmit queue to the receive queue at the SONET framer to test operation of the SONET framer.
- SONET PHY internal loopback (**t2r**)—Connects the transmit queue to the receive queue at the SONET PHY to test operation of the SONET PHY.
- SONET PHY line loopback (**r2t**)—Connects the receive queue to the transmit queue at the SONET PHY to test end-to-end operation on the port.

To configure a loopback on a POS port, enter the following command in port configuration mode:

loopback {**diag** | **line** | **local** | **r2t** | **t2r**}

Set the Encapsulation Type

You can select the encapsulation type for a POS port. The supported encapsulation types are as follows:

- Frame Relay encapsulation—This is the default encapsulation for all POS ports. You can optionally configure additional parameters for Frame Relay-encapsulated ports. See Chapter 18, “Configuring Frame Relay” for additional information on configuring Frame Relay.
- Cisco High-level Data Link Control (HDLC) encapsulation—This is Cisco’s proprietary encapsulation of IP.
- PPP encapsulation—Internet Engineering Task Force (IETF) standard Point-to-Point Protocol (PPP) encapsulation over SONET and SDH circuits, as specified in RFC 2615, *PPP over SONET/SDH*, and RFC 2662, *PPP in HDLC-like Framing*.

To configure the encapsulation type for a POS port, enter the following command in port configuration mode:

encapsulation {**cisco-hdlc** | **frame-relay** | **ppp**}

If you select Cisco HDLC encapsulation, you can optionally configure the keepalive timer. By default, keepalives are sent every 10 seconds. To configure a different value, enter the following command in port configuration mode:

keepalive *seconds*

Configure Common Port Parameters

Chapter 9, “Configuring Common Port, Circuit, and Channel Parameters,” describes how to configure common port parameters.

Enable the Port

By default, all ports are configured to be shut down. To begin operations on the port, enter the following command in port configuration mode:

no shutdown

Configuration Examples

The following example shows a sample configuration for a POS OC-3 port. The port is configured to enable payload data scrambling, to use SDH framing with a 16-bit CRC, and to use PPP encapsulation:

```
[local]RedBack(config)#port pos 4/0
[local]RedBack(config-port)#description POS port
[local]RedBack(config-port)#scramble
[local]RedBack(config-port)#framing sdh
[local]RedBack(config-port)#crc16
[local]RedBack(config-port)#encapsulation ppp
[local]RedBack(config-port)#no shutdown
```

Configuring ATM

This chapter provides an overview of the Asynchronous Transfer Mode (ATM) features supported by the Access Operating System (AOS) and describes the tasks involved in configuring ATM features through AOS. For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "ATM Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

Using ATM, data flows over permanent virtual circuits (PVCs) on an ATM port. These PVCs are identified by a virtual path identifier (VPI) and virtual channel identifier (VCI). After you have configured the physical ATM port characteristics (see Chapter 11, "Configuring ATM Ports"), you can create these PVCs as well as configure other ATM features.

Configuration Tasks

Configuring ATM features through the AOS typically consists of the following tasks:

- Configure an ATM Profile
- Configure ATM PVCs
- Configure IMA

Configure an ATM Profile

An ATM profile is required to create ATM PVCs. ATM profiles contain common configuration information that is used by all PVCs that reference the profile.

Typically, you will configure at least one ATM profile for each traffic class that you will support on a system. You might also require additional ATM profiles for PVCs with other special requirements, such as counters or bulk statistics collection.

To configure an ATM profile, you must perform the following tasks:

- Name the ATM Profile
- Configure Traffic Shaping
- Set the Transmit Queue Size
- Set the Cell Loss Priority
- Set RADIUS Attributes
- Enable PVC Statistics
- Enable and Configure Bulk Statistics

See the "Configuration Examples" section for examples on creating profiles.

Name the ATM Profile

To create a new ATM profile, or to modify an existing ATM profile, enter the following command in global configuration mode:

atm profile *prof-name*

After you enter this command, you are in ATM profile configuration mode, and can configure the parameters for the named profile.

Configure Traffic Shaping

Each ATM profile must define the type of traffic shaping to use for the PVCs that reference the profile. The following types of service are supported:

- Unspecified bit rate (UBR)—UBR mode is the simplest type of traffic shaping. It provides no specific quality of service or guaranteed throughput. UBR mode is typically used to carry LAN and WAN traffic. You do not specify any parameters when you configure the UBR service class on AOS. AOS does not support configuration of a peak cell rate (PCR) when running in UBR mode.

  To configure UBR service for all PVCs referencing an ATM profile, enter the following command in ATM profile configuration mode:

  shaping ubr

- Constant bit rate (CBR)—CBR shaping supports realtime applications that are sensitive to delay variations; for example, voice and video. To configure the CBR traffic class, you must specify two parameters:

  — Rate is the traffic bit rate, specified in kbps.

  — Cell delay variation tolerance (CDVT) controls how much cell clustering is allowed and is specified in microseconds.

To configure CBR shaping, enter the following command in ATM profile configuration mode:

shaping cbr rate *rate* **cdv** *cdv*

- Guaranteed Frame Rate (GFR)—GFR differs from the other traffic classes in that it is frame-based instead of cell-based; the service accepts or rejects entire frames. This service provides a minimum service rate during network congestion, while users are able to send at a higher rate during noncongested intervals.

Note The GFR service class is available only for ATM Version 2 I/O modules for the Subscriber Management System (SMS) device.

To configure GFR, three parameters must be specified:

— Minimum Cell Rate (MCR) specifies the minimum cell rate that should be guaranteed on a PVC, in kbps.

— Sustained Cell Rate (SCR) specifies the average rate at which traffic should be maintained in kbps AOS supports a minimum SCR value of 64 kbps, and a maximum value equal to the line rate for a given port. The minimum effective increment for the value is 8 kbps.

— Burst Tolerance (BT) specifies the amount of time (in microseconds) that traffic can be transmitted at the peak cell rate.

To configure an ATM profile with GFR service, enter the following command in ATM profile configuration mode:

shaping gfr mcr *mcr* **scr** *scr* **bt** *bt*

- Variable bit rate nonrealtime (VBR-nrt)—VBR-nrt mode supports applications that have variable rate, bursty traffic characteristics. This mode is suitable for critical data applications.

  With the VBR-nrt shaping, four parameters must be specified:

  — Peak cell rate (PCR) is the maximum rate at which traffic can be sent, measured in kbps. AOS supports a minimum PCR value of 64 kbps, and a maximum value equal to the line rate for a given port. The minimum effective increment for the value is 8 kbps.

  — Cell delay variation tolerance (CDVT) controls how much cell clustering is allowed and is specified in microseconds.

  — Sustained cell rate (SCR) represents the average rate at which traffic should be maintained in kbps. AOS supports a minimum SCR value of 64 kbps, and a maximum value equal to the line rate for a given port. The minimum effective increment for the value is 8 kbps.

  — Burst tolerance (BT) specifies the amount of time that traffic can be transmitted at the peak cell rate.

  To configure VBR-nrt service, enter the following command in ATM profile configuration mode:

  shaping vbr-nrt pcr *pcr* **cdvt** *cdvt* **scr** *scr* **bt** *bt*

- Variable bit rate realtime (VBR-rt)—VBR-rt mode supports time-sensitive applications that also require constrained delay and delay variation; for example, compressed audio.

  With the VBR-rt shaping, two parameters must be specified:

  — Peak cell rate (PCR) is the maximum rate at which traffic can be sent, measured in kbps. AOS supports a minimum PCR value of 64 kbps, and a maximum value equal to the line rate for a given port. The minimum effective increment for the value is 8 kbps.

— Cell delay variation tolerance (CDVT) controls how much cell clustering is allowed and is specified in microseconds.

To configure VBR-nrt service for an ATM profile, enter the following command in ATM profile configuration mode:

shaping vbr-rt pcr *pcr* **cdvt** *cdvt* **scr** *scr* **bt** *bt*

Set the Transmit Queue Size

You can limit the total number of outbound packet buffers that can be consumed by any PVC referencing this ATM profile. The default is 50 packet buffers. To modify this value, enter the following command in ATM profile configuration mode:

buffers transmit *value*

Caution Improper setting of this value can have severe consequences on overall system performance.

Set the Cell Loss Priority

By default, the cell loss priority (CLP) bit in all cells transmitted by AOS is not set. To set the CLP bit in all cells transmitted over PVCs referencing this ATM profile, enter the following command in ATM profile configuration mode:

clpbit

Set RADIUS Attributes

By default, the Redback vendor-specific attribute is not sent in Remote Access Dial-In User Service (RADIUS) Access-Request and Accounting-Request packets. To enable the sending of the attribute, and to select the value to be sent, enter the following command in ATM profile configuration mode:

radius attribute medium-type {**dsl** | **cable** | **wireless** | **satellite**}

Enable PVC Statistics

By default, AOS does not collect PVC statistics because of the potentially large amount of memory needed. To enable statistics collection for all PVCs referencing this ATM profile, enter the following command in ATM profile configuration mode:

counters [**l2** | **multicast**]

To obtain multicast statistics, Internet Group Management Protocol (IGMP) proxy must be enabled on the interface and context to which the PVC is bound. See Chapter 36, “Configuring IGMP Proxy” for additional information on IGMP.

Enable and Configure Bulk Statistics

To define the statistics schema for the contents of the bulkstats collection file for any PVC referencing this ATM profile, enter the following command in ATM profile configuration mode:

bulkstats schema *schema-name* **format** *format-string* [*AOS-variable* [*AOS-variable...*]]

Note You can configure multiple schemas, each gathering a different type and format of data. However, you should restrict the use of multiple schemas to global data collection, and create only one schema per ATM profile. Otherwise, you could apply a profile with several schemas to a large number of PVCs, affecting overall system performance.

See Chapter 43, "Configuring Bulk Statistics," for additional information on configuring bulk statistics.

Display Profile Information

To display ATM profile information for a single ATM profile, or all ATM profiles, enter the following command in operator exec mode:

show atm profile [*prof-name*]

Configure ATM PVCs

ATM permanent virtual circuits (PVCs) can be created explicitly via the configuration file or on-demand, as activity is detected on a port. Regardless of how ATM PVCs are to be created, each ATM PVC must reference a previously-configured ATM profile and specify an encapsulation type for the PVC.

The following sections explain how to configure ATM PVCs:

- Create Explicit ATM PVCs
- Create On-Demand ATM PVCs
- Configure ATM PVC Parameters
- Bind the ATM PVCs
- Debug an ATM PVC

Create Explicit ATM PVCs

To create a single ATM PVC or a range of explicit ATM PVCs with similar characteristics, enter the following command in port configuration mode:

atm pvc *vpi vci* [**through** *end-vci*] **profile** *prof-name* **encapsulation** {**auto1483** | **bridge1483** | **route1483** | **dot1q** | **l2tp** [**vc-muxed**] | **multi** | **ppp** [**auto** | **over-ethernet** | **serial** | **nlpid** | **llc** | **vc-muxed**]}

When you use the **through** keyword to create a range of PVCs, AOS generates an **atm pvc** command for each individual PVC in the specified range. This can result in a large configuration file. To create an explicit range of ATM PVCs that generates only6 a single command line in the configuration file, enter the following command in port configuration mode:

atm pvc explicit *start-vpi***:***start-vci* **through** *end-vpi***:***end-vci* **profile** *prof-name* **encapsulation** {**auto1483** | **bridge1483** | **route1483** | **multi** | **ppp** [**auto** | **over-ethernet** | **serial** | **nlpid** | **llc** | **vc-muxed**]}

The **atm pvc explicit** command generates a single command line in the configuration file.

Note You can use the **atm pvc** command to create explicit PVCs within an on-demand range of ATM PVCs; the **atm pvc** command overrides the **atm pvc explicit** configuration for the specified PVCs.

After you enter either of these commands, AOS enters circuit configuration mode, and you can configure the PVC or range of PVCs. If you specify the **dot1q** keyword for the encapsulation type, you enter dot1q encapsulation configuration mode. In this mode, you can define 802.1Q PVCs. See Chapter 19, "Configuring 802.1Q" for additional information on configuring 802.1Q features.

Create On-Demand ATM PVCs

To create a range of on-demand ATM PVCs, enter the following command in port configuration mode:

atm pvc on-demand *start-vpi:start-vci* **through** *end-vpi:end-vci* {**profile** *prof-name* **encapsulation** {**auto1483** | **bridge1483** | **route1483** | **multi** | **ppp** [**auto** | **over-ethernet** | **serial** | **nlpid** | **llc** | **vc-muxed**]} | **aaa context** *ctx-name* [**prefix-string** *text*]}

Note You can use the **atm pvc** command to create explicit PVCs within an on-demand range of ATM PVCs; the **atm pvc** command overrides the on-demand configuration for the specified PVCs.

After you have entered this command, AOS enters circuit configuration mode, and you can configure the range of PVCs.

Configure ATM PVC Parameters

You can optionally configure the following ATM PVC parameters:

- Description—To provide a textual description for the PVC, enter the following command in circuit configuration mode:

  description *text*

- IP Host—You can install a permanent entry in the [system or context?] host table for a host where dynamic address resolution through the Address Resolution Protocol (ARP) is not possible or not wanted. To add a host table entry for the PVC, enter the following command in circuit configuration mode:

  ip host *ip-address* [*mac-address*]

Note This command is not available when you are configuring a range of PVCs. It is also not available for PPP-encapsulated PVCs. You can only use this command after you have bound a PVC to an interface (see Chapter 20, "Configuring Bindings").

- MAC Address—If you have configured the PVC with the PPP over Ethernet (PPPoE) encapsulation, enter the following command in circuit configuration mode to establish the source Ethernet MAC address to use for PPPoE packets sent on the PVC:

  mac address *mac-address*

- Common circuit parameters—See Chapter 9, "Configuring Common Port, Circuit, and Channel Parameters" for information on configuring common circuit parameters.

Bind the ATM PVCs

You must configure the type of binding that is used for the PVCs. See Chapter 20, "Configuring Bindings," for information on how to configure bindings.

Debug an ATM PVC

Once an ATM PVC is bound, you can enable debugging on the PVC to see a display of the packet header and 60 bytes of payload data for all packets received and sent on an ATM PVC. To enable debugging on an ATM PVC, enter the following command in administrator exec mode:

debug atm *slot/port* **pvc** *vpi vci*

Configure IMA

Our Inverse Multiplexing for ATM (IMA) feature enables you to configure multiple ports on an ATM T1 I/O module to operate as a single ATM link. For example, you can configure the four ports on the 4-port ATM T1 I/O module for the SMS 500 to operate as a single link resulting in a bandwidth of 6 Mbps.

To configure IMA, perform the tasks in the following sections:

- Create an IMA Group
- Define the Constituent Ports
- Configure Optional IMA Group Parameters
- Configure IMA Ports
- Configure IMA PVCs
- Enable the IMA Group

See the "Configuration Examples" section for IMA configuration examples.

Create an IMA Group

To create an IMA group and enter IMA group configuration mode, enter the following command in global configuration mode:

ima group *group-id*

Define the Constituent Ports

To define the ports that constitute the IMA group, enter the following command in IMA group configuration mode:

ports *slot/port* [*slot/port ...*] **pvc-config** *slot/port*

You can specify one or more ports that constitute the IMA group. All ports in an IMA group must exist on the same module.

The **pvc-config** *slot/port* construct specifies which port will contain the PVC configuration for the IMA group. The port must be one of the constituent ports specified earlier in the command. Any PVCs defined in other ports that are a member of the group are deleted. If you remove a port from an IMA group, you must reconfigure the PVCs for the port.

Use the **no** form of the command to remove a constituent port from the IMA group as follows:

no ports *slot/port* [*slot/port ...*]

Configure Optional IMA Group Parameters

This section describes optional IMA group parameters that you can modify. If you do not modify these parameters, the default values will be used for the IMA group.

- Configure Transmit Clock Source Parameters

  You can configure the IMA group to use a common transmit clock (CTC) source for all ports in the group, or to use an independent transmit clock (ITC) source for each port. If you configure the IMA group to use a common transmit clock source, you can then specify whether to use the on-board oscillator as the transmit clock source, or to derive the common transmit clock source from the receive clock of one of the IMA ports. If you configure the IMA group to use independent transmit clock sources, then the transmit clock for each port is derived from the receive clock on each individual port in the group.

  — To configure the clock mode for the IMA group, enter the following command in IMA group configuration mode:

  clock mode {**common** | **independent**}

  — If you have specified the common clock mode, you must specify the common transmit clock source. To specify the common transmit clock source, enter the following command in IMA group configuration mode:

  clock source {**internal** | **line** [*slot/port*]}

  The keyword **internal** (the default) specifies that the on-board oscillator is used as the transmit clock source for the IMA group. The keyword **line** specifies that all ports in the IMA group should use a transmit clock source derived from the receive clock on the specified port. If the port is not specified, the lowest numbered port in the group is used.

- Configure the Frame Length

  The default frame length for the IMA group is 128 bytes. You can optionally change this value. To change the frame length used for the IMA group, enter the following command in IMA group configuration mode:

  frame-length *length*

  The valid values for length are: 32, 64, 128, and 256.

- Configure the Delay Tolerance

  The default delay tolerance for the IMA group is 25 milliseconds. You can optionally change this value. To change the delay tolerance for the IMA group, enter the following command in IMA group configuration mode:

  delay-tolerance *time*

  The valid range of values is 0 through 100.

- Configure the Description

  You can optionally configure a textual description for the IMA group. To do so, enter the following command in IMA group configuration mode:

  description *text*

- Configure the Number of Active Links (optional)

  By default, one link in the IMA group must be up for the IMA group to be active. To configure the minimum number of links that must be up for the IMA group to be active, enter the following command in IMA group configuration mode:

  minimum-links *count*

- Configure Symmetry Parameters

  You can define the following symmetry parameters for an IMA group:

  — Configuration

  Symmetric configuration (the default) specifies that the same number of links must be configured in each direction. Asymmetric configuration enables you to use a different number of links in each direction.

  — Operation

  Symmetric operation (the default) specifies that a port is not used to forward traffic if the port has failed in the receive direction. Using asymmetric operation enables a port to be used to forward traffic, even if the port has failed in the receive direction. You cannot specify symmetric operation if you have specified asymmetric configuration.

  By default, the group uses symmetric configuration and symmetric operation. To modify the symmetry parameters for the group, enter the following command in IMA group configuration mode:

  symmetry configuration {**symmetric** | **asymmetric**} **operation** {**symmetric** | **asymmetric**}

Configure IMA Ports

Configure the parameters of each constituent port in the IMA group. To enter port configuration mode for each port, enter the following command in global configuration mode:

port atm

The following T1 port commands apply to the ATM T1 I/O module. See the *Access Operating System (AOS) Command Reference* publication for a complete description of each command.

- **clock source**
- **framing**
- **linecode**
- **loopback**
- **police**
- **rate-limit**

Configure IMA PVCs

You must configure PVCs for the IMA group. The PVCs are configured on the port that you specified in the **pvc-config** construct in the **ports** command in IMA group configuration mode. PVCs configured for any other port in the IMA group are ignored.

See the "Configure ATM PVCs" section for information on creating and configuring ATM PVCs.

Enable the IMA Group

After you have configured an IMA group, you must enable the group. To enable an IMA group, enter the following command in global configuration mode:

ima enable *group-id*

Configure 802.1Q to RFC 1483 Bridged Internetworking

See Chapter 20, "Configuring Bindings," for information on how to bind an ATM PVC to an Ethernet port to configure 802.1Q to RFC 1483 bridged internetworking.

Configuration Examples

This section provides several example configurations for the various ATM software features:

- ATM Profiles
- ATM PVCs
- IMA

ATM Profiles

The following example shows the configuration of two ATM profiles. The ATM profile named `vbrnrt-basic` provides a basic profile to use for PVCs that will support VBR-nrt traffic. The ATM profile named `vbrnrt-stats` also supports VBR-nrt traffic, as well as layer 2 statistics collection. This profile would be used for PVCs that require monitoring.

```
[local]RedBack(config)#atm profile vbrnrt-basic
[local]RedBack(config-atmpro)#shaping vbr-nrt pcr 100000 cdv 5000 scr 80000 bt 8000
[local]RedBack(config)#radius attribute medium-type dsl
[local]RedBack(config)#atm profile vbrnrt-stats
[local]RedBack(config-atmpro)#shaping vbr-nrt pcr 100000 cdv 5000 scr 80000 bt 8000
[local]RedBack(config)#radius attribute medium-type dsl
[local]RedBack(config-atmpro)#counters l2
```

ATM PVCs

The following example shows a configuration that contains both explicit and on-demand PVCs on a single ATM port. First, the example creates an explicit range of ATM PVCs consisting of `1:1` through `1:499`. These PVCs use profile `ubr-basic` and are configured for multiple encapsulations. Bindings are created for the PVCs. Next, a range of on-demand PVCs, consisting of `2:100` through `2:150` is created. These PVCs also use profile `ubr-basic`, and are configured for multiple encapsulations. Bindings are created for these PVCs. Next, the example overwrites a PVC in this range: `1:100`. This PVC is configured to use bridged RFC 1483 encapsulation.

```
[local]RedBack(config)#atm profile ubr-basic
[local]RedBack(config-atmpro)#shaping ubr
[local]RedBack(config-atmpro)#radius attribute medium-type dsl
[local]RedBack(config-atmpro)#exit
[local]RedBack(config)#port atm 3/1
[local]RedBack(config-pvc)#atm pvc explicit 1:1 through 1:499 profile ubr-basic
encapsulation multi
[local]RedBack(config-pvc)#description Explicit PVC Range
[local]RedBack(config-pvc)#bind multi interface if1 isp1.net authentication chap
[local]RedBack(config-pvc)#atm pvc on-demand 2:1 through 2:499 profile ubr-basic
encapsulation multi
[local]RedBack(config-pvc)#description On-Demand PVCs
[local]RedBack(config-pvc)#bind multi interface if1 isp1.net authentication chap
[local]RedBack(config-pvc)#atm pvc 1 100 profile ubr-basic encapsulation bridge-1483
[local]RedBack(config-pvc)#description One Special PVC
[local]RedBack(config-pvc)#atm pvc 2 100 through 150 profile ubr-basic encapsulation
bridge-1483
[local]RedBack(config-pvc)#description Special PVC Range
[local]RedBack(config-pvc)#end
```

IMA

The following example configuration creates an IMA group (1) comprising ports 4/0, 4/1, and 4/2. Port 4/2 is specified as the **pvc-config** port. The example specifies a description for the group, and configures the ports to use a common transmit clock derived from the receive clock on port 4/2. It also sets the delay tolerance to 50, sets the frame-length to 256, and specifies that at least two of the links in the IMA group must be operational for the IMA group to be up.

```
[local]RedBack(config)#ima group 1
[local]RedBack(config-ima)#ports 4/0 4/1 4/2 pvc-config 4/2
[local]RedBack(config-ima)#description DSL feed from Provider X
[local]RedBack(config-ima)#clock mode common
[local]RedBack(config-ima)#clock source line 4/2
[local]RedBack(config-ima)#delay-tolerance 50
[local]RedBack(config-ima)#frame-length 256
[local]RedBack(config-ima)#minimum-links 2
```

After the group is configured, the physical-layer parameters for each port are configured. The following example sets the framing for each port in the IMA group and creates 100 PVCs for the IMA group by configuring the PVCs on the port that was specified as the pvc-config port for the IMA group:

```
[local]RedBack(config)#port atm 4/0
[local]RedBack(config-port)#framing sf
[local]RedBack(config)#port atm 4/1
[local]RedBack(config-port)#framing sf
[local]RedBack(config)#port atm 4/2
[local]RedBack(config-port)#framing sf
[local]RedBack(config-port)#end
[local]RedBack(config)#port atm 4/2
[local]RedBack(config-port)#atm pvc 10 1 through 100 profile ubr encapsulation ppp
[local]RedBack(config-pvc)#bind authentication pap
[local]RedBack(config-pvc)#exit
[local]RedBack(config-port)#exit
```

After the IMA group is created, and the ports are configured, the IMA group is enabled:

```
[local]RedBack(config)#ima enable 1
[local]RedBack(config)#exit
```

Configuring Frame Relay

This chapter provides an overview of Frame Relay and describes the tasks involved in configuring Frame Relay through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "Frame Relay Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

Frame Relay can be run over High-level Data Link Control (HDLC-)oriented ports and channels, such as clear-channel DS-3, channelized DS-3, packet T1, and so on. To run Frame Relay a port or channel must be configured to use Frame Relay encapsulation using the port configuration or HDLC channel configuration **encapsulation** command.

Using Frame Relay, data flows over Permanent Virtual Circuits (PVCs), identified by a Data Link Circuit Identification (DLCI) number.

Configuration Tasks

Configuring Frame Relay typically consists of the following tasks:

- Configure a Frame Relay Profile
- Configure LMI Parameters
- Configure Frame Relay Circuits

Configure a Frame Relay Profile

Frame Relay profiles contain a set of configuration commands that applies to one or more PVCs. All Frame Relay circuits must be associated with a Frame Relay profile. The following tasks are associated with configuring a Frame Relay profile:

- Create a Frame Relay Profile
- Set the Transmit Queue Size
- Set RADIUS Attributes
- Enable Circuit Statistics
- Enable and Configure Bulk Statistics
- Display Profile Information

Create a Frame Relay Profile

To create a new Frame Relay profile, or to modify an existing Frame Relay profile, enter the following global configuration command:

frame-relay profile *prof-name*

Set the Transmit Queue Size

You can limit the total number of outbound packet buffers that can be consumed by any circuit referencing this Frame Relay profile. The default is 50 packet buffers. To modify this value, enter the following command in Frame Relay profile configuration mode:

buffers transmit *value*

Caution Improper setting of this value can have severe consequences on overall system performance.

Set RADIUS Attributes

By default, the Redback vendor-specific attribute Medium-Type is not sent in RADIUS Access-Request and Accounting-Request packets. To enable the sending of the attribute, and to select the value to be sent, enter the following command in Frame Relay profile configuration mode:

radius attribute medium-type {**dsl** | **cable** | **wireless** | **satellite**}

Enable Circuit Statistics

Circuit statistics are not collected by default because of the potentially large amount of memory needed. To enable layer 2 statistics collection, multicast statistics collection, or both for all circuits referencing this Frame Relay profile, enter the following command in Frame Relay profile configuration mode:

counters [**l2** | **multicast**]

To obtain multicast statistics, Internet Group management Protocol (IGMP) proxy must be enabled on the interface and context to which the circuit is bound. See Chapter 36, “Configuring IGMP Proxy” for additional information on IGMP.

Enable and Configure Bulk Statistics

To define the statistics schema for the contents of the bulkstats collection file for any circuit referencing this Frame Relay profile, enter the following command in Frame Relay profile configuration mode:

bulkstats schema *schema-name* **format** *format-string* [*AOS-variable* [*AOS-variable...*]]

You can configure multiple schemas, each gathering a different type and format of data. However, you should restrict the use of multiple schemas to global data collection, and create only one schema per Frame Relay profile. Otherwise, you could apply a profile with several schemas to a large number of circuits, affecting overall system performance.

For complete information on configuring bulk statistics, see Chapter 43, "Configuring Bulk Statistics."

Display Profile Information

To display Frame Relay profile information for a single Frame Relay profile, or all Frame Relay profiles, enter the following command in operator exec mode:

show frame-relay profile [*prof-name*]

Configure LMI Parameters

For Frame Relay to operate, the Local Management Interface (LMI) must be configured. Configuring the LMI typically consists of the following tasks:

- Configure the Interface Type
- Configure the LMI Type
- Modify the Keepalive Interval
- Determine the Number of Keepalive Messages
- Set the Error Threshold
- Set the Monitored Event Count
- Set the Polling Verification Timer
- Configure LMI Auto-detect

Configure the Interface Type

A Frame Relay interface can be configured as data terminal equipment (DTE), data communications equipment (DCE), or Network to Network Interface (NNI). By default, a Frame Relay interface operates as DTE. To modify the interface type, enter the following port or HDLC channel configuration command:

frame-relay intf-type {**dce** | **dte** | **nni**}

Configure the LMI Type

By default, the SMS uses the ANSI Annex D Local Management Interface (LMI). To modify the LMI type, enter the following port or HDLC channel configuration command:

frame-relay lmi-type {**ansi** | **group-of-4** | **itu**}

Note Packet over Synchronous Optical Network (POS) ports only support the ANSI LMI type.

Modify the Keepalive Interval

By default, keepalive messages are sent every 10 seconds on a Frame Relay DTE or NNI interface. To modify this value, enter the following port or HDLC channel configuration command:

frame-relay keepalive *seconds*

Determine the Number of Keepalive Messages

By default, a Frame Relay DTE or NNI interface sends six keepalive messages before a request for a full status message is sent. To modify the number of keepalives sent before the status message request is sent, enter the following port or HDLC channel configuration command:

frame-relay lmi-n391dte *exchanges*

Set the Error Threshold

To configure the error threshold before LMI is considered to have failed on a Frame Relay DCE or NNI interface, enter the following port or HDLC channel configuration command:

frame-relay lmi-n392dce *threshold*

To configure the error threshold for DTE or NNI interfaces, enter the following port or HDLC channel configuration command:

frame-relay lmi-n392dte *threshold*

By default, the LMI is considered to have failed after three errors.

Set the Monitored Event Count

To set the monitored event count on a DCE or NNI interface, enter the following port or HDLC channel configuration command:

frame-relay lmi-n393dce *event-count*

To set the monitored event count on a DTE or NNI interface, enter the following port or HDLC channel configuration command:

frame-relay lmi-n393dte *event-count*

The default monitored event count is four.

Set the Polling Verification Timer

To configure the polling verification timer for a DCE or NNI interface, using the following port or HDLC channel configuration command:

frame-relay lmi-t392dce

Configure LMI Auto-detect

The original "group of 4" LMI uses DLCI number 1023 as the LMI PVC, while both the ANSI and ITU LMI use DLCI number 0. For a DCE interface that is not set to group-of-4 LMI type, you can configure the software to automatically detect which LMI type and use that same LMI type at the local end.

Auto-detect tells the system to look at the first LMI message received from the remote end, determine from the message the LMI type of the remote end, and reconfigure the LMI type at the local end to match. To configure the auto-detect feature, enter the following port or HDLC channel configuration command:

frame-relay auto-detect

Configure Frame Relay Circuits

Frame Relay circuits can be created on HDLC-oriented ports or channels that are configured for **encapsulation frame-relay**. Before you create any Frame Relay circuits, you must create at least one Frame Relay profile. See the "Configure a Frame Relay Profile" section for information on how to configure Frame Relay profiles.

Frame relay PVCs can be created explicitly, which means that individual PVCs are defined in the configuration file, or on-demand, which means that PVCs are created as needed, when activity is detected on a circuit.

The following sections describe how to configure Frame Relay circuits:

- Configure Explicit Frame Relay PVCs
- Configure On-Demand Frame Relay PVCs
- Configure Circuit Parameters
- Bind the Circuit

Configure Explicit Frame Relay PVCs

You can create explicit Frame Relay circuits individually, or you can create a range of Frame Relay circuits with identical characteristics. You must specify both a Frame Relay profile and an encapsulation type when you create Frame Relay circuits.

To create a Frame Relay PVC, or a range of Frame Relay PVCs, enter the following port or HDLC channel configuration command:

frame-relay pvc *dlci* [**through** *end-dlci*] **profile** *prof-name* **encapsulation** {**auto1490** | **bridge1490** | **route1490** | **dot1q** | **l2tp** | **multi** | **ppp** [**auto** | **over-ethernet**]}

When you use the **through** keyword to create a range of PVCs, AOS generates a **frame-relay pvc** command for each individual PVC in the specified range. This can result in a large configuration file. You can also create a range of explicit Frame Relay PVCs using the following command in port configuration mode:

frame-relay pvc explicit *start-dlci* **through** *end-dlci* **profile** *prof-name* **encapsulation** {**auto 1490** | **bridge 1490** | **route 1490** | **multi** | **ppp** [**auto** | **over-ethernet**]}

The **frame-relay pvc explicit** command generates a single command line in the configuration file.

Note You can use the **frame-relay pvc** command to create explicit PVCs within an on-demand range of Frame Relay PVCs; the **frame-relay pvc** command overrides the **frame-relay pvc explicit** configuration for the specified PVCs.

After you enter either of these commands, AOS enters circuit configuration mode, and you can configure the PVC or range of PVCs. If you select the dot1q keyword for the **frame-relay pvc** command, you enter dot1q encapsulation configuration mode. In this mode, you can create 802.1Q PVCs. See Chapter 19, "Configuring 802.1Q" for additional information on configuring 802.1Q features.

Configure On-Demand Frame Relay PVCs

To create a range of on-demand Frame Relay PVCs, enter the following command in port configuration mode:

frame-relay pvc on-demand *start-dlci* **through** *end-dlci* {**profile** *prof-name* **encapsulation** {**auto 1490** | **bridge 1490** | **route 1490** | **multi** | **ppp** [**auto** | **over-ethernet**]} | **aaa context** *ctx-name* [**prefix-string** *text*]}

Note You can use the **frame-relay pvc** command to create explicit PVCs within an on-demand range of Frame Relay PVCs; the **frame-relay pvc** command overrides the on-demand configuration for the specified PVCs.

After you have entered this command, AOS enters circuit configuration mode, and you can configure the range of PVCs.

Configure Circuit Parameters

You can optionally configure the following:

- Description—to provide a textual description for the circuit, enter the following command in circuit configuration mode:

  description *text*

- IP Host—You can install a permanent entry in the [system or context?] host table for a host where dynamic address resolution (ARP) is not possible or not wanted. To add a host table entry for the circuit, enter the following command in circuit configuration mode:

  ip host *ip-address* [*mac-address*]

- Mac Address—If you have configured the circuit with the PPPoE encapsulation, you can specify the source Ethernet MAC address to use for PPPoE packets sent on the circuit. To configure the MAC address, enter the following command in circuit configuration mode:

  mac address *mac-address*

Bind the Circuit

After you have configured the circuit, you must configure a binding for the circuit. See Chapter 20, "Configuring Bindings," for information on how to configure bindings.

Configuration Examples

The following example shows a complete configuration for a Frame Relay port. First, a Frame Relay profile is created:

```
[local]RedBack(config)#frame-relay profile fr-profile
[local]RedBack(config-frpro)#buffers transmit 40
[local]RedBack(config-frpro)#bulkstats schema fr-schema format "frm: %s, %u/%u bytes
rcvd: %u, bytes xmtd: %u", description, slot, port, inoctets, outoctets
[local]RedBack(config-frpro)#no counters
[local]RedBack(config-frpro)#radius attribute medium-type dsl
[local]RedBack(config-frpro)#exit
```

Next, a packet T1 port is configured for Frame Relay encapsulation. The port is configured as a DTE interface, and the appropriate LMI parameters are configured for the link:

```
[local]RedBack(config)#port ds1 3/0
[local]RedBack(config-port)#encapsulation frame-relay
[local]RedBack(config-port)#frame-relay lmi-type ansi
[local]RedBack(config-port)#frame-relay int-type dte
[local]RedBack(config-port)#frame-relay keepalive 5
[local]RedBack(config-port)#frame-relay lmi-n391dte 10
[local]RedBack(config-port)#frame-relay lmi-n392dte 2
[local]RedBack(config-port)#frame-relay lmi-n393dte 5
```

The following example shows a configuration that contains both explicit and on-demand PVCs on the port. First, the example creates an explicit range of PVCs consisting of DLCIs `100` through `600`. These PVCs use profile `fr-profile` and are configured for multiple encapsulations. Bindings are created for the PVCs. Next, a range of on-demand PVCs, consisting of DLCIs `601` through `1000` is created. These PVCs also use profile `fr-profile`, and are configured for multiple encapsulations. Bindings are created for these PVCs. Next, the example overwrites DLCI `700`. This PVC is configured to use bridged RFC 1490 encapsulation.

```
[local]RedBack(config-port)#frame-relay pvc explicit 100 through 600 profile fr-profile
encapsulation multi
[local]RedBack(config-pvc)#description Explicit PVC Range
[local]RedBack(config-pvc)#bind multi interface if1 isp1.net authentication chap
[local]RedBack(config-pvc)#frame-relay pvc on-demand 601 through 1000 profile
fr-profile encapsulation multi
```

```
[local]RedBack(config-pvc)#description On-Demand PVCs
[local]RedBack(config-pvc)#bind multi interface if1 isp1.net authentication chap
[local]RedBack(config-pvc)#frame-relay pvc 700 profile fr-profile encapsulation
bridge-1490
[local]RedBack(config-pvc)#description One Special PVC
[local]RedBack(config-pvc)#frame-relay pvc 800 through 850 profile fr-profile
encapsulation bridge-1490
[local]RedBack(config-pvc)#description Special PVC Range
[local]RedBack(config-pvc)#end
```

Configuring 802.1Q

This chapter provides an overview of the Access Operating System (AOS) support for the *802.1Q IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks* specification and describes the tasks involved in 802.1Q features through the AOS. For detailed information on the syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "802.1Q Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

The *802.1Q IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks* specification defines an architecture and bridging protocols for the partitioning of a bridged LAN into separate administratively defined sub-groups, called virtual LANs (VLANs).

The AOS software supports the following options for interoperating within an 802.1Q network:

- 802.1Q internetworking
- 802.1Q encapsulation

Each of these options is described in the following sections.

802.1Q Internetworking

The AOS provides support for internetworking between 802.1Q-tagged Ethernet ports and Asynchronous Transfer Mode (ATM) permanent virtual circuits (PVCs) using RFC 1483 bridged encapsulation or Frame Relay PVCs using RFC 1490 bridged encapsulation. This feature creates a static mapping between a specific 802.1Q tag (VLAN ID) on a specific Ethernet port and a specific ATM or Frame Relay PVC, providing a bypass for 802.1Q-tagged traffic.

802.1Q Encapsulation

The AOS supports 802.1Q encapsulation on Ethernet ports, ATM PVCs, and Frame Relay PVCs through the creation of 802.1Q PVCs on these ports and circuits. This feature provides more than the simple bypass capability provided with the 802.1Q internetworking feature.

When an 802.1Q frame comes in on an 802.1Q PVC, the 802.1Q header is stripped from the packet. The Ethernet packet is then processed normally by AOS (in other words, bridged or routed, depending on the configuration). When an outbound packet is forwarded (bridged or routed) to an 802.1Q PVC, AOS adds the appropriate 802.1Q header to the packet. This feature also allows for the channelization of Ethernet ports.

Configuration Tasks

This section describes the tasks involved in configuring the AOS 802.1Q features. See the following sections, depending on which type of 802.1Q support you prefer:

- Configure 802.1Q Internetworking
- Configure 802.1Q Encapsulation

Configure 802.1Q Internetworking

AOS provides support for internetworking between 802.1Q-tagged Ethernet ports and ATM or Frame Relay permanent virtual circuits (PVCs) using bridged RFC 1483 encapsulation for ATM, and bridged RFC 1490 encapsulation for Frame Relay. This feature allows a static mapping between a specific 802.1Q tag (VLAN ID) on a specific Ethernet port and a specific PVC. When such a mapping is configured, tagged traffic received on the Ethernet port is stripped of its tag and transmitted over the PVC using standard bridged RFC 1483 encapsulation for ATM, or standard bridged RFC 1490 encapsulation for Frame Relay. When traffic is received on the PVC, the configured tag is added before transmitting the frame on the configured Ethernet port.

Untagged frames carrying IP can be routed through an AOS context if the Ethernet port is bound to an interface in that context.

Follow these steps to configure 802.1Q internetworking between 802.1Q-tagged traffic on an Ethernet port and an ATM or Frame Relay PVC:

- Configure the Ethernet Port
- Configure the ATM or Frame Relay PVC
- Bind the ATM or Frame Relay PVC
- Display 802.1Q Internetworking Counters

Configure the Ethernet Port

See Chapter 10, “Configuring Ethernet Ports,” for information on configuring Ethernet ports. You can create a binding for the Ethernet port to handle all untagged traffic received on the Ethernet port. See Chapter 20, “Configuring Bindings,” for additional information on bindings.

Configure the ATM or Frame Relay PVC

See Chapter 17, “Configuring ATM,” for information on configuring ATM PVCs. To configure 802.1Q internetworking on an ATM PVC, you must set the encapsulation to bridged RFC 1483.

See Chapter 18, “Configuring Frame Relay,” for information on configuring Frame Relay PVCs. To configure 802.1Q internetworking on a Frame Relay PVC, you must set the encapsulation to bridged RFC 1490.

Bind the ATM or Frame Relay PVC

To bind an ATM or Frame Relay circuit to 802.1Q-tagged Ethernet frames on an Ethernet port, enter the following command in circuit configuration mode:

bind dot1q *slot/port vlan-tag-ID*

Display 802.1Q Internetworking Counters

To display 802.1Q statistics for Ethernet ports that are bound using the **bind dot1q** command in circuit configuration mode, enter the following command in operator exec mode:

show port dot1q [*slot/port*]

This command does not display statistics for 802.1Q-encapsulated circuits.

To clear 802.1Q statistics for Ethernet ports that are bound using the **bind dot1q** command in circuit configuration mode, enter the following command in administrator exec mode:

clear port dot1q [*slot/port*]

Configure 802.1Q Encapsulation

Follow these steps to configure 802.1Q encapsulation on Ethernet ports, ATM PVCs, or Frame Relay PVCs:

- Create an 802.1Q Profile
- Create an 802.1Q PVC
- Provide a Description for the 802.1Q PVC
- Bind the 802.1Q PVC
- Display 802.1Q PVC Information

Create an 802.1Q Profile

Follow these steps to configure an 802.1Q profile:

- Create an 802.1Q Profile
- Configure the P-bit setting

Create an 802.1Q Profile

An 802.1Q profile contains configuration information that is applied to all 802.1Q PVCs that reference the profile. To create a new 802.1Q profile, or modify an existing profile, enter the following command in global configuration mode:

dot1q profile *prof-name*

Configure the P-bit setting

The 802.1 packet headers contain a three-bit field called the p-bits, as specified in the *802.1P IEEE Standard for Local and Metropolitan Area Networks: Supplement to Media Access Control (MAC) Bridges: Traffic Class Expediting and Dynamic Multicast Filtering* specification. To configure the p-bit settings to be used in the packet headers for all 802.1Q PVCs that reference a particular profile, enter the following command in dot1q profile configuration mode:

pbit-setting *value*

Create an 802.1Q PVC

After you have created at least one 802.1Q profile, you can create 802.1Q PVCs on Ethernet ports or within ATM or Frame Relay PVCs.

First, you must specify the encapsulation for the Ethernet port, ATM PVC, or Frame Relay PVC as dot1q. For Ethernet ports, enter the following command in port configuration mode:

encapsulation dot1q

Note You cannot specify dot1q encapsulation or create 802.1Q PVCs on the Ethernet management port.

For an ATM PVC or Frame Relay PVC, you must select the dot1q encapsulation type when you create the PVC. To create an ATM or Frame Relay PVC, enter the appropriate command in port configuration mode:

atm pvc *vpi vci* [**through** *end-vci*] **profile** *prof-name* **encapsulation** {**auto1483** | **bridge1483** | **route1483** | **dot1q** | **l2tp** [**vc-muxed**] | **multi** | **ppp** [**auto** | **over-ethernet** | **serial** | **nlpid** | **llc** | **vc-muxed**]}

frame-relay pvc *dlci* [**through** *end-dlci*] **profile** *prof-name* **encapsulation** {**auto1490** | **bridge1490** | **route1490** | **dot1q** | **l2tp** | **multi** | **ppp** [**auto** | **over-ethernet**]}

After you have entered the appropriate command to select 802.1Q encapsulation, you can create dot1q PVCs on the Ethernet port, ATM circuit, or Frame Relay circuit. To create a dot1q PVC, enter the following command in port or circuit configuration mode:

dot1q pvc {*vlan-id* | **untagged**} **profile** *prof-name* **encapsulation** {**ipoe** | **multi** | **pppoe**}

Provide a Description for the 802.1Q PVC

You can create a textual description for the 802.1Q PVC. To do so, enter the following command in dot1q PVC configuration mode:

description *text*

Bind the 802.1Q PVC

Before traffic can flow across the dot1q PVC, you must configure a binding. See Chapter 20, "Configuring Bindings" for information on how to configure a binding.

Display 802.1Q PVC Information

To display information on 802.1Q profiles, enter the following command in operator exec mode:

show dot1q profile [*prof-name*]

To display information on configured 802.1Q PVCs, enter the following command in operator exec mode:

show dot1q pvc [**all**] [**profile** *prof-name*] [*slot/port* [**hdlc-channel** *chan-name*] [{**all** | *vpi* [**through** *end-vpi* | *vci* [**through** *end-vci*]] | *dlci* [**through** *end-dlci*]} [**dot1q-pvc** {*vlan-id* [**through** *end-vlan-id*] | **untagged**}]]] [**up** | **down** | **summary**]

To display statistics information for configured 802.1Q PVCs, enter the following command in operator exec mode:

show dot1q counters [**all**] [**profile** *prof-name*] [*slot/port* [**hdlc-channel** *chan-name*] [{**all** | *vpi* [**through** *end-vpi* | *vci* [**through** *end-vci*]] | *dlci* [**through** *end-dlci*]} [**dot1q-pvc** {*vlan-id* [**through** *end-vlan-id*] | **untagged**}]]] [**summary**]

To clear the counters for 802.1Q PVCs, enter the following command in administrator exec mode, including the **dot1q-pvc** construct:

clear port counters *slot/port* [**hdlc-channel** *chan-name*] [**pvc** {**all** | *vpi* [*vc* [**through** *end-vci*]] | *dlci* [**through** *end-dlci*]} [**dot1q-pvc** {**all** | *vlan-id* | **untagged**}]] [**-noconfirm**]

Configuration Examples

This section provides configuration examples for the AOS 802.1Q internetworking and 802.1Q encapsulation features.

802.1Q Internetworking Example

The following example shows how to configure 802.1Q-to-Bridge1483 internetworking between VLAN-ID `44` on Ethernet port `2/0` and an ATM PVC in port `4/1` with a VPI:VCI of `0:31`. The example also uses the **bind interface** command to associate untagged frames that arrive over Ethernet port `2/0` with the `local` context:

```
[local]RedBack(config)#port ethernet 2/0
[local]RedBack(config-port)#bind interface downstream local
[local]RedBack(config-port)#exit
[local]RedBack(config)#port atm 4/1
[local]RedBack(config-port)#atm pvc 0 31 profile ubr encapsulation bridge1483
[local]RedBack(config-pvc)#bind dot1q 2/0 44
```

802.1Q Encapsulation Example

The following example configures two 802.1Q PVCs on Ethernet port 2/0, and three 802.1Q PVCs on ATM port 3/0, VPI:VCI 20:20:

```
[local]RedBack(config)#dot1q profile telecommuter
[local]RedBack(config-dot1qpro)#pbit-setting 3
[local]RedBack(config)#port ethernet 2/0
[local]RedBack(config-port)#encapsulation dot1q
[local]RedBack(config-port)#dot1q pvc 27 profile telecommuter encapsulation ipoe
[local]RedBack(config-dot1q-pvc)#description bigcorp network 26
[local]RedBack(config-dot1q-pvc)#bind interface bigcorp-if local
[local]RedBack(config-dot1q-pvc)#dot1q pvc 42 profile internetsurfer encapsulation
pppoe
[local]RedBack(config-dot1q-pvc)#bind authentication pap
[local]RedBack(config-dot1q-pvc)#dot1q pvc untagged encapsulation ipoe
[local]RedBack(config-dot1q-pvc)#bind subscriber joe@local
[local]RedBack(config-dot1q-pvc)#exit
[local]RedBack(config-port)#port atm 3/0
[local]RedBack(config-port)#atm pvc 45 profile ubr encapsulation dot1q
[local]RedBack(config-pvc)#dot1q pvc 27 profile telecommuter encapsulation ipoe
[local]RedBack(config-dot1q-pvc)#description bigcorp network 26
[local]RedBack(config-dot1q-pvc)#bind interface bigcorp-if local
[local]RedBack(config-dot1q-pvc)#dot1q pvc 42 profile internetsurfer encapsulation
pppoe
[local]RedBack(config-dot1q-pvc)#bind authentication pap
[local]RedBack(config-dot1q-pvc)#dot1q pvc untagged encapsulation ipoe
[local]RedBack(config-dot1q-pvc)#bind subscriber joe@local
```

Configuring Bindings

This chapter provides an overview of Access Operating System (AOS) bindings and describes the tasks involved in configuring bindings through AOS. For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "Bind Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

Overview

Bindings form the association in the AOS between the circuits or tunnels and the higher-layer bridging, routing, and switching protocols configured for a given context. No user data can flow on a circuit or Ethernet port until some higher-layer service is configured and associated with it. Bindings are either statically mapped during configuration or dynamically created based on subscriber characteristics as defined in the local database or on a RADIUS server. Once bound, traffic flows through the context as it would through any IP router.

Throughout this chapter, the word circuit refers to ATM permanent virtual circuits (PVCs), Frame Relay PVCs, and 802.1Q PVCs.

The following sections provide tables that summarize the types of bindings supported of various port and circuit types:

- Binding Summary for ATM Circuits
- Binding Summary for Frame Relay Circuits
- Binding Summary for HDLC-Oriented Ports and Channels
- Binding Summary for Ethernet Ports
- Binding Summary for 802.1Q PVCs

Binding Summary for ATM Circuits

Table 20-1 shows the binding types that are available for the various encapsulation types permitted on ATM circuits.

Table 20-1 Binding and Encapsulation Types for ATM Circuits

	Encapsulation								
Type of Bind	Auto 1483	Bridged 1483	Routed 1483	L2TP	PPP (including vc-muxed, serial, nlpid, llc)	PPPoE	PPP Auto	Multi	dot1q
auto-subscriber	X	X	X		X				
bypass		X	X						
interface		X	X						
subscriber	X	X	X		X				
dot1q		X							
tunnel				X					
authentication					X	X	X		
session		X			X	X	X		
multi								X	

Note If you configure an ATM PVC with 802.1Q encapsulation, no **bind** commands are allowed for the ATM PVC. You must configure and bind 802.1Q permanent virtual circuits. See the "Binding Summary for 802.1Q PVCs" section.

Binding Summary for Frame Relay Circuits

Table 20-2 shows the binding types that are available for the various encapsulation types permitted on Frame Relay circuits.

Table 20-2 Binding and Encapsulation Types for Frame Relay Circuits

	Encapsulation								
Type of Bind	Auto 1490	Bridged 1490	Routed 1490	L2TP	PPP (including vc-muxed, serial, nlpid, llc)	PPPoE	PPP Auto	Multi	dot1q
auto-subscriber	X	X	X		X				
bypass		X	X						
interface		X	X						
subscriber	X	X	X		X				
dot1q		X							

Table 20-2 Binding and Encapsulation Types for Frame Relay Circuits

	Encapsulation								
Type of Bind	Auto 1490	Bridged 1490	Routed 1490	L2TP	PPP (including vc-muxed, serial, nlpid, llc)	PPPoE	PPP Auto	Multi	dot1q
tunnel				X					
authentication					X	X	X		
session		X			X	X	X		
multi								X	

Note If you configure a Frame Relay PVC with 802.1Q encapsulation, no **bind** commands are allowed for the Frame Relay PVC. You must configure and bind 802.1Q permanent virtual circuits. See the "Binding Summary for 802.1Q PVCs" section.

Binding Summary for HDLC-Oriented Ports and Channels

Table 20-3 shows the binding types that are available for the various encapsulation types permitted on HDLC-oriented ports and channels. This table applies when you do not use the default encapsulation of Frame Relay. "Channels" refers to HDLC channels on a channelized DS-3 module. HDLC-oriented ports include clear channel DS-3, HSSI, packet T1, and packet E1.

Table 20-3 Binding and Encapsulation Types for HDLC-Oriented Ports and Channels

	Encapsulation	
Type of Bind	Cisco HDLC	PPP/HDLC
bypass	X	
interface	X	
subscriber	X	X
authentication		X
session		X

Binding Summary for Ethernet Ports

Table 20-4 shows the binding types that are available for the various encapsulation types permitted on Ethernet ports.

Table 20-4 Binding and Encapsulation Types for Ethernet Ports

	Encapsulation			
Type of Bind	**IP over Ethernet**	**PPPoE**	**Multi**	**802.1Q**
bypass	X			
interface	X			
authentication		X		
session		X		
multi			X	

Note The management Ethernet port is a special case; it only supports IP over Ethernet encapsulation and static interface binding (**bind interface**).

Note If you configure an Ethernet port with 802.1Q encapsulation, no **bind** commands are allowed for the port. You must configure and bind 802.1Q permanent virtual circuits. See the "Binding Summary for 802.1Q PVCs" section.

Binding Summary for 802.1Q PVCs

Table 20-5 shows the binding types that are available for the various encapsulation types permitted on 802.1Q permanent virtual circuits (PVCs).

Table 20-5 Binding and Encapsulation Types for 802.1Q PVCs

	Encapsulation		
Type of Bind	**IP over Ethernet**	**PPPoE**	**Multi**
authentication		X	
interface	X		
multi			X
session	X	X	
subscriber	X		

Configuration Tasks

The AOS supports many types of bindings. Bindings are defined for ports, channels, or circuits. Only a single **bind** command can be configured at any time.

Perform one of the following tasks to configure a binding:

- Create a Static Binding Between a Port, Channel, or Circuit and an Interface
- Create a Static Binding Between a Port, Channel, or Circuit and a Subscriber Record
- Bind a Set of ATM or Frame Relay Circuits to Subscriber Records
- Dynamically Bind a Port or Circuit to an Authenticated Subscriber
- Create a Static Binding for PPP-Encapsulated Circuits
- Bind a Port, Channel, or Circuit to an L2TP or L2F Peer or L2TP Group
- Bind a Port, Channel, or Circuit to a Bypass
- Bind Bridge-Encapsulated ATM or Frame Relay Circuits to 802.1Q Ethernet Frames
- Display Bindings

See the "Overview" section in this chapter for a summary of the binding types supported for various port, channel, and circuit types, as well as encapsulation types.

Create a Static Binding Between a Port, Channel, or Circuit and an Interface

Static binding occurs when you bind a circuit directly to an interface. In this case, the circuit is hard-wired to the higher-layer protocols defined for the interface. This is the simplest form of binding available in AOS, providing functionality similar to that provided by traditional network devices, such as routers. You can use static bindings for any circuit with any encapsulation type.

To bind a port, circuit, or channel to a previously created interface, enter the following command in port, circuit, dot1q pvc, channel, or tunnel circuit configuration mode:

bind interface *if-name ctx-name*

Note If multiple circuits are bound to an interface, the Routing Information Protocol (RIP) is not active. RIP is only active for interfaces that are bound to an Ethernet port or bound to a single ATM PVC or Frame Relay PVC.

You must create an interface before you can create a binding to it. See Chapter 7, "Configuring Interfaces" for additional information on how to configure an interface.

Create a Static Binding Between a Port, Channel, or Circuit and a Subscriber Record

You can also statically bind a circuit to a particular subscriber in a given context. In this case, the binding between the circuit and the higher-layer protocols is determined indirectly, through the subscriber record. The subscriber record can be defined locally, or on a RADIUS server.

To bind a port, circuit, or channel to a subscriber record, enter the following command in port, circuit, dot1q pvc, or channel configuration mode:

bind subscriber *sub-name* [**password** *password*]

Bind a Set of ATM or Frame Relay Circuits to Subscriber Records

When configuring ATM PVCs using RFC 1483 encapsulation to use RADIUS, a subscriber name must be associated with each circuit via configuration. Although you can enter these names manually, AOS supports a quick method for preallocating a collection of ATM PVCs with contiguous virtual circuit identifiers (VCIs), or a collection of Frame Relay PVCs with contiguous data link connection identifiers (DLCIs), and automatically generating subscriber names and optional passwords.

To configure AOS to automatically generate the required subscriber records and the bind subscriber commands for a range of PVCs, enter the following command in circuit configuration mode:

bind auto-subscriber *prefix1 ctx-name* [**password** *prefix2*]

The following commands create five PVCs, each bound through an automatically generated subscriber name. The string following the keyword **auto-subscriber** can be any arbitrary string and is used as the leading characters in the subscriber names.

```
[local]RedBack(config)#port atm 2/0
[local]RedBack(config-port)#atm pvc 0 100 through 105 profile adsl
encapsulation route1483
[local]RedBack(config-pvc)#bind auto-subscriber green local
```

The following lines are entered into the system configuration as a result of entering the previous commands:

```
port atm 3/0
 atm pvc 0 100 profile adsl encapsulation route1483
  bind subscriber green2.0.0.100@local
 atm pvc 0 101 profile adsl encapsulation route1483
  bind subscriber green2.0.0.101@local
 atm pvc 0 102 profile adsl encapsulation route1483
  bind subscriber green2.0.0.102@local
 atm pvc 0 103 profile adsl encapsulation route1483
  bind subscriber green2.0.0.103@local
 atm pvc 0 104 profile adsl encapsulation route1483
  bind subscriber green2.0.0.104@local
 atm pvc 0 105 profile adsl encapsulation route1483
  bind subscriber green2.0.0.105@local
```

The subscriber names are of the form *<string><slot>.<port>.<vpi>.<vci>,* and are automatically generated.

Dynamically Bind a Port or Circuit to an Authenticated Subscriber

Dynamic binding occurs when you bind a circuit to the higher-layer protocols based on session information. For example, a Point-to-Point Protocol (PPP) encapsulated session can be bound to a particular context and interface by examining the authenticated structured username in the form: *user@context*.

Dynamic binding is the key to enabling advanced features such as dynamic service and provider selection. Dynamic binding also enables simultaneous access to multiple services on a single circuit.

To create a dynamic, implicit binding locally through a subscriber record, or remotely through a RADIUS record, enter the following command in port, circuit, dot1q pvc, or channel configuration mode:

bind authentication {**pap** | **chap** [**wait**] | **chap pap** [**wait**]} [**maximum** *sessions*] [**context** *ctx-name* | **service-group** *svc-name*]

A bind authentication command creates a dynamic, implicit binding locally through a subscriber record or remotely through a RADIUS record.

The command keyword **chap** indicates that the Challenge Handshake Authentication Protocol (CHAP) is used. Other options are possible (see the *Access Operating System (AOS) Command Reference* publication for a complete list of choices). CHAP uses a challenge/response protocol to provide authentication without sending cleartext passwords over the network. In addition to authenticating subscribers to the Subscriber Management System (SMS) device, CHAP allows the SMS device to be authenticated to subscribers. To authenticate the SMS device to a subscriber, an **outbound password** must be configured in that subscriber's record.

Note If authentication is being done remotely using RADIUS, the local subscriber records are replaced by the corresponding subscriber records in the RADIUS database. For further information on RADIUS, see Chapter 41, "Configuring RADIUS."

The string configured with the **password** command must match the password string sent by the remote PPP user to the SMS device. The **outbound password** command configures the password string AOS sends to the remote PPP user. The Password Authentication Protocol (PAP) does not require an outbound password.

In the case of CHAP, the passwords referred to are actually shared secret keys used by the various systems to compute and verify cryptographic checksums in response to their peer's challenge. To the command-line interface (CLI), however, these values are entered identically to the way PAP passwords are entered. The keyword **password** is used in all cases.

Note The system hostname is used by the AOS as the username string for all outbound PPP authentication.

An IP address is also required. This IP address is assigned to the remote end of the PPP link. If the authentication procedure is successful, the PPP link is established and the ATM PVC is implicitly bound to the interface whose address mask includes the address of the remote PPP end point.

Note If no such interface exists, then the bind fails. That is, there must be an interface whose address/mask range includes the address assigned to a subscriber during the IP Control Protocol (IPCP) phase of PPP (or that includes the address that has been statically configured for the subscriber). This has implications for RADIUS servers too, in that they must return addresses for subscribers that fall within the range of an interface configured in the appropriate context.

If the remote PPP device is a router (or the remote segment of any other encapsulation type contains a router), it might be necessary to configure one or more static routes whenever the link is brought up. This is accomplished by one or more RIP configuration commands in the subscriber record.

AOS also supports the notion of a default subscriber record. If a **subscriber default** record exists, the information in that record automatically becomes a part of every other subscriber record in the context. For example, to configure the system to supply a primary Domain Name System (DNS) address to every PPP subscriber in the current context (see RFC 1877, *PPP Internet Protocol Control Protocol Extensions for Name Server Addresses*), the following commands would be used:

```
[local]RedBack(config-ctx)#subscriber default
[local]RedBack(config-sub)#dns primary 10.10.1.1
```

Although it is possible to place the **dns** command used in this example in every individual subscriber record, the default subscriber record can greatly simplify configuration files.

Note If you modify a subscriber record for a subscriber that is already bound, you must use the **clear subscriber** command in administrator exec mode for the changes to take effect. The subscriber session is ended and restarted with the new parameters. This is true regardless of whether subscriber records are configured locally or in RADIUS.

Create a Static Binding for PPP-Encapsulated Circuits

Static binding allows an administrator to "hardwire" a PPP-encapsulated PVC to a specific context; in other words, this feature denies the end-user the ability to dynamically select a context (service). To configure a static binding for a circuit, use the **context** option for the following command in circuit configuration mode:

bind authentication {**pap** | **chap** [**wait**] | **chap pap** [**wait**]} [**maximum** *sessions*] [**context** *ctx-name* | **service-group** *svc-name*]

Unlike the **bind subscriber** command for PPP circuits, this feature requires authentication of the subscriber session for the PPP session to come up.

Note When using global authentication, the Context-Name attribute returned by RADIUS must be identical to the context specified on the **bind authentication** command line; otherwise, the binding fails.

Service access lists provide a way to create more complex rules to determine which contexts, domains, and tunnels should be available to subscribers on a per-circuit basis. See Chapter 39, "Configuring Service Access Lists."

Bind Multiple Encapsulations on a Single Circuit or Port

You can enable both IP over Ethernet encapsulation (Bridged RFC 1483, Bridged RFC 1490, or Ethernet) and PPPoE encapsulation to be specified on the same ATM or Frame Relay circuit or Ethernet port. Additionally, each of the two encapsulations can be separately bound. The **multi** encapsulation must be specified for these circuits or ports using the **atm pvc**, **frame-relay pvc**, or **encapsulation** command.

To specify the bindings for the multiple encapsulations on the multi-encapsulated circuit, enter the following command in port, circuit, or dot1q pvc configuration mode:

bind multi {**interface** *if-name ctx-name* | **subscriber** *sub-name*} **authentication** {**pap** | **chap** [**wait**] | **chap pap** [**wait**]} [**maximum** *sessions*] [**context** *ctx-name* | **service-group** *svc-name*]

Use the **interface** or **subscriber** constructs to configure the static binding information for the IP over Ethernet portion of the circuit.

The **interface** construct specifies the interface and context to which the IP over Ethernet portion of the circuit is bound. You must create an interface before you can create a binding to it. See Chapter 7, "Configuring Interfaces" for additional information on how to configure an interface.

The **subscriber** construct specifies the subscriber name to which the IP over Ethernet portion of the circuit is bound. The subscriber record can be defined locally, or on a RADIUS server.

The **authentication** construct specifies the authentication protocol for the PPPoE portion of the circuit. Optionally, you can also specify a maximum number of sessions for the PPPoE portion as well as context or service access list restrictions.

Bind a Port, Channel, or Circuit to an L2TP or L2F Peer or L2TP Group

To bind a port or circuit to a tunnel peer, enter the following command in port, circuit, dot1q pvc, or channel configuration mode as appropriate:

bind session *peer-name ctx-name* [**maximum** *sessions*]

where *peer-name* is the name of the peer to which the circuit or port is to be bound and *context* is the context in which that peer exists. If you are binding an Ethernet port, the bind session command puts the port into "promiscuous mode," which means that it ignores MAC addresses and tunnels everything to the LNS. The concept of promiscuous mode is implicit for RFC 1483 bridged and RFC 1490 bridged encapsulated circuits.

Both the name of the peer and the context must be specified. In this command, a domain name for the peer can be used for the *peer-name* argument. Dynamic tunnel selection is not available for L2F. The **bind session** command creates a hard-wired binding to a specific peer in a specific context.

See Chapter 25, "Configuring L2TP," for information on configuring L2TP and Chapter 26, "Configuring L2F," for additional information on configuring L2F.

Bind a Port, Channel, or Circuit to a Bypass

Bypasses allow a network administrator to bind two circuits together without protocol translation. The SMS device simply relays link-layer frames between the two circuits without interpretation of the higher-layer protocols.

A bypass has at most two ports or circuits bound to it, and the ports or circuits must be configured with the same encapsulation type. AOS also supports bindings between ATM and Frame Relay PVCs, as long as both PVCs use bridged encapsulation or both PVCs use routed encapsulation (RFC 1483 and RFC 1490).

To configure a binding to a bypass, enter the following command in port, circuit, or channel configuration mode, as appropriate:

bind bypass *bypass-name ctx-name*

Note This command does not apply to 802.1Q permanent virtual circuits (PVCs).

You must create the bypass before you can create a binding to it. See Chapter 22, "Configuring Bypasses," for additional information on how to configure a bypass.

Bind Bridge-Encapsulated ATM or Frame Relay Circuits to 802.1Q Ethernet Frames

AOS provides support for internetworking between 802.1Q-tagged Ethernet ports and ATM or Frame Relay permanent virtual circuits (PVCs) using bridged RFC 1483 encapsulation for ATM, and bridged RFC 1490 encapsulation for Frame Relay. See Chapter 19, "Configuring 802.1Q." for a detailed description of this feature.

To bind an ATM or Frame Relay PVC to 802.1Q Ethernet frames, enter the following command in circuit configuration mode:

bind dot1q *slot/port vlan-tag-ID*

Display Bindings

To display the configured bindings on your system, enter the following command in operator exec mode:

show bindings [**all**] [**bound** | **unbound**] [*slot/port* [**hdlc-channel** *chan-name* [*dlci* [**through** *end-dlci*]]] | [*vpi* [*vci* [**through** *end-vci*]]]] [**auth** | **bypass** [*bypass-name*] | **dot1q** | **interface** [*if-name*] | **none** | **session** [*peer-name*] | **subscriber** [*sub-name*] | **summary** | **tunnel** [*tunnel-peer*] | **multi**]

If you are logged in to the local context, this command shows all bindings on the system. If you are logged into any other context, this command only displays bindings in the context to which you are logged in.

The various keywords and arguments help to restrict the output to particular bindings of interest. For details on the command syntax, see the *Access Operating System (AOS) Command Reference* publication.

Configuration Examples

This section provides several configuration examples that show how to configure the bindings discussed in the "Configuration Tasks" section. The examples include the following:

- Interface Binding
- Multiple Encapsulations Binding
- Static Bindings for PPP-Encapsulated Circuits
- Binding an L2TP Tunnel over an ATM PVC
- Bypass Binding

Interface Binding

The following is an example of a static binding, for the management Ethernet port on an SMS 1800:

```
[local]RedBack#configure
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#interface mgmt
[local]RedBack(config-if)#ip address 1.2.3.4 255.255.255.0
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#exit
[local]RedBack(config)#port ethernet 0/0
[local]RedBack(config-port)#bind interface mgmt local
```

Multiple Encapsulations Binding

The next example shows an example of binding multiple encapsulations on a single ATM circuit. The IP over Ethernet traffic is bound to interface `corpa` in the local context, and the PPPoE traffic is bound through the results of **pap** authentication, with a limit of five simultaneous PPPoE sessions:

```
[local]RedBack#configure
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#interface corpa
[local]RedBack(config-if)#ip address 1.2.3.4 255.255.255.0
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#exit
[local]RedBack(config)#port atm 3/0
[local]RedBack(config-port)#atm pvc 10 100 profile ubr encapsulation multi
[local]RedBack(config-pvc)#bind multi interface corpa local authentication pap
maximum 5
```

The following commands configure an inbound password, an outbound password, and an IP address in the subscriber record named `pppuser` in the local context:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#subscriber name pppuser
[local]RedBack(config-sub)#password in-test
[local]RedBack(config-sub)#outbound password out-test
[local]RedBack(config-sub)#ip address 10.1.3.30
```

Static Bindings for PPP-Encapsulated Circuits

In the example that follows, the PPP-encapsulated PVC is constrained to be bound only in the context `isp.net`:

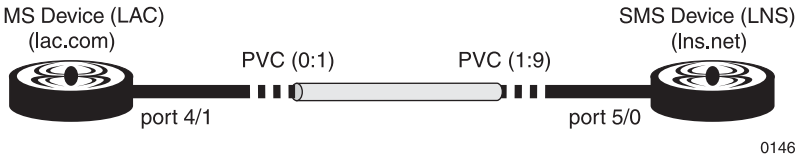
```
[local]RedBack(config)#port atm 4/0
[local]RedBack(config-port)#atm pvc 0 1 profile ubr encapsulation ppp
[local]RedBack(config-pvc)#bind authentication pap context isp.net
```

Binding an L2TP Tunnel over an ATM PVC

This section provides an example of a LAC and an LNS connected directly by an ATM PVC.

Figure 20-1 shows an example of `lac.com` connected to `lns.net` by an ATM PVC. Here we omit details such as tunnel authenticators and concentrate on the configuration of the ATM PVC, the tunnel, and the binding of the PVC to the tunnel.

Figure 20-1 L2TP Tunnel over ATM PVC



The following commands configure the LAC side; first, the tunnel itself:

```
[local]RedBack(config)#system hostname lac.com
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#l2tp-peer name lns.net media pvc
```

Next, we configure the PVC and bind it to the specified tunnel. The encapsulation on the circuit connecting the tunnel peers is `l2tp`:

```
[local]RedBack(config)#port atm 4/1
[local]RedBack(config-port)#atm pvc 0 1 profile ubr encapsulation l2tp
[local]RedBack(config-pvc)#bind l2tp-tunnel lns.net local
```

The configuration for the LNS side is similar. First, the tunnel:

```
[local]RedBack(config)#system hostname lns.net
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#l2tp-peer name lac.com media pvc
```

And finally, the circuit:

```
[local]RedBack(config)#port atm 5/0
[local]RedBack(config-port)#atm pvc 1 9 profile ubr encapsulation l2tp
[local]RedBack(config-pvc)#bind l2tp-tunnel lac.com local
```

Bypass Binding

The following example associates two ATM PVCs with a bypass named `swoosh`. Both circuits have identical encapsulations (RFC 1483 bridged).

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#bypass swoosh
[local]RedBack(config-bypass)#description For bigisp.net
[local]RedBack(config-bypass)#exit
[local]RedBack(config)#port atm 3/0
[local]RedBack(config-port)#atm pvc 1 1 profile ubr_pro encapsulation bridge1483
[local]RedBack(config-pvc)#bind bypass swoosh local
[local]RedBack(config-pvc)#atm pvc 1 2 profile ubr_pro encapsulation bridge1483
[local]RedBack(config-pvc)#bind bypass swoosh local
```

802.1Q Binding

The following example shows how to configure 802.1Q-to-RFC 1483 bridged internetworking between VLAN-ID 44 on Ethernet port `2/0` and an ATM PVC in port `4/1` with a VPI:VCI of `0:31`. The example also uses the **bind interface** command to associate untagged frames that arrive over Ethernet port `2/0` with the `local` context:

```
[local]RedBack(config)#port ethernet 2/0
[local]RedBack(config-port)#bind interface downstream local
[local]RedBack(config-port)#exit
[local]RedBack(config)#port atm 4/1
[local]RedBack(config-port)#atm pvc 0 31 profile ubr encapsulation bridge1483
[local]RedBack(config-pvc)#bind dot1q 2/0 44
```

Part 4

Bridges and Bypasses

Configuring Bridging

This chapter provides an overview of bridging and describes the tasks involved in configuring bridging features through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "Bridging Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks and Examples

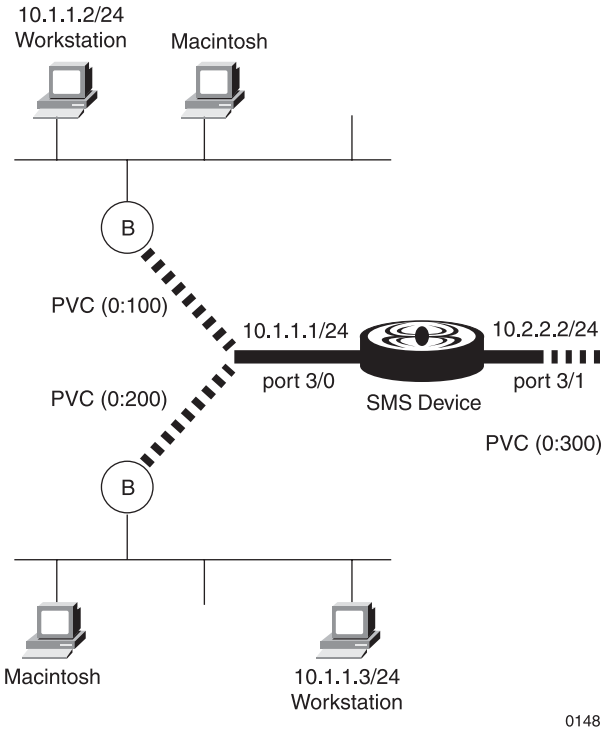
Overview

This section discusses the configuration of bridging in the AOS. The AOS supports two flavors of bridging: bridging-routing and bridging-only. The following are some basic features of the bridging implementation in the AOS:

- Bridges are context-specific and cannot span multiple contexts.
- The bridge instance can be a Media Access Control (MAC)-based (transparent) or IEEE 802.1D Spanning-Tree Protocol bridging.
- Multiple bridges can exist in a context.
- Bridges can be bridging-only or they can be bridging-routing bridges. Bridging-routing bridges are designed to bridge all protocols, except IP, which they route.
- A bridge instance, by default, is a bridging-routing bridge.
- A bridge-only bridge cannot be bound to an interface configured with an IP address.
- Only circuits that support a MAC layer can be part of a bridge group. These include Ethernet ports, Asynchronous Transfer Mode (ATM) permanent virtual circuits (PVCs) with RFC 1483 bridged encapsulation, and Frame Relay PVCs with RFC 1490 bridged encapsulation. Specifically, Point-to-Point Protocol (PPP)-encapsulated circuits cannot be members of a bridge group.
- Once a bridge has a circuit or an interface attached to it, its type is immutable. For example, to change the bridge type from a bridging-only bridge to a bridging-routing bridge, all circuits and interfaces must first be unbound.
- A subscriber record can contain either a bridge group or an IP address, but not both.

Figure 21-1 shows a simple configuration of an SMS device providing transparent bridging between two network segments. This topology can provide a transparent LAN service for all protocols, or, for only protocols other than IP, wherein IP is routed rather than bridged. The rest of this section describes these implementations in more detail.

Figure 21-1 Bridging Between Two Network Segments



Configuration Tasks and Examples

First, configure the SMS device in a bridging-routing configuration and then in a bridging-only configuration. In the bridging-routing example, the SMS device is configured to route IP packets and to bridge all other packets. For example, the AppleTalk packets sent between the two Macintosh machines, each connected to a separate, physical, Ethernet segment, are bridged by the SMS 1000 device.

To configure the bridge, first create a bridge group and specify its parameters. The following commands create the bridge group (ensure that the bridge group is not bridging-only):

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#bridge CO_WORKERS
[local]RedBack(config-bridge)#exit
```

Because our intention is to configure bridging-routing, create IP interfaces for the selected context's router, provide IP addresses, and enable secured-Address Resolution Protocol (ARP). In the case of bridging-routing (as opposed to the case of routing-only), the example specifies the name of the bridge-group to be bound to the desired interface within that context:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#interface BRIDGED_INTERFACE
[local]RedBack(config-if)#ip address 10.1.1.1 255.255.255.0
[local]RedBack(config-if)#ip arp arpa
[local]RedBack(config-if)#ip secured-arp
[local]RedBack(config-if)#bridge-group CO_WORKERS
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#interface ROUTED_ONLY_INTERFACE
[local]RedBack(config-if)#ip address 10.2.2.2 255.255.255.0
```

Next, create subscriber records to be used in the configuration and binding of the virtual circuits that are to be bridged. In particular, the following example ensures that the IP address assigned to each bridged subscriber is one that causes the subscriber to be bound to the BRIDGED_INTERFACE created in the previous example:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#subscriber name SW_GURU
[local]RedBack(config-sub)#ip address 10.1.1.2
[local]RedBack(config-sub)#exit
[local]RedBack(config-ctx)#subscriber name HW_GURU
[local]RedBack(config-sub)#ip address 10.1.1.3
```

Finally, create the circuits and bind our subscribers to those circuits. Assume that the ATM shaping profile ATM_PROFILE already exists. Also, note that the encapsulation on each ATM circuit is RFC 1483 bridged:

```
[local]RedBack(config)#port atm 3/0
[local]RedBack(config-port)#atm pvc 0 100 profile ATM_PROFILE encapsulation bridge1483
[local]RedBack(config-pvc)#bind subscriber SW_GURU@local
[local]RedBack(config-pvc)#exit
[local]RedBack(config-port)#atm pvc 0 200 profile ATM_PROFILE encapsulation bridge1483
[local]RedBack(config-pvc)#bind subscriber HW_GURU@local
```

For completeness, create a PVC for the right-hand side of Figure 21-1 and bind it to the indicated interface as follows:

```
[local]RedBack(config)#port atm 3/1
[local]RedBack(config-port)#atm pvc 0 300 profile ATM_PROFILE encapsulation route1483
[local]RedBack(config-pvc)#bind interface ROUTED_ONLY_INTERFACE local
```

To turn off IP routing, that is, to turn the BRIDGED_INTERFACE into a bridge-only interface in which all link-layer frames (including those carrying IP packets) are bridged instead of routed, use the **bridge-only** command within the bridge definition and configure the desired bridge group within each subscriber record.

Disassociate the IP address from the interface BRIDGED_INTERFACE. The following commands configure this:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#interface BRIDGED_INTERFACE
[local]RedBack(config-if)#no ip address 10.1.1.1 255.255.255.0
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#bridge CO_WORKERS
[local]RedBack(config-bridge)#bridge-only
[local]RedBack(config-bridge)#subscriber name SW_GURU
[local]RedBack(config-sub)#bridge-group CO_WORKERS
[local]RedBack(config-sub)#exit
[local]RedBack(config-ctx)#subscriber name HW_GURU
[local]RedBack(config-sub)#bridge-group CO_WORKERS
[local]RedBack(config-sub)#exit
[local]RedBack(config-ctx)#port atm 3/0
[local]RedBack(config-port)#atm pvc 0 100 profile ATM_PROFILE encapsulation bridge1483
[local]RedBack(config-pvc)#bind subscriber SW_GURU@local
[local]RedBack(config-pvc)#exit
[local]RedBack(config-port)#atm pvc 0 200 profile ATM_PROFILE encapsulation bridge1483
[local]RedBack(config-pvc)#bind subscriber HW_GURU@local
[local]RedBack(config-pvc)#exit
```

To bind an Ethernet port to a bridge-only bridge, configure an interface that has the preferred bridge group and no IP address, and then bind the port to the interface. The following example shows these steps:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#interface ether20
[local]RedBack(config-if)#bridge-group CO_WORKERS
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#port ether 2/0
[local]RedBack(config-port)#bind interface ether20 local
```

Note The SMS device is not reachable through Simple Network Management Protocol (SNMP) or Telnet via circuits or ports configured to be part of a bridging-only bridge. These packets are bridged through—as are all other packets.

Enable Station Move Logging

You can enable the logging of station moves detected by the system. A large number of station move messages could indicate a problem in the network configuration. This command applies to all bridge groups on the system.

To enable station move logging, enter the following command in global configuration mode:

bridge station-move verbose

Configuring Bypasses

This chapter describes the tasks related to configuring bypasses. For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "Bypass Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

Bypasses allow a network administrator to bind two circuits together without protocol translation. The Subscriber Management System (SMS) device simply relays link-layer frames between the two circuits without interpretation of the higher-layer protocols. The two circuits being bound to a bypass can reside on the same port or on different ports.

A bypass consists of two circuits, usually of the same type. For example, two Ethernet ports can be bound together in a bypass as can two Asynchronous Transfer Mode (ATM) permanent virtual circuits (PVCs) or two Frame Relay PVCs. A Frame Relay PVC can only be bound to an ATM PVC if both PVCs use bridged encapsulation or both PVCs use routed encapsulation. The AOS can automatically convert between RFC 1483 and RFC 1490-routed encapsulation and RFC 1483 and RFC 1490-bridged encapsulation.

Note Point-to-Point Protocol (PPP) encapsulation is not supported for bypasses. No element being bound to a bypass can have PPP encapsulation.

Bypasses are context-specific. When you bind two ports or circuits together in a bypass, you must know the context in which the bypass exists.

Once two PVCs are bound together, all incoming traffic from one PVC is sent out the other PVC, regardless of content. This means, for example, that if a ping is received on one PVC, the local system does not respond to it. Instead, the local system sends it out the other PVC like all other traffic.

Configuration Tasks

To configure a bypass between two circuits, perform the tasks in the following sections:

- Create the Bypass and Enter Bypass Configuration Mode
- Create a Description for the Bypass
- Bind Two Circuits to the Bypass

Create the Bypass and Enter Bypass Configuration Mode

To create a bypass and enter bypass configuration mode, enter the following command in context configuration mode:

bypass *bypass-name*

The new bypass is created in the context, and you enter bypass configuration mode.

Create a Description for the Bypass

To create a description for the bypass, enter the following command in bypass configuration mode:

description *text*

The *text* argument is an alphanumeric string (including spaces) that provides descriptive information about the bypass. The description appears in the output of the **show bypass** and **show configuration** commands. If you ever want to change the description, simply create a new one, and it overwrites the existing one.

Bind Two Circuits to the Bypass

To bind circuits to the bypass, enter the following command in circuit configuration, High-Speed Data Link Control (HDLC) channel configuration mode, or port configuration mode:

bind bypass *bypass-name ctx-name*

The *bypass-name* argument is the name of a configured bypass and the *ctx-name* argument is the context in which that bypass exists. This command is not valid in port configuration mode or HDLC channel configuration mode for ports or channels with Frame Relay or PPP encapsulation.

You must create the bypass before you can use the name of the bypass in a **bind bypass** command. Only two circuits can be bound to one bypass. See the "Bind Commands" chapter in the *Access Operating System (AOS) Command Reference* publication for a full description of this command.

Configuration Examples

The following example associates two ATM PVCs with a bypass named swoosh. Both circuits have identical encapsulations (RFC 1483 bridged).

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#bypass swoosh
[local]RedBack(config-bypass)#description For bigisp.net
[local]RedBack(config-bypass)#exit
[local]RedBack(config)#port atm 3/0
[local]RedBack(config-port)#atm pvc 1 1 profile ubr_pro encapsulation bridge1483
[local]RedBack(config-pvc)#bind bypass swoosh local
[local]RedBack(config-pvc)#atm pvc 1 2 profile ubr_pro encapsulation bridge1483
[local]RedBack(config-pvc)#bind bypass swoosh local
```

Part 5

Point-to-Point Protocol

Configuring PPP and PPPoE

This chapter provides an overview of Point-to-Point Protocol (PPP) and PPP over Ethernet (PPPoE), and describes the tasks involved in configuring PPP and PPPoE features through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands mentioned in this chapter, see the "PPP and PPPoE Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

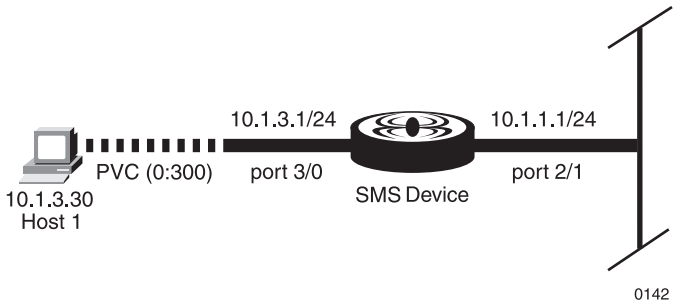
- Configuring PPP
- Configuring PPP over Ethernet

Configuring PPP

The AOS supports PPP over Asynchronous Transfer Mode (ATM) as described in RFC 2364, *PPP Over AAL5*—both virtual circuit multiplexed (VC muxed) and Logical Link Control (LLC) encapsulations, and PPP over Frame Relay as possible encapsulation types. PPP circuits, unlike the RFC 1483 or Ethernet circuits described in earlier sections, can be dynamically bound to an interface. Although you can use a **bind interface** or **bind subscriber** command as previously described to directly or indirectly bind a PPP over ATM (or Frame Relay) permanent virtual circuit (PVC) to an interface, you can use the **bind authentication** command to dynamically bind the PVC to an interface on the basis of authentication. If you use a **bind interface** or **bind subscriber** command, the PPP link is brought up unauthenticated.

In Figure 23-1, the host on the left is configured to run PPP over ATM. In this example, a Subscriber Management System (SMS) 1000 is configured to dynamically bind this user to an IP interface assumed to be previously configured with an IP address of 10.1.3.1 and a mask of 255.255.255.0.

Figure 23-1 Two-Port ATM-to-Ethernet Network (PPP)



The following commands create the appropriate ATM PVC, and indicate to the system that binding of the PVC is to be accomplished through an authentication process. We assume that the ATM profile `adsl` already exists:

```
[local]RedBack(config)#port atm 3/0
[local]RedBack(config-port)#atm pvc 0 300 profile adsl encapsulation ppp
[local]RedBack(config-pvc)#bind authentication chap
```

The **chap** keyword indicates that the Challenge Handshake Authentication Protocol (CHAP) is to be used. Other options are possible (see the *Access Operating System (AOS) Command Reference* publication for a complete list of choices). CHAP uses a challenge/response protocol to provide authentication without sending cleartext passwords over the network. In addition to authenticating subscribers to the SMS device, CHAP allows the device to be authenticated to subscribers. To authenticate the device to a subscriber, an **outbound password** command must be configured in that subscriber's record.

Note If authentication is being done remotely using Remote Authentication Dial-In User Service (RADIUS), the local subscriber records are replaced by the corresponding subscriber records in the RADIUS database. For further information on RADIUS, see Chapter 41, "Configuring RADIUS."

The string configured with the **password** command must match the password string sent by the remote end of the PPP link to the SMS device. The **outbound password** command configures the password string AOS sends to the remote end of the PPP link. The Password Authentication Protocol (PAP) does not require an outbound password.

In the case of CHAP, the passwords referred to are actually shared secret keys used by the various systems to compute and verify cryptographic checksums in response to their peer's challenge. To the command-line interface (CLI), however, these values are entered identically to the way PAP passwords are entered. The **password** keyword is used in all cases.

Note The system hostname is used by the SMS device as the username string for all outbound PPP authentication.

An IP address is also required. This IP address is assigned to the remote end of the PPP link. If the authentication procedure is successful, the PPP link is established and the ATM PVC is implicitly bound to the interface whose address mask includes the address of the remote PPP end point.

Note If no such interface exists, and if a default PPP interface has not been configured (see the "Default PPP Interface" section), then the bind fails. That is, there must be an interface whose address/mask range includes the address assigned to a subscriber during the IP Control Protocol (IPCP) phase of PPP (or that includes the address that has been statically configured for the subscriber). This has implications for RADIUS servers too, in that they must return addresses for subscribers that fall within the range of an interface configured in the appropriate context.

The following commands configure an inbound password, an outbound password, and an IP address in the subscriber record named `pppuser` in the local context:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#subscriber name pppuser
[local]RedBack(config-sub)#password in-test
[local]RedBack(config-sub)#outbound password out-test
[local]RedBack(config-sub)#ip address 10.1.3.30
```

If the remote PPP device is a router (or the remote segment of any other encapsulation type contains a router), it may be necessary to configure one or more static routes whenever the link is brought up. This is accomplished by one or more Routing Information Protocol (RIP) configuration commands in the subscriber record.

The AOS also supports the concept of a default subscriber record. If a default subscriber record (created with the **subscriber default** command) exists, the information in that record automatically becomes a part of every other subscriber record in the context. For example, to configure the system to supply a primary Domain Name System (DNS) address to every PPP subscriber in the current context (see RFC 1877, *PPP Internet Protocol Control Protocol Extensions for Name Server Addresses*), enter the following commands:

```
[local]RedBack(config-ctx)#subscriber default
[local]RedBack(config-sub)#dns primary 10.10.1.1
```

Although it is possible to place the **dns** command used in this example in every individual subscriber record, the default subscriber record can greatly simplify configuration files. Attributes specified in the default subscriber record are superseded by values specified in individual subscriber records.

Note If you modify a subscriber record for a subscriber that is already bound, you must use the **clear subscriber** command for the changes to take effect. The subscriber session is terminated and restarted with the new parameters. This is true regardless of whether subscriber records are configured locally or via RADIUS.

Static Binding for PPP-Encapsulated Circuits

Static binding allows an administrator to hard-wire a PPP-encapsulated PVC to a specific context; in other words, this feature denies the subscriber the ability to dynamically select a context (service). To configure a static binding for a circuit, use the optional **context** *ctx-name* construct in the following channel, circuit, or port configuration command:

bind authentication {**pap** | **chap** [**wait**] | **chap pap** [**wait**]} [**maximum** *sessions*] [**context** *ctx-name* | **service-group** *group-name*]

Unlike the **bind subscriber** command for PPP circuits, this feature requires authentication of the subscriber session for the PPP session to come up.

The following example constrains the PPP-encapsulated PVC to be bound only in the `isp.net` context:

```
[local]RedBack(config)#port atm 4/0
[local]RedBack(config-port)#atm pvc 0 1 profile ubr encapsulation ppp
[local]RedBack(config-pvc)#bind authentication pap context isp.net
```

Note When using global authentication, the Context-Name attribute returned by RADIUS must be identical to the context specified on the **bind authentication** command line; otherwise, the binding fails.

Service access lists provide a way to create more complex rules to determine which contexts, domains, and tunnels should be available to subscribers on a per-circuit basis. See Chapter 39, "Configuring Service Access Lists," for more information.

PPP Oversubscription

Ordinarily, any **bind authentication** command would cause the subscriber to be counted toward the maximum number of bind authentications allowed, whether or not the subscriber is active. The alternative is to configure the system to operate in passive mode, which means that only active PPP sessions count toward the maximum number of bind authentications. The effect is that the number of bind authentications you can have is increased, beyond the number that could actually bind and come up.

In passive mode, no PPP structures are allocated unless or until a peer initiates a session. Once established, the subscriber is considered a bind authentication in terms of the maximum subscribers that are allowed. When a peer ends a session, that subscriber is no longer counted and the associated PPP structures are deallocated.

In the default mode, PPP structures are allocated for every bind authentication at the time the circuit is configured. It may not be necessary to use passive mode in circumstances where every bind authentication is active. With passive mode set, the peers must always initiate their sessions; in other words, the SMS device never initiates sessions, even to reestablish disconnected sessions. This is not the case when PPP passive mode is disabled.

Passive mode does not affect the maximum number of subscribers that can be terminated in a particular context (established by the **aaa max subscribers** command) or the hard limits allowed by the SMS device.

The following example configures the system to operate in passive mode:

```
[local]RedBack(config)#ppp passive
```

The following example disables passive mode operation:

```
[local]RedBack(config)#default ppp passive
```

PPP Idle and Absolute Timeout

You can configure idle timeouts and absolute timeouts for subscriber PPP sessions. An idle timeout causes a session to be terminated if there is no activity on that session for the configurable timeout interval. An absolute timeout causes a session to be terminated after a configurable interval, regardless of whether any activity occurs on that session. Counters on the PPP circuit must be turned on before you configure idle timeouts.

Note Keepalive messages are considered traffic for purposes of measuring idle time.

The following example configures a subscriber named `roger` in the `corp.com` context to have a maximum PPP session time of 120 minutes (2 hours):

```
[local]RedBack(config)#context corp.com
[local]RedBack(config-ctx)#subscriber name roger
[local]RedBack(config-sub)#timeout absolute 120
```

PPP idle and absolute timeouts can be served by RADIUS, as is the case of all subscriber attributes. See Appendix C, "RADIUS Attributes," for a description of all standard and vendor-specific RADIUS attributes supported by the AOS software. See the "Subscriber Commands" chapter in the *Access Operating System (AOS) Command Reference* publication for the **timeout** command syntax description and usage guidelines.

Default PPP Interface

Ordinarily, PPP sessions that attempt to come up and cannot bind to a valid interface simply fail. A PPP default interface acts as a fall back for those incoming PPP connections. If a PPP session is established, and there is no valid interface to which it can bind, the session binds to the default interface. The default interface is a virtual interface; there is no actual outgoing circuit. Therefore, a proxy is necessary. One or more interfaces that are not the default interface are set up as proxies using the **ip ppp-proxy-arp** command. This command enables proxy Address Resolution Protocol (ARP) functionality on behalf of PPP circuits that are bound to the PPP default interface.

To create a default PPP interface, enter the following command in context configuration mode:

interface *if-name* **ppp-default**

The **interface** *if-name* **ppp-default** command takes you into interface configuration mode where you can enter commands to configure the PPP default interface. Only a subset of the interface configuration commands are available for interfaces designated as PPP default interfaces. The following interface configuration commands are available for PPP default interfaces:

description

ip access-group

ip address

ip igmp

ip mtu

Note This subset of interface configuration commands does not allow for Dynamic Host Control Protocol (DHCP) relay enabling, Address Resolution Protocol (ARP) enabling, secured ARP enabling, setting the SMS device to look at the host table first when selecting the next-hop interface, configuring IP address pools, or RIP interface configuration. These features are not compatible with the functionality of a PPP default interface. See Chapter 7, "Configuring Interfaces" for descriptions of these excluded features. The **ppp-proxy-arp** command is also not available because an interface cannot serve as both a PPP default and a PPP default proxy.

To configure an interface to serve as a proxy for the default PPP interface, enter the following command in interface configuration mode for the proxy interface:

ip ppp-proxy-arp

Multilink Point-to-Point Protocol

Multilink PPP (MP) is an extension to PPP that allows a peer to use more than one physical link for communication. A good example is an ISDN connection from a home that actually uses two physical links to connect to the Service Provider. When using more than one physical link to connect two peers, you need a mechanism to load balance the connection across the two (or more) links in the bundle. MP is used to fragment the datagrams and send them across the multiple links in the bundle in a way that achieves optimum use of the media.

Both ends of the point-to-point links must be capable of supporting MP connections. The two ends configure the data link by swapping Link Control Protocol (LCP) packets during a link establishment phase. If MP is not successfully negotiated by the two ends of the link, MP is not enabled for the connection.

MP is most frequently used with an SMS device for dial aggregation because dial sessions may have multiple PPP links via ISDN or Windows dial-up networking with multiple analog lines.

To configure an SMS device to be capable of negotiating MP, perform the tasks described in the following sections:

- Enable MP Negotiation
- Change Default Endpoint Discriminator Class and Value
- Change the Default MRRU for LCP Negotiation
- Limit the Number of Concurrent Multilink Sessions
- Display Multilink State and Statistics
- Enable PPP Multilink Debugging

Enable MP Negotiation

To enable MP negotiation, enter the following command in global configuration mode:

ppp multilink enable

This allows other MP-related commands to be entered, commands that are not available when MP is disabled. The default condition is for MP to be disabled, so entering this command is required when you want to use MP.

Change Default Endpoint Discriminator Class and Value

You can change the class and value used for endpoint discriminator negotiation from the default settings, although it should not normally be necessary to do so. To change the settings, enter the following command in global configuration mode:

ppp multilink endpoint-discriminator {**class-1** *text* | **class-2** *ip-address* | **class-3** *mac-address* | **class-5** *text* | **local-ip-address** | **local-mac-address**}

A Class 1 endpoint discriminator (**class-1** keyword) is a locally assigned address. The *text* argument is a string of up to 20 characters.

A Class 2 endpoint discriminator (**class-2** keyword) is an IP address. The *ip-address* argument is the specific address you want to use. If you want a Class 2 endpoint discriminator that uses the IP address of the management port, use the **local-ip-address** keyword instead.

A Class 3 endpoint discriminator (**class-3** keyword) is a Media Access Control (MAC) address in the format hh:hh:hh:hh:hh:hh, where hh is a hexadecimal number. The *mac-address* argument is the specific address you want to use. Do not use this option to select a locally assigned MAC address; use the **class-1** *text* construct instead. If you want a Class 3 endpoint discriminator that uses the MAC address of the management port, use the **local-mac-address** keyword.

A Class 5 endpoint discriminator (**class-5** keyword) is a public switched network directory number. The *text* argument is a string of up to 15 characters representing an E.164 international telephone directory number.

Change the Default MRRU for LCP Negotiation

By default, the size of the maximum received reconstructed unit (MRRU) for LCP is 1,500 bytes. This value is usually satisfactory, but you can change it if necessary. To change the maximum size of information fields of reassembled packets, enter the following command in global configuration mode:

ppp multilink mrru *bytes*

Limit the Number of Concurrent Multilink Sessions

To place a limit on the number of concurrent multilink sessions that subscribers can use, enter the following command in subscriber configuration mode:

port-limit *value*

The *value* argument is the number of PPP links to which the subscriber is entitled.

You can use this command to set a port limit for the default subscriber record or for individual subscriber records. It can be useful to limit ISDN users, for example, to the two PPP links that ISDN provides for them. It can also be useful for preventing a single user's account from being accessed by multiple users at the same time.

The RADIUS Port-Limit attribute is supported for purposes of setting a port limit remotely via RADIUS. See Appendix C, "RADIUS Attributes," for detailed information on the Port-Limit attribute.

Display Multilink State and Statistics

To display multilink state and statistics information, enter the following command in administrator exec mode:

show ppp multilink [**all** | **bundle** *bundle-id* | **summary**]

Enable PPP Multilink Debugging

To enable logging of MP-related debugging messages, enter the following command in operator exec mode:

debug ppp multilink

PPP Compression

PPP compression results in link efficiency by substantially reducing the size of many PPP packets. Compression is achieved at the expense of increased packet processing, and is, therefore, not always beneficial. For example, in the case of narrowband access, data is generally already compressed, making this step redundant and needlessly expensive in terms of system resources.

SMS devices support two types of PPP compression, Microsoft Point-to-Point Compression (MPPC) and Stac Lempel-Ziv-Stac (Stac LZS). In either case, PPP compression must be negotiated with the peer. MPPC compression is negotiated first, followed by Stac LZS if necessary. MPPC is generally used for Microsoft Windows clients. Stac LZS is generally used for other clients, such as Macintosh.

PPP compression on an SMS device is a hardware-assist method, meaning that the appropriate hardware is required. You must have an IPSec/Compression Transform Engine (TE) module installed in your SMS device to configure and use the PPP compression feature.

To enable PPP compression, enter the following command in subscriber configuration mode:

ppp compression

To display PPP compression information, enter the following command in operator exec mode:

show ppp compression [*slot*/*port* [**counters** | **summary**] | **all** [**counters** | **summary**]] [**subscriber** *sub-name*]

Configuring PPP over Ethernet

The Redback Networks implementation of PPP over Ethernet (PPPoE) supports the following:

- PPPoE encapsulation on RFC 1483 (ATM) bridged circuits, RFC 1490 (Frame Relay) bridged circuits, and physical Ethernet ports.
- Both IP over Ethernet encapsulation (RFC 1483 bridged, RFC 1490 bridged, or Ethernet) and PPPoE encapsulation on the same ATM or Frame Relay circuit or Ethernet port. Each of the two encapsulations can be separately bound. You must specify the **multi** encapsulation for these circuits or ports using the **atm pvc**, **frame-relay pvc**, or **encapsulation** command.
- Policing and rate-limiting on a per-PPP-session basis.

- Ability to configure a maximum number of concurrent sessions allowed on a bridge-encapsulated circuit, a physical Ethernet port, or both.
- Multiple simultaneous PPPoE sessions arriving over the same circuit while being bound to different services (contexts).
- Ability to advertise a list of services (domains) to a client during the discovery protocol.
- Ability to send messages to subscribers including messages of the minute (MOTMs).
- Ability to direct the subscriber's browser to open on a specific, optionally customized URL.

PPPoE for Bridge-Encapsulated ATM and Frame Relay Circuits

Configuring RFC 1483 bridged (ATM) or RFC 1490 bridged (Frame Relay) circuits for PPPoE encapsulation is similar to configuring PPP over ATM or PPP over Frame Relay circuits. The two differences are:

1. The encapsulation is set to PPP over Ethernet in both cases. In the case of ATM, this implies PPP sessions being carried by RFC 1483 bridged circuits. Thus, the encapsulation is PPP/Ethernet/SNAP/LLC/AAL5/ATM. Below the PPP layer is standard RFC 1483 bridged encapsulation. The Frame Relay case is analogous.
2. Like PPP-encapsulated circuits, PPPoE circuits allow dynamic service selection through the **bind authentication** command. PPPoE circuits, unlike PPP over ATM (RFC 2364, *PPP Over AAL5*) and PPP over Frame Relay (RFC 1973, *PPP in Frame Relay*) circuits, allow the administrator to specify a maximum number of concurrent sessions allowed for that circuit. RFC 2364 and RFC 1973 circuits are limited to a single PPP session per circuit.

To configure a RFC 1483 bridged or RFC 1490 bridged circuit for PPPoE encapsulation, use the **encapsulation ppp over-ethernet** command when configuring the PVC. The following example configures a RFC 1483 bridged circuit for PPPoE encapsulation and limits the number of concurrent PPP sessions for that circuit to a maximum of two:

```
[local]RedBack(config)#port atm 4/0
[local]RedBack(config-port)#atm pvc 0 1 profile ubr encapsulation ppp over-ethernet
[local]RedBack(config-pvc)#bind authentication chap maximum 2
```

Similarly, for Frame Relay, we have the following:

```
[local]RedBack(config)#port ds3 7/1
[local]RedBack(config-port)#frame-relay pvc 17 profile fast encapsulation ppp
over-ethernet
[local]RedBack(config-pvc)#bind authentication pap
```

PPPoE for Physical Ethernet Ports

Setting up a physical Ethernet port to carry PPPoE sessions requires the following:

1. Specifying the encapsulation type through the **encapsulation** Ethernet port configuration mode command.
2. Binding the PPPoE-encapsulated port through authentication.

The following commands configure the indicated Ethernet port for PPPoE encapsulation and limit the maximum, concurrent, PPP sessions for that port to a total of 200:

```
[local]RedBack(config)#port ethernet 6/0
[local]RedBack(config-port)#encapsulation ppp over-ethernet
[local]RedBack(config-port)#bind authentication chap pap maximum 200
```

Configuring Routes for Multiple PPPoE Sessions

You can configure the SMS device to provide different routes for different PPPoE sessions. For each session, routes are sent in a PPPoE Active Discovery Network (PADN) and installed on the subscriber's machine. In this way, subscribers are enabled with seamless client route provisioning on a per PPPoE session basis. The subscriber's PC client must support PADN.

To configure routes for multiple PPPoE sessions, enter the following command in subscriber configuration mode:

pppoe client route *ip-address netmask metric*

The *ip-address* argument is the address of the destination host, the *netmask* argument is the network mask for the route entry, and the *metric* argument is the cost (in number of hops) to the destination.

Advertising a List of Services

You can specify a list of services (domains) advertised to a client during the PPPoE discovery protocol. You can configure the AOS to advertise all domains (services) available in the SMS device; you can also select on a domain-by-domain basis whether the service is to be advertised.

Note Domain names, not context names, are advertised in PPPoE discovery.

The following example configures an SMS device to advertise all of its domains (isp1, isp2, and isp3) in PPPoE discovery:

```
[local]RedBack(config)#context isp1.net
[local]RedBack(config-ctx)#domain isp1
[local]RedBack(config-ctx)#exit
[local]RedBack(config)#context isp2.net
[local]RedBack(config-ctx)#domain isp2
[local]RedBack(config-ctx)#exit
[local]RedBack(config)#context isp3.net
[local]RedBack(config-ctx)#domain isp3
[local]RedBack(config-ctx)#exit
[local]RedBack(config)#pppoe services all-domains
```

The next example configures an SMS device to advertise only the indicated domains, namely, isp1 and isp2. Domains corp1 and corp2 are not advertised, because the **advertise** keyword is not specified in the definitions of the two domains, and the **marked-domains** keyword is specified in the **pppoe services** command:

```
[local]RedBack(config)#context isp1.net
[local]RedBack(config-ctx)#domain isp1 advertise
[local]RedBack(config-ctx)#exit
```

```
[local]RedBack(config)#context isp2.net
[local]RedBack(config-ctx)#domain isp2 advertise
[local]RedBack(config-ctx)#exit
[local]RedBack(config)#context corp1.com
[local]RedBack(config-ctx)#domain corp1
[local]RedBack(config-ctx)#exit
[local]RedBack(config)#context corp2.com
[local]RedBack(config-ctx)#domain corp2
[local]RedBack(config-ctx)#exit
[local]RedBack(config)#pppoe services marked-domains
```

Sending MOTMs to Subscribers

Messages of the minute (MOTMs) are messages displayed to subscribers when their PPPoE sessions are established and they have been authenticated. MOTMs are sent to subscribers in a PPP Active Discovery Message (PADM) packet. Only one MOTM can be active at a time and it can be no more than 256 characters in length. MOTMs are typically used to inform subscribers about planned system downtime, new available services, and other notices of potential interest to all subscribers.

Enter the **pppoe motm** command in subscriber configuration mode to create or delete an MOTM. The following example creates an MOTM:

```
[local]RedBack(config-sub)#pppoe motm System down 0400 today for scheduled maintenance
```

The following example replaces the first MOTM with a new one:

```
[local]RedBack(config-sub)#pppoe motm Scheduled maintenance canceled for 08/29/2001.
```

The following example removes the existing MOTM so that no message is sent to subscribers:

```
[local]RedBack(config-sub)#no pppoe motm
```

Note If you are using Layer 2 Tunneling Protocol (L2TP) tunneling and PPPoE subscribers are authenticated in the L2TP Network Server (LNS), the tunnel must be configured for Ethernet over L2TP for MOTMs to work. See "Configuring Ethernet over L2TP" in Chapter 25, "Configuring L2TP." Then, configure the MOTM information in the LNS under the subscriber profile. The PPPoE user on the L2TP Access Concentrator (LAC) side is not then able to do service selection. They are always tunneled to the designated LNS by the **bind session** command.

Pointing the Subscriber's Browser to a URL

You can also use PADMs to instruct subscribers' browsers to open on a particular URL when the session is established. A set of special-character sequences is available for purposes of customizing this URL for individual subscribers. The SMS device expands these sequences prior to inclusion in the PADM.

Table 23-1 lists the special-character sequences you can include in the URL.

Table 23-1 Special Sequences for Use in PADMs

Sequence	Description
%U	The entire subscriber name used in PPP authentication.

Table 23-1 Special Sequences for Use in PADMs

Sequence	Description
%u	The user portion of the subscriber name used in PPP authentication. If there is no separator character, distinguishing between a user portion and a domain portion, then %u expands to the entire subscriber name.
%d	The domain portion of the subscriber name used in PPP authentication. If there is no separator character, distinguishing between a user portion and a domain portion, %d expands to a zero length string.
%D	The name of the context to which the subscriber was authenticated. This may be different than the domain portion of the subscriber name.
%%	Expands to a single % character.

The **pppoe url** command is configured in each subscriber record or in the subscriber default.

For subscriber joe@local, the following example would cause a PADM containing the URL http://www.loe.com/members/joe@local to be sent to the PPPoE client when the PPP session is established:

```
[local]RedBack(config-ctx)#subscriber name joe
[local]RedBack(config-sub)#pppoe url http://www.loe.com/members/%U
```

The next example uses the **pppoe url** command to configure the subscriber default. For every subscriber to which the subscriber default is applied, a PADM containing http://www.loe.com/members/*name* is sent to the PPPoE client when the PPP session is established:

```
[local]RedBack(config-ctx)#subscriber default
[local]RedBack(config-sub)#pppoe url http://www.loe.com/members/%u
```

Note If you are using L2TP tunneling and PPPoE subscribers are authenticated in the LNS, the tunnel must be configured for Ethernet over L2TP for the **pppoe url** command to work. See "Configuring Ethernet over L2TP" in Chapter 25, "Configuring L2TP." Then, configure the URL information in the LNS under the subscriber profile. The PPPoE user on the LAC side is not then able to do service selection. They are always tunneled to the designated LNS by the **bind session** command.

Part 6

Tunnels

Configuring GRE

This chapter provides an overview of generic routing encapsulation (GRE) over IP Version 4 (IPv4) tunnels and the GRE Virtual Private Network (VPN) model, describes the tasks used to configure GRE through the Access Operating System (AOS), and provides GRE configuration examples.

For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "GRE Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

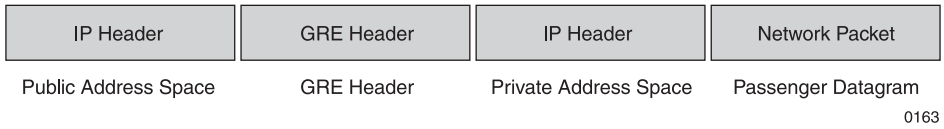
- Overview
- Configuration Tasks
- Configuration Examples

Overview

GRE is a simple, stateless protocol that allows for the tunneling of IP in IP. One of the more common applications of GRE tunneling is the use of VPNs to connect remote sites using private IP addresses via a public network using publicly routable IP addresses.

In our model, the GRE tunnel is defined in a context connected to the public network, while other contexts on the same SMS device act as VPNs, each with their own separate IP address space. IP packets going through the tunnel from the VPN are encapsulated with an IP header from the public address space as illustrated in Figure 24-1.

Figure 24-1 GRE Tunnel Packet Encapsulation



You can configure GRE tunnels via Remote Authentication Dial-In User Service (RADIUS), and a Subscriber Management System (SMS) device can be placed in server (or listen) mode to allow for on-demand autoconfiguration of GRE tunnels. You can also configure GRE tunnels statically.

Redback's implementation of GRE over IPv4 is based on these IETF documents:

- RFC 1702, *Generic Routing Encapsulation over IPv4 Networks*
- RFC 2784, *Generic Routing Encapsulation*
- RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*

You can use GRE tunnels in conjunction with AOS contexts to provide an IP-based VPN service. A single tunnel is created between a pair of edge SMS devices, and tunnel keys demultiplex traffic into AOS contexts. A context acts as a dedicated virtual router for each VPN, where the IP address space (for example, private addresses as described in RFC 1918, *Address Allocation for Private Internets*) and routing databases are maintained separately from other contexts. Each tunnel key appears as a point-to-point circuit connection. Like any other circuit on an SMS device, the tunnel key can be bound to an interface in any context. Using GRE, an arbitrary network topology can be overlaid on the physical topology; that is, each VPN can have a topology independent of the topology to which the physical SMS device is connected. To facilitate IP connectivity between VPNs on different SMS devices over GRE, several options exist:

- For simple topologies, static routes in each VPN context can be used.
- For hub-and-spoke and dual hub-and-spoke topologies, a combination of static routing and Routing Information Protocol (RIP) can be used. Each spoke VPN is configured with a static default route to the GRE tunnel attached to the hub site, and is configured via RIP to disseminate downstream prefixes to the hub. Each hub VPN is configured to run RIP in passive mode to listen for prefixes from spoke routers.
- For more complex topologies, each VPN can run its own instance of a routing protocol, such as RIP or Open Shortest Path First (OSPF).

Configuration Tasks

To configure GRE, perform the tasks described in the following sections:

- Configure GRE Tunneling Statically
- Enable GRE Configuration Via RADIUS
- Configure GRE Server Mode
- Clear and Reset GRE Parameters
- Display GRE Information

Configure GRE Tunneling Statically

To configure GRE tunneling statically, perform the following steps:

1. To configure GRE tunnel parameters, enter the following command in context configuration mode:

   gre-peer name *peer-name* **remote** *ip-address* **local** *ip-address*

   This command also causes the AOS to switch to GRE peer configuration mode. The remote IP address at one end of a GRE tunnel is the local IP address of the other end of the tunnel and vice versa.

2. Optionally, in GRE peer configuration mode, you can:
   - Provide a description of the GRE tunnel by entering the **description** command.
   - Verify the integrity of each packet by entering the **checksum** command.
   - Limit the aggregate packet stream received over a GRE tunnel by entering the **police** command.
   - Limit the aggregate packet stream sent over a GRE tunnel by entering the **rate-limit** command.

   Modifications to checksum and rate limitations on incoming and outgoing traffic for an established GRE tunnel do not take effect until you clear the tunnel using the **clear gre-peer** command.
3. To enable the GRE tunnel and switch to tunnel map configuration mode, enter the following command in global configuration mode:

   tunnel map
4. To create the GRE tunnel circuit, enter the following command in tunnel map configuration mode:

   gre-tunnel *tun-name ctx-name* [**key** *key-id*] [**server**]

   This command also causes the AOS to switch to tunnel circuit configuration mode. Enter the name of the GRE tunnel (created with the **gre-peer** command in step 1) and the context in which the tunnel was created. A key can optionally be provided to allow for the sharing of the same tunnel configuration between multiple VPNs. If no key is provided, the GRE encapsulation header does not contain a key field.

   The **server** keyword causes the tunnel circuit to behave as the server side of a connection. See the "RADIUS Considerations" subsection for more details. If the **server** keyword is specified for a particular tunnel, all keys for that tunnel must also be configured with the **server** keyword, or tunnel creation fails. The remote and local IP addresses specified in the **gre-peer** command are swapped.
5. To bind the GRE tunnel circuit to an interface, enter the following command in tunnel circuit configuration mode:

   bind interface *if-name ctx-name*

   A GRE tunnel, which acts like a virtual circuit, must be bound to an interface. Use the *if-name* and *ctx-name* arguments to identify the interface and the context in which the interface resides.
6. To resolve the remote end of the tunnel with an IP address, enter the following command in tunnel circuit configuration mode:

   ip host *ip-address*

Enable GRE Configuration Via RADIUS

As an alternative to explicitly using command-line interface (CLI) commands, tunnel configuration can be stored in a central RADIUS database. Our model can use RADIUS attributes defined in RFC 2868 and a set of vendor specific attributes (VSAs) for GRE configuration. The RADIUS attributes are distributed between two records, the tunnel record and the tunnel key record. The tunnel record corresponds to parameters configured via the **gre-peer** command, is based on RFC 2868, and contains the configuration parameters for the entire tunnel, such as the local and remote endpoints and the media type. The tunnel key record contains VSAs and correspond to key-specific information configured via the **gre-tunnel** command.

The tunnel attributes are indexed on the RADIUS server through the tunnel name, while the tunnel key attributes are indexed by the concatenation of the client IP address, the server IP address, and the key. For example, using the following command, the system uses the name SMS-BeverlyHills to retrieve the tunnel attributes such as local and remote IP address and checksum:

```
gre-tunnel SMS-BeverlyHills key 210
```

Next, the system constructs the name for the tunnel-key record using the local and remote IP addresses and key and retrieves the tunnel key attributes. In the example, if the tunnel `SMS-BeverlyHills` has as remote endpoint IP address `2.2.2.2` and local endpoint IP address `1.1.1.1`, the system uses the name `GRE-1.1.1.1:2.2.2.2:210` to query RADIUS for the tunnel key record.

Perform the following steps to enable GRE to download configuration parameters from RADIUS:

1. To configure AAA to use RADIUS for GRE, enter the following command in context configuration mode:

   aaa authorization gre radius

2. To enter tunnel map configuration mode, enter the following command in global configuration mode:

   tunnel map

3. To configure a GRE tunnel circuit, enter the following command in tunnel map configuration mode:

   gre-tunnel *tun-name ctx-name* [**key** *key-id*] [**server**]

   This command triggers the system to query RADIUS for the GRE configuration parameters. If, however, a **bind** command is entered in tunnel circuit configuration mode, the system uses the local configuration instead of the binding information provided in the RADIUS response.

   The **server** keyword causes the tunnel circuit to behave as the server side of a connection. See the "RADIUS Considerations" subsection for more details. If the **server** keyword is specified for a particular tunnel, all keys for that tunnel must also be configured with the **server** keyword, or tunnel creation fails.

Configure GRE Server Mode

You can place an SMS device into server mode for GRE tunnels. When in GRE server mode, the SMS device listens for incoming GRE traffic and creates tunnels on demand as authorized by RADIUS. A typical application of the server-side router is to act as a hub for client-spoke routers.

To configure GRE server mode, perform the following configuration tasks:

1. To configure AAA to use RADIUS for GRE, enter the following command in context configuration mode:

   aaa authorization gre radius

2. To enter tunnel map configuration mode, enter the following command in global configuration mode:

   tunnel map

3. To configure GRE autodetect, enter the following command in tunnel map configuration mode:

   gre-circuit creation on-demand aaa [**context** *ctx-name*]

   This command also places you in GRE creation configuration mode. The **gre-circuit creation** command establishes listen mode for new tunnels and keys. The optional **context** *ctx-name* construct specifies the context in which the AAA parameters for GRE are configured.

   When a packet for an unknown GRE tunnel, key, or both is received by an SMS device that has been configured for GRE autodetect, the AOS constructs a name for the tunnel key record using the source address of the IP packet as the client address and the destination address as the server address. The key is obtained from the GRE header in the received packet. If no key is present, the key component does not become part of the name. This name is then used to query RADIUS for the tunnel key attributes.

4. Optionally, in GRE creation configuration mode, you can enable reverse path forwarding (RPF) check to determine whether the source IP address of a particular GRE packet is reachable via the interface through which the packet entered the system. This method provides some protection against spoofing attacks. If a packet fails RPF check, the tunnel is not created.

   To enable GRE RPF checking, enter the following command in GRE creation configuration mode:

   gre-rpf-check

RADIUS Considerations

GRE can use RADIUS attributes described in RFC 2868 and proprietary Redback VSAs. RADIUS attributes are split into tunnel and tunnel key sets, and each attribute in each set has a client and server version. The server attributes are primarily used for on-demand creation of tunnels, while the client attributes are primarily used by explicitly-configured tunnels.

To optimize server access to RADIUS, tunnel attributes are duplicated in every key record of a tunnel, allowing a GRE server to bring up tunnels via a single RADIUS query. Duplicated attributes are ignored if a tunnel is configured using the **gre-tunnel** command. If at any point a RADIUS query fails, or if consistency checks fail for the attributes, configuration of the tunnel key circuit fails and a message is recorded in the system log.

GRE tunnel attributes are listed in Appendix C, “RADIUS Attributes.” Table 24-1 lists a few of the common attributes found in a GRE tunnel record and their CLI equivalents.

Table 24-1 Common Tunnel Record Attributes

Attribute	CLI Equivalent
Tunnel-Client-Endpoint	**gre-peer** command using the **local** *ip-address* construct
Tunnel-Server-Endpoint	**gre-peer** command using the **remote** *ip-address* construct
Tunnel-Checksum	**checksum** command
Rate-Limit-Rate	**rate-limit** command
Rate-Limit-Burst	**rate-limit** command
Police-Rate	**police** command
Police-Burst	**police** command

In server mode, the system uses the rate limit attributes as the police attributes and the police attributes as the rate limit attributes.

The tunnel key record contains the attributes listed in Table 24-2. When checksum, rate limit, or police parameters are configured in the tunnel record, the attributes listed in Table 24-2 must also exist in the tunnel key record.

Table 24-2 Common Tunnel Key Record Attributes

Attribute	Description
Tunnel-Client-VPN	**gre-tunnel** command, client side
Tunnel-Server-VPN	**gre-tunnel** command, server side
Tunnel-Client-Int-Addr	**bind interface** command, client side
Tunnel-Server-Int-Addr	**bind interface** command, server side
Tunnel-Client-RHost	**ip host** command, client side
Tunnel-Server-RHost	**ip host** command, server side

You can omit the Tunnel-Client-RHost and Tunnel-Server-RHost attributes. If these attributes are not present, the server uses the Tunnel-Client-Int-Addr attribute in place of Tunnel-Server-RHost, and the client uses Tunnel-Server-Int-Addr as the Tunnel-Client-RHost. Similarly, the Tunnel-Server-VPN attribute is also optional; if omitted, the server uses the Tunnel-Client-VPN attribute.

If the server keyword is specified via the **gre-tunnel** command, the system treats all client attributes as server attributes. The system also treats all server attributes as client attributes when appropriate. This allows the SMS devices at the two ends of a tunnel to share the same RADIUS database while explicitly configured via the **gre-tunnel** command. In this scheme, one end of the tunnel is configured using the **server** keyword, forcing it to pick up the server attributes.

In the following sample RADIUS record, the **server** keyword is specified via the **gre-tunnel** command:

```
Vpn2 password = redback
Tunnel-Client-Endpoint = 1.1.1.1
Tunnel-Server-Endpoint = 2.2.2.2
```

Instead of using the Tunnel-Client-Endpoint attribute as the local IP address for the tunnel, the system uses Tunnel-Server-Endpoint. Similarly, the Tunnel-Client-Endpoint is used as the remote IP address of the tunnel. When the **gre-tunnel** command is used, the system queries RADIUS to retrieve the tunnel and tunnel key records in two passes, regardless of whether the **server** keyword is specified with the **gre-tunnel** command.

Clear and Reset GRE Parameters

To trigger the AOS to reread and reapply new parameters to the GRE tunnel, enter the following command in operator exec mode:

clear gre peer *peer-name* [**key** *key-id* | **all**]

Changes to an existing active tunnel are applied when the tunnel is cleared. You can clear a specific key on the tunnel by using the **key** keyword, or you can clear all keys on the tunnel by using the **all** keyword. If no key is specified, the system uses the tunnel with the key as specified through the gre-tunnel command. A key with a value of 0 is not allowed.

Note If you issue the **clear gre-peer** command while keys within the tunnel are waiting for AAA to learn bind information, the request may time out according to preset rules governing AAA RADIUS request-and-retrieval interaction. When such timeouts occur, existing parameters are retained and used.

Display GRE Information

To display the status and configuration for a particular key or for all keys of a GRE tunnel, enter the following command in operator exec mode:

show gre info peer *peer-name* [**key** *key-id* | **all**]

To display statistics for a particular key or all keys of a GRE tunnel, enter the following command in operator exec mode:

show gre counters peer *peer-name* [**key** *key-id* | **all**]

To display the status and configuration for one or all GRE tunnels in the context, enter the following command in operator exec mode:

show gre tunnel info [**peer** *peer-name*]

To display statistics for one or all GRE tunnels in the context, enter the following command in operator exec mode:

show gre tunnel counters [**peer** *peer-name*]

Configuration Examples

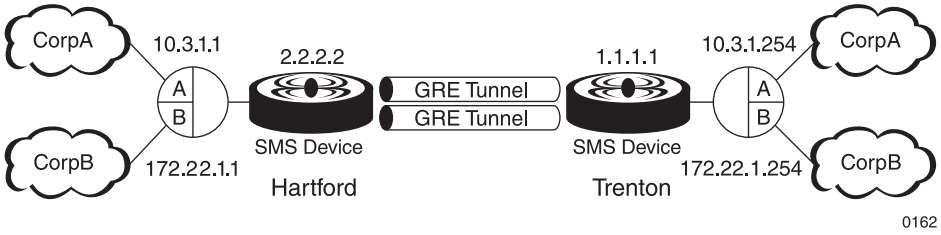
This section provides the following GRE configuration examples:

- Basic GRE Configuration
- Back-to-Back Tunnel Configuration Using RADIUS
- GRE Server

Basic GRE Configuration

Figure 24-2 shows a basic GRE configuration with two VPNs sharing a single tunnel:

Figure 24-2 GRE Tunneling Example



The SMS device in Hartford is configured as follows:

```
[local]Hartford(config)#context local
[local]Hartford(config-ctx)#gre-peer name toTrenton remote 1.1.1.1 local 2.2.2.2
[local]Hartford(config)#context CorpA
[local]Hartford(config-ctx)#interface tunnel0
[local]Hartford(config-if)#ip address 10.3.1.1 255.255.255.0
[local]Hartford(config)#tunnel map
[local]Hartford(config-tunnel)#gre-tunnel toTrenton local key 370
[local]Hartford(config-tun-circuit)#bind int tunnel0 CorpA
[local]Hartford(config-tun-circuit)#ip host 10.3.1.254
.
.
.
[local]Hartford(config-gre)#context CorpB
[local]Hartford(config-ctx)#interface tunnel0
[local]Hartford(config-if)#ip address 172.22.1.1 255.255.255.0
[local]Hartford(config)#tunnel map
[local]Hartford(config-tunnel)#gre-tunnel toTrenton local key 1205
[local]Hartford(config-tun-circuit)#bind int tunnel0 CorpB
[local]Hartford(config-tun-circuit)#ip host 172.22.1.254
```

The SMS device in Trenton is configured as follows:

```
[local]Trenton(config)#context local
[local]Trenton(config-ctx)#gre-peer name toHartford remote 2.2.2.2 local 1.1.1.1
[local]Trenton(config)#context CorpA
[local]Trenton(config-ctx)#interface tunnel0
[local]Trenton(config-if)#ip address 10.3.1.254 255.255.255.0
[local]Trenton(config)#tunnel map
[local]Trenton(config-tunnel)#gre-tunnel toHartford local key 370
[local]Trenton(config-tun-circuit)#bind int tunnel0 CorpA
[local]Trenton(config-tun-circuit)#ip host 10.3.1.1
.
.
.
[local]Trenton(config-gre)#context CorpB
[local]Trenton(config-ctx)#interface tunnel0
```

```
[local]Trenton(config-if)#ip address 172.22.1.254 255.255.255.0
[local]Trenton(config)#tunnel map
[local]Trenton(config-tunnel)#gre-tunnel toHartford local
[local]Trenton(config-tun-circuit)#bind int tunnel0 CorpB key 1205
[local]Trenton(config-tun-circuit)#ip host 172.22.1.1
```

Back-to-Back Tunnel Configuration Using RADIUS

The topology for the following example is identical to that of the previous section, "Basic GRE Configuration."

The following attributes are configured in the RADIUS server used by the two SMS devices:

```
Hartford-Trenton password = redback
  Tunnel-Medium-Type = 1 (IP)
  Tunnel-Type = 10 (GRE)
  Tunnel-Client-Endpoint = 2.2.2.2
  Tunnel-Server-Endpoint = 1.1.1.1
GRE-2.2.2.2:1.1.1.1:370 password = redback
  Tunnel-Client-VPN = CorpA
  Tunnel-Server-VPN = CorpA
  Tunnel-Client-Int-Addr = 10.3.1.1
  Tunnel-Server-Int-Addr = 10.3.1.254
  Tunnel-Client-RHost = "10.3.1.254"
  Tunnel-Server-RHost = "10.3.1.1"
GRE-2.2.2.2:1.1.1.1:1205 password = redback
  Tunnel-Client-VPN = CorpB
  Tunnel-Server-VPN = CorpB
  Tunnel-Client-Int-Addr = 172.22.1.1
  Tunnel-Server-Int-Addr = 172.22.1.254
  Tunnel-Client-RHost = "172.22.1.254"
  Tunnel-Server-RHost = "172.22.1.1"
```

The SMS device in Hartford is configured as follows:

```
[local]Hartford(config)#context local
[local]Hartford(config-ctx)#aaa authentication gre radius
[local]Hartford(config)#context CorpA
[local]Hartford(config-ctx)#interface tunnel0
[local]Hartford(config-if)#ip address 10.3.1.1 255.255.255.0
[local]Hartford(config)#context CorpB
[local]Hartford(config-ctx)#interface tunnel0
[local]Hartford(config-if)#ip address 172.22.1.1 255.255.255.0
[local]Hartford(config)#tunnel map
[local]Hartford(config-tunnel)#gre-tunnel Hartford-Trenton local key 370
[local]Hartford(config-tun-circuit)#gre-tunnel Hartford-Trenton local key 1205
```

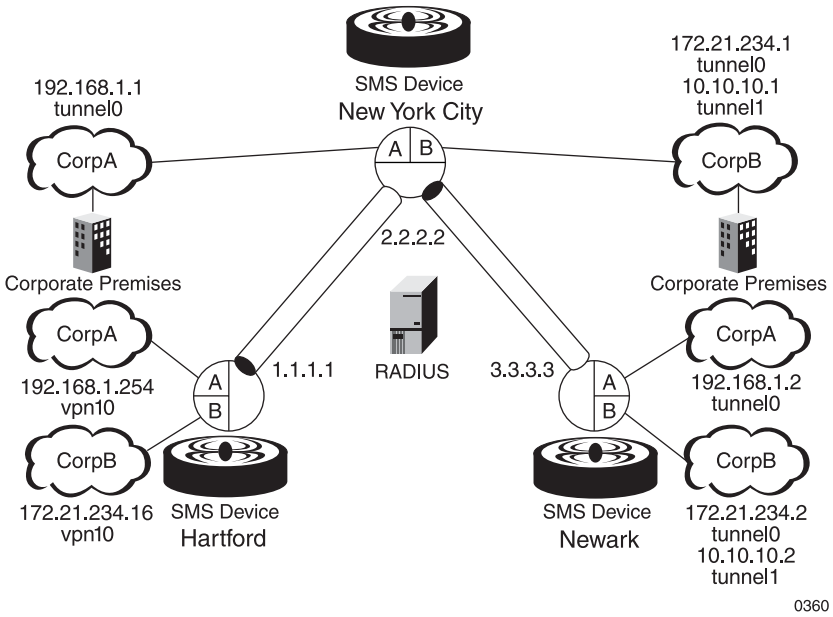
The SMS device in Trenton is configured as follows. The use of the **server** keyword causes the SMS device to pick up the RADIUS server attributes.

```
[local]Hartford(config)#context local
[local]Hartford(config-ctx)#aaa authentication gre radius
[local]Hartford(config)#context CorpA
[local]Hartford(config-ctx)#interface tunnel0
[local]Hartford(config-if)#ip address 10.3.1.254 255.255.255.0
[local]Hartford(config)#context CorpB
[local]Hartford(config-ctx)#interface tunnel0
[local]Hartford(config-if)#ip address 172.22.1.254 255.255.255.0
[local]Hartford(config)#tunnel map
[local]Hartford(config-tunnel)#gre-tunnel Hartford-Trenton local key 370 server
[local]Hartford(config-tun-circuit)#gre-tunnel Hartford-Trenton local key 1205 server
```

GRE Server

Figure 24-3 illustrates the use of GRE server mode in a hub-and-spoke VPN topology. The hub router in New York in is designated as the server, which listens for new GRE clients.

Figure 24-3 Hub-and-Spoke VPN Topology



The following attributes are configured on the RADIUS server and are used by Hartford and New York:

```
Hartford-NewYork password = redback
  Tunnel-Medium-Type = 1
  Tunnel-Type = 10 (GRE)
  Tunnel-Client-Endpoint = 1.1.1.1
  Tunnel-Server-Endpoint = 2.2.2.2
GRE-1.1.1.1:2.2.2.2:5370 password = redback
  Tunnel-Client-VPN = CorpA
  Tunnel-Client-Int-Addr = 192.168.1.254
  Tunnel-Server-Int-Addr = 192.168.1.1
GRE-1.1.1.1:2.2.2.2:9112 password = redback
  Tunnel-Client-VPN = CorpB
  Tunnel-Client-Int-Addr = 172.21.234.16
  Tunnel-Server-Int-Addr = 172.21.234.1
```

The following attributes configured on the RADIUS server and are used by Newark and New York:

```
Newark-NewYork password = redback
  Tunnel-Medium-Type = 1
  Tunnel-Type = 10
  Tunnel-Client-Endpoint = 3.3.3.3
  Tunnel-Server-Endpoint = 2.2.2.2
GRE-3.3.3.3:2.2.2.2:5370 password = redback
  Tunnel-Client-VPN = CorpA
  Tunnel-Client-Int-Addr = 192.168.1.2
  Tunnel-Server-Int-Addr = 192.168.1.1
GRE-3.3.3.3:2.2.2.2:5050 password = redback
  Tunnel-Client-VPN = CorpB
  Tunnel-Client-Int-Addr = 172.21.234.2
  Tunnel-Server-Int-Addr = 172.21.234.1
GRE-3.3.3.3:2.2.2.2:5200 password = redback
  Tunnel-Client-VPN = CorpB
  Tunnel-Client-Int-Addr = 10.10.10.2
  Tunnel-Server-Int-Addr = 10.10.10.1
```

Although it can be practical to use a single domain-wide key for a VPN, as is the case with `CorpA` (`key 5370`), it is not required. In addition, tunnel traffic can either terminate on a single interface in the VPN context, as is the case with `CorpA` (the Tunnel-Server-Int-Addr attribute is `192.168.1.1`), or it can terminate on multiple interfaces, as is the case with `CorpB`.

The following commands are configured on the Hartford SMS device:

```
[local]Hartford(config)#context local
[local]Hartford(config-ctx)#aaa authentication gre radius
[local]Hartford(config)#context CorpA
[local]Hartford(config-ctx)#interface vpn10
[local]Hartford(config-if)#ip address 192.168.1.254 255.255.255.0
[local]Hartford(config)#context CorpB
[local]Hartford(config-ctx)#interface vpn10
[local]Hartford(config-if)#ip address 172.21.234.16 255.255.255.0
```

```
[local]Hartford(config)#tunnel map
[local]Hartford(config-tunnel)#gre-tunnel Hartford-NewYork local key 5370
[local]Hartford(config-tun-circuit)#gre-tunnel Hartford-NewYork local key 9112
```

The following commands are configured on the Newark SMS device:

```
[local]Hartford(config)#context local
[local]Hartford(config-ctx)#aaa authentication gre radius
[local]Hartford(config)#context CorpA
[local]Hartford(config-ctx)#interface tunnel0
[local]Hartford(config-if)#ip address 192.168.1.2 255.255.255.0
[local]Hartford(config)#context CorpB
[local]Hartford(config-ctx)#interface tunnel0
[local]Hartford(config-if)#ip address 172.21.234.2 255.255.255.0
[local]Hartford(config-ctx)#interface tunnel1
[local]Hartford(config-if)#ip address 10.10.10.2 255.255.255.0
[local]Hartford(config)#tunnel map
[local]Hartford(config-tunnel)#gre-tunnel Hartford-NewYork local key 5370
[local]Hartford(config-tun-circuit)#gre-tunnel Hartford-NewYork local key 5050
```

The New York hub SMS device is configured as follows:

```
[local]Hartford(config)#context local
[local]Hartford(config-ctx)#aaa authentication gre radius
[local]Hartford(config)#context CorpA
[local]Hartford(config-ctx)#interface tunnel0
[local]Hartford(config-if)#ip address 192.168.1.1 255.255.255.0
[local]Hartford(config)#context CorpB
[local]Hartford(config-if)#interface tunnel0
[local]Hartford(config-if)#ip address 172.21.234.1 255.255.255.0
[local]Hartford(config-ctx)#interface tunnel1
[local]Hartford(config-if)#ip address 10.10.10.1 255.255.255.0
[local]Hartford(config)#tunnel map
[local]Hartford(config-tunnel)#gre-circuit creation on-demand aaa local
```

Configuring L2TP

This chapter provides an overview of the Layer 2 Tunneling Protocol (L2TP) and describes the tasks involved in configuring L2TP features through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands mentioned, see the "L2TP Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuring a LAC
- Configuring an LNS
- Configuring Tunnel Switching
- RADIUS One-Pass Feature
- DNIS-Based Tunnel Switching
- Configuring L2TP Groups
- Making Configuration Changes
- Configuring Ethernet over L2TP

Overview

The AOS implementation of L2TP supports the following:

- You can configure the AOS to function as an L2TP Access Concentrator (LAC), an L2TP Network Server (LNS), and as a tunnel switch. Any or all of these functions can be active in any given context—L2TP tunnels are per-context entities.
- A tunnel can be defined in a first context, while the sessions within that tunnel can be terminated (LNS) or can be further tunneled (tunnel switch) in any contexts.
- L2TP tunnel configurations can be configured locally (in the AOS configuration file) or they can be served by a Remote Access Dial-In User Service (RADIUS) server.
- L2TP tunnels can be encapsulated in User Datagram Protocol/Internet Protocol (UDP/IP) or they can be directly encapsulated in Asynchronous Transfer Mode (ATM) adaptation layer 5 (AAL5) or Frame Relay permanent virtual circuits (PVCs).

- In any given context, an individual Point-to-Point Protocol (PPP) session can be terminated and routed or tunneled, based on the subscriber's configuration. The AOS implementation of L2TP also allows tunnel-selection via Dialed Number Identification Service (DNIS) when serving as a tunnel switch.
- The aggregate traffic in any given L2TP tunnel can be policed or rate-limited to a specified speed and burst tolerance. This is independent of the optional per-PPP-session, per-PVC, and per-port policing and rate-limiting functions also available in the AOS.
- Subscriber circuits can be configured to allow dynamic tunnel selection. Alternatively, a subscriber circuit can be hard-wired to a specific L2TP tunnel. In the case of Ethernet over an L2TP tunnel, dynamic tunnel selection is not an option.
- The AOS supports multiple named tunnels to a given peer.
- You can configure an LNS to accept incoming tunnel Start-Control-Connection-Request (SCCRQ) packets that contain a hostname not found in the local named L2TP peer configurations or in RADIUS configurations. The unnamed tunnel can contain a password that the peer must provide before a tunnel is established. This feature is sometimes called anonymous tunnels.
- You can create L2TP groups that consist of LNS peer members. This facilitates strict-priority redundancy or load balancing among the members according to the group's configured algorithm.
- You can change the factory default settings that are applied to new L2TP peers when they are created.
- Ethernet can be tunneled over L2TP. An Ethernet session cannot be tunnel-switched, but the individual PPP sessions within PPP over Ethernet (PPPoE) can be. Ethernet and PPP sessions can use the same L2TP tunnel.

To fully utilize the L2TP tunnel features described in this chapter, you should be familiar with the configuration and operational model of the AOS and, specifically, be familiar with multiple contexts.

See Appendix D, "L2TP Attribute Value Pairs," for information on all standard and vendor-specific attribute value pairs (AVPs) supported by the AOS.

Dynamic Tunnel Selection

The mechanics of dynamic tunnel selection within the AOS are similar to those of dynamic *context* selection. In both cases, a subscriber provides a structured username of the form *user@service*. In the case of dynamic tunnel selection, however, the service is actually a tunnel name; in other words, the structured username is interpreted as *user@tunnel-name*.

To be eligible for dynamic selection, a tunnel must have a name (or alias) identical to an AOS context name or to an alias name for a context created with the **domain** context configuration mode command. In addition, a subscriber record must be configured to allow that subscriber to dynamically select a tunnel, through the **tunnel domain** command.

This latter configuration can be accomplished for all subscribers by providing the **tunnel domain** command within the default subscriber record. This configuration can be overridden in any specific subscriber records. The command instructs the AOS to map the subscriber's PPP session into a tunnel with the same name as the *@tunnel-name* portion of the structured username used by that subscriber. The *tunnel-name* provided by the subscriber must be identical to a context name or to a context domain name.

Note Dynamic tunnel selection is not available for tunneling Ethernet sessions.

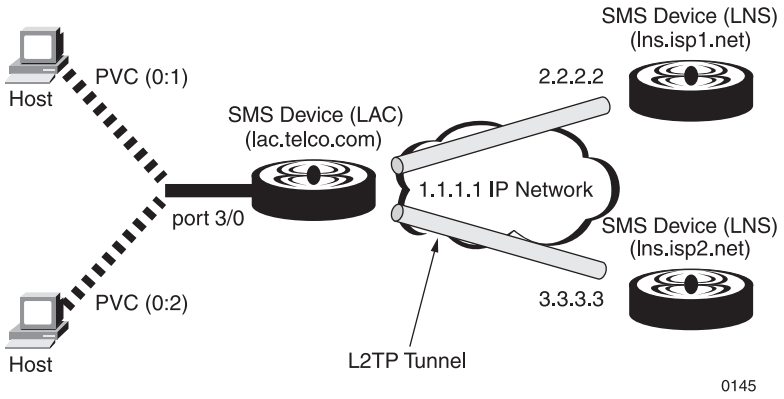
Configuring a LAC

This section provides a simple example of configuring the AOS to provide LAC functionality for a number of subscriber lines. Here we illustrate both dynamic tunnel selection and the alternative; namely, that of hard-wiring a subscriber circuit to a specific tunnel.

Figure 25-1 shows a Subscriber Management System (SMS) device terminating subscriber PVCs and tunneling these subscribers' PPP sessions to a number of respective L2TP peers. Each of these peers is assumed to have LNS functionality.

To configure a tunnel, you must know the hostname that the peer is going to use during L2TP-tunnel establishment. By default, the AOS uses the system's hostname as set by the **system hostname** command in packets sent to a peer.

Figure 25-1 L2TP Tunnels over UDP/IP



The first step is to use the **domain** command to configure alias names for the context that are identical to the tunnel names as shown in the following example:

```
[local]RedBack(config)#system hostname lac.telco.com
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#domain isp1.net
[local]RedBack(config-ctx)#domain isp2.net
```

Next, configure the tunnels themselves. The **l2tp-peer name** command defines the attributes of one or more tunnels to an L2TP tunnel peer, and requires the peer's hostname to be specified as part of the command. Specifically, the hostname specified on the **l2tp-peer name** command line is the one the peer is expected to use in the hostname field of packets exchanged in L2TP.

Because such hostnames can be unwieldy—often in the form of fully qualified domain names—the AOS allows you to create an alias for the peer that can be any character string. For example, an L2TP peer can have a fully qualified domain name of hssi_3_0.chi.core.isp.net, whereas you might want to refer to this peer as isp.net. Such aliases are created in L2TP configuration mode by using the **domain** command. An domain name for a tunnel peer can be used any place that the fully qualified hostname (that appears in the **l2tp-peer name** command) can be used. Examples of these places include **bind** commands and within subscriber records (either locally or in RADIUS).

It is common to put a fully qualified domain name of a peer in the **l2tp-peer name** command, and put the "service name that you want to expose to a subscriber in a **domain** command within L2TP configuration mode when defining the tunnel. In the example in this section, the actual hostnames are of the form lns.isp1.net, while we want to expose to subscribers service names of the form isp1.net.

The following commands configure the LT2P-over-UDP-over-IP tunnels, as well as the keys to be used by the peers to authenticate the establishment of the tunnels. By default, all L2TP tunnels function as both LAC and LNS. There might be valid administrative reasons to restrict operation to LAC-only or LNS-only, as the following example illustrates:

```
[local]lac.telco.com(config)#context local
[local]lac.telco.com(config-ctx)#l2tp-peer name lns.isp1.net media udp-ip remote
2.2.2.2 local 1.1.1.1
[local]lac.telco.com(config-l2tp)#domain isp1.net
[local]lac.telco.com(config-l2tp)#tunnel-auth SeCrEt1
[local]lac.telco.com(config-l2tp)#secondary-tunnel-auth SeCrEt2
[local]lac.telco.com(config-l2tp)#function lac-only
[local]lac.telco.com(config-l2tp)#exit
[local]lac.telco.com(config-ctx)#l2tp-peer name lns.isp2.net media udp-ip remote
3.3.3.3 local 1.1.1.1
[local]lac.telco.com(config-l2tp)#domain isp2.net
[local]lac.telco.com(config-l2tp)#tunnel-auth sEcReT2
[local]lac.telco.com(config-l2tp)#function lac-only
```

You can use the **secondary-tunnel-auth** command, as in the previous example, to create a secondary password to the peer. The secondary password is only used on a LAC that initiates a connection, and only if the primary password (set by the **tunnel-auth** command) fails. Although an L2TP network server (LNS) can also initiate a connection, the secondary password feature is not supported in that case. A secondary password is useful when transitioning from an old password to a new one. You can change the password on the LAC side of an L2TP tunnel without first notifying LNSs and other LACs.

To enable dynamic tunnel selection, we configure the default subscriber record to contain the **tunnel domain** command, as described in the previous section:

```
[local]lac.telco.com(config)#context local
[local]lac.telco.com(config-ctx)#subscriber default
[local]lac.telco.com(config-sub)#tunnel domain
```

We can specify that a specific subscriber be mapped to a particular tunnel by providing the name of that tunnel within the subscriber's record, as follows:

```
[local]lac.telco.com(config)#context local
[local]lac.telco.com(config-ctx)#subscriber name fred
[local]lac.telco.com(config-sub)#tunnel name freds-corp.com
```

As configured, a subscriber can specify *user*@isp1.net and the AOS places the subscriber's PPP session into the L2TP tunnel named `isp1.net`, provided that the user portion is not `fred`. Similarly, if the subscriber specifies *user*@isp2.net, the session is mapped to the tunnel named `isp2.net`. In fred's case, the AOS always places the session into an L2TP tunnel named `freds-corp.com`, assuming that the tunnel exists. The subscriber records can be provided in local configuration or they can be served by RADIUS.

With the aliases, tunnels, and subscriber commands in place, all that remains is to configure the subscriber circuits and to set them up for either dynamic tunnel selection or for hard-wired tunnels. The encapsulation on the subscriber circuits is simply Point-to-Point Protocol (PPP). It is in the upstream direction where these PPP sessions are tunneled to the remote peer. The first circuit below is configured for dynamic tunnel selection—note the similarity to dynamic context selection. The second circuit is hard-wired to the tunnel named `isp2.net` in the local context. We can refer to the peer whose hostname is `lns.isp2.net` by using the alias for the peer we created using the **domain** command.

```
[local]lac.telco.com(config)#port atm 3/0
[local]lac.telco.com(config-port)#atm pvc 0 1 profile ubr encapsulation ppp
[local]lac.telco.com(config-pvc)#bind authentication pap
[local]lac.telco.com(config-pvc)#exit
[local]lac.telco.com(config-port)#atm pvc 0 2 profile ubr encapsulation ppp
[local]lac.telco.com(config-pvc)#bind session isp2.net local
```

Configuring an LNS

This section describes how to configure the LNS mentioned in the previous section (see Figure 25-1). In the case of LNS operation, we must specify the method for authenticating subscriber sessions that arrive over L2TP tunnels.

The following commands configure the LNS for `isp1.net`. In this example, we have no motivation for creating an alias name for the tunnel and simply use the peer's hostname (specifically, the hostname used by the peer in L2TP-tunnel establishment, `lac.telco.com`. We assume that subscriber records exist either locally or in RADIUS for configuring and authenticating subscriber sessions.

```
[local]RedBack(config)#system hostname lns.isp1.net
[local]lns.isp1.net(config)#context local
[local]lns.isp1.net(config-ctx)#l2tp-peer name lac.telco.com media udp-ip remote
1.1.1.1 local 2.2.2.2
[local]lns.isp1.net(config-l2tp)#tunnel-auth SeCrEt1
[local]lns.isp1.net(config-l2tp)#function lns-only
[local]lns.isp1.net(config-l2tp)#session-auth chap pap
```

The configuration for the LNS in `isp2.net` is similar to that for `isp1.net`:

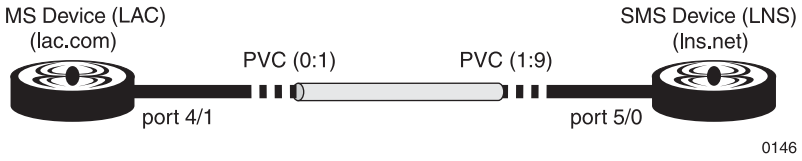
```
[local]RedBack(config)#system hostname lns.isp2.net
[local]lns.isp2.net(config)#context local
[local]lns.isp2.net(config-ctx)#l2tp-peer name lac.telco.com media udp-ip remote
1.1.1.1 local 3.3.3.3
[local]lns.isp2.net(config-l2tp)#tunnel-auth sEcReT2
[local]lns.isp2.net(config-l2tp)#function lns-only
[local]lns.isp2.net(config-l2tp)#session-auth chap pap
```

Configuring Tunnels over PVCs

The previous sections illustrated the configuration of L2TP tunnels over User Datagram Protocol (UDP)/IP. This section provides an example of a LAC and an LNS connected directly by an ATM PVC.

Figure 25-2 shows an example of `lac.com` connected to `lns.net` by an ATM PVC. Here, we omit details such as tunnel authenticators and concentrate on the configuration of the ATM PVC, the tunnel, and the binding of the PVC to the tunnel.

Figure 25-2 L2TP Tunnel over ATM PVC



The following commands configure the LAC side; first, the tunnel itself:

```
[local]RedBack(config)#system hostname lac.com
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#l2tp-peer name lns.net media pvc
```

Next, we configure the PVC and bind it to the specified tunnel. Note that the encapsulation on the circuit connecting the tunnel peers is `l2tp`:

```
[local]RedBack(config)#port atm 4/1
[local]RedBack(config-port)#atm pvc 0 1 profile ubr encapsulation l2tp
[local]RedBack(config-pvc)#bind l2tp-tunnel lns.net local
```

The configuration for the LNS side is similar. First, the tunnel:

```
[local]RedBack(config)#system hostname lns.net
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#l2tp-peer name lac.com media pvc
```

And finally, the circuit:

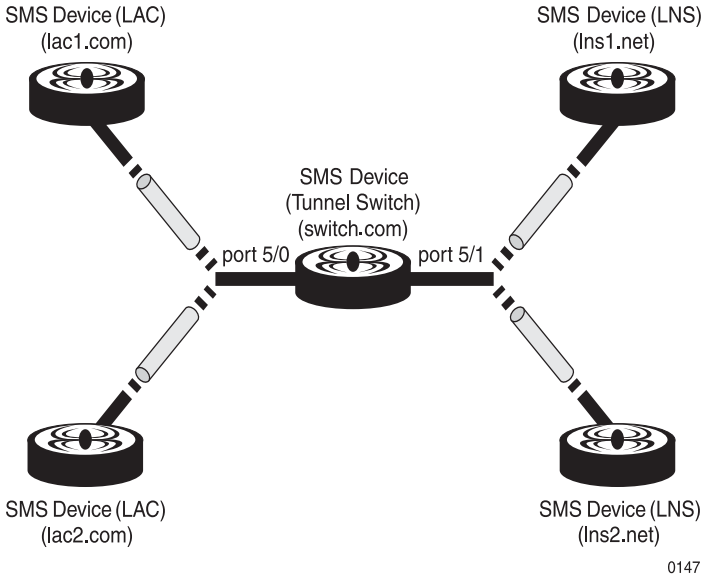
```
[local]RedBack(config)#port atm 5/0
[local]RedBack(config-port)#atm pvc 1 9 profile ubr encapsulation l2tp
[local]RedBack(config-pvc)#bind l2tp-tunnel lac.com local
```

Configuring Tunnel Switching

The AOS can also act as an L2TP tunnel switch, accepting PPP sessions over a first tunnel and relaying them over one or more other tunnels. A tunnel switch has aspects of both LAC and LNS operation and its configuration is explored in this section.

Figure 25-3 shows two LACs (lac1.com and lac2.com) feeding into a tunnel switch (switch.com), which provides upstream connectivity to each indicated LNS (lns1.net and lns2.net). Here, we assume that the two LACs are configured to tunnel appropriate PPP sessions (perhaps all of them) to switch.com. Also, we assume that each LNS is configured to accept an L2TP tunnel from switch.com.

Figure 25-3 L2TP Tunnel Switching



The following commands configure the tunnel switch. First, we provide alias names for the local context that are identical to the upstream (LNS) tunnel peer names, define the PVC-encapsulated tunnels, and set up the default subscriber in the local context such that subscriber PPP sessions are placed into the tunnel whose peer name is identical to the *@service* portion of the structured username as follows:

```
[local]RedBack(config)#system hostname switch.com
[local]switch.com(config)#context local
[local]switch.com(config-ctx)#aaa authentication subscriber none
[local]switch.com(config-ctx)#domain lns1.net
[local]switch.com(config-ctx)#domain lns2.net
[local]switch.com(config-ctx)#l2tp-peer name lac1.com media pvc
[local]switch.com(config-l2tp)#exit
[local]switch.com(config-ctx)#l2tp-peer name lac2.com media pvc
[local]switch.com(config-l2tp)#exit
[local]switch.com(config-ctx)#l2tp-peer name lns1.net media pvc
[local]switch.com(config-l2tp)#exit
[local]switch.com(config-ctx)#l2tp-peer name lns2.net media pvc
[local]switch.com(config-l2tp)#exit
[local]switch.com(config-ctx)#subscriber default
[local]switch.com(config-sub)#tunnel domain
```

Next, we define the circuits that carry the tunnels and bind the tunnels to those circuits. First, the downstream tunnels toward the LACs:

```
[local]switch.com(config)#port atm 5/0
[local]switch.com(config-port)#atm pvc 0 1 profile ubr encapsulation l2tp
```

```
[local]switch.com(config-pvc)#bind l2tp-tunnel lac1.com local
[local]switch.com(config-pvc)#exit
[local]switch.com(config-port)#atm pvc 0 2 profile ubr encapsulation l2tp
[local]switch.com(config-pvc)#bind l2tp-tunnel lac2.com local
```

Finally, the upstream tunnels toward the LNSs:

```
[local]switch.com(config)#port atm 5/1
[local]switch.com(config-port)#atm pvc 0 1 profile ubr encapsulation l2tp
[local]switch.com(config-pvc)#bind l2tp-tunnel lns1.net local
[local]switch.com(config-pvc)#exit
[local]switch.com(config-port)#atm pvc 0 2 profile ubr encapsulation l2tp
[local]switch.com(config-pvc)#bind l2tp-tunnel lns2.net local
```

Of course, we could embellish this configuration with tunnel authenticators, quality of service (QoS) parameters, and other attributes. Such things are omitted above for simplicity.

As configured, all PPP sessions that arrive at `switch.com` over the downstream tunnels `lac1.com` and `lac2.com` are mapped into an upstream tunnel selected by the structured username. For example, a subscriber name of mary@lns1.net is mapped into the `lns1.net` tunnel.

RADIUS One-Pass Feature

When both subscribers and tunnels are served via RADIUS, the RADIUS server is ordinarily queried twice—once for subscriber information and a second time for the tunnel attributes. The AOS also supports a one-pass option whereby the tunnel attributes are provided in the subscriber record so that a second query to the RADIUS server is not necessary. A minimum set of RADIUS attributes and values are required to avoid the second query. Additional tunnel attributes can also be included and, if included, are used when building the peer. The minimum attributes are:

Tunnel-Medium-Type = 1 (IP)

Tunnel-Assignment-ID = *peer name*

Tunnel-Server-Endpoint = *remote peer IP address*

Tunnel-Type = 3 (L2TP)

The **aaa authorization tunnel** command can be set to **local** or **radius** for this one-pass feature.

DNIS-Based Tunnel Switching

The AOS implementation of L2TP also allows tunnel-selection via DNIS, configurable on a per-tunnel basis. When operating as a tunnel switch, the AOS supports three possible mechanisms for determining the disposition of sessions that arrive over an incoming tunnel:

- Use DNIS to select an outgoing tunnel and reject all sessions for which no DNIS is provided—configured by providing the **dnis only** command within the incoming tunnel configuration.

- Use DNIS to select an outgoing tunnel if it is provided; otherwise, use the structured username as determined by Password Authentication Protocol (PAP) or Challenge Handshake Authentication Protocol (CHAP), to select an outgoing tunnel—configured by providing the **dnis** command (*without* the **only** keyword) and the **session-auth** command within the incoming tunnel configuration.
- Ignore DNIS completely and use the structured username as determined by PAP or CHAP to select an outgoing tunnel—configured by providing the **session-auth** command within the incoming tunnel configuration.

Configuring DNIS-based tunnel switching involves the following steps:

1. Create the incoming tunnels and provide the **dnis** command within L2TP configuration mode to indicate that sessions in those tunnels are to be switched based upon DNIS.
2. Create the outgoing tunnels and provide tunnel names or aliases (via the **domain** L2TP configuration mode command) that match the phone number to be used to select those tunnels.

As an example, consider the tunnel switch (switch.com) shown in Figure 25-3. To set up the incoming tunnels (from `lac1.com` and `lac2.com`) for DNIS-based tunnel selection, we would configure the peers as follows. Note that `lac1.com` is configured to reject sessions that do not provide DNIS, while `lac2.com` is configured to check first for DNIS, then fall back to a PAP-negotiated structured username if DNIS is not provided:

```
[local]RedBack(config)#system hostname switch.com
[local]switch.com(config)#context local
[local]switch.com(config-ctx)#aaa authentication subscriber none
[local]switch.com(config-ctx)#domain lns1.net
[local]switch.com(config-ctx)#domain lns2.net
[local]switch.com(config-ctx)#l2tp-peer name lac1.com media pvc
[local]switch.com(config-l2tp)#dnis only
[local]switch.com(config-l2tp)#exit
[local]switch.com(config-ctx)#l2tp-peer name lac2.com media pvc
[local]switch.com(config-l2tp)#dnis
[local]switch.com(config-l2tp)#session-auth pap
[local]switch.com(config-l2tp)#exit
[local]switch.com(config-ctx)#subscriber default
[local]switch.com(config-sub)#tunnel domain
```

Next, we define the outgoing tunnels and provide aliases for those tunnels that match the phone numbers used by subscribers to reach the services represented by those tunnels:

```
[local]switch.com(config-ctx)#l2tp-peer name lns1.net media pvc
[local]switch.com(config-l2tp)#domain 8005555555
[local]switch.com(config-l2tp)#exit
[local]switch.com(config-ctx)#l2tp-peer name lns2.net media pvc
[local]switch.com(config-l2tp)#domain 8005551212
```

As configured, if a subscriber dials a phone number of (800) 555-5555, and the subscriber's session arrives at `switch.com` over a tunnel from either `lac1.com` or `lac2.com`, the session is placed into the tunnel `lns1.net`. Similarly, if a subscriber dials a phone number of (800) 555-1212, the session is placed into the tunnel `lns2.net`. Note also, that sessions arriving over `lac2.com` that have no DNIS are tunnel-switched according to the structured username. For example, a session arriving over `lac2.com` with no DNIS and with a structured username of fred@lns2.net is placed into the `lns2.net` tunnel.

RADIUS Support for DNIS-Based Tunnel Switching

You can configure tunnels in RADIUS and never have to enter **domain** L2TP configuration commands for all the possible DNIS strings as described in the previous section on DNIS-based tunnel switching.

To configure DNIS-based tunnel switching using RADIUS, first create the incoming tunnels and provide the **dnis only** command (or its RADIUS equivalent) to indicate that sessions in those tunnels are to be switched based upon DNIS.

For example, if you entered the RADIUS equivalent of the following:

```
[local]RedBack(config)#context TunnelsA
[local]RedBack(config-ctx)#aaa authorization tunnel radius
[local]RedBack(config-ctx)#l2tp-peer name Incoming media pvc
[local]RedBack(config-ctx)#dnis only
[local]RedBack(config-ctx)#max-sessions 128
```

The resulting RADIUS record would look something like this:

```
Incoming
RedBack:Tunnel_Dnis_Only = TRUE
RedBack:Tunnel_Max_Sessions = 128
```

If you have two outgoing tunnels named `Outgoing_One` and `Outgoing_Two` in the context `TunnelsA`, you would need to map the actual DNIS values (phone numbers) into the outgoing tunnels. To do that, create pseudo-tunnel records in RADIUS to provide that mapping.

For example, if you want to map the DNIS value 5553945 to `Outgoing_One` and 5559393 to `Outgoing_Two`, the RADIUS pseudo-tunnel records would look something like this:

```
5553945
RedBack:Tunnel_Name = “Outgoing_One”
RedBack:Context_Name = “TunnelsA”

5559393
RedBack:Tunnel_Name = “Outgoing_Two”
RedBack:Context_Name = “TunnelsA”
```

The AOS then queries RADIUS based on the incoming DNIS (assuming that the **aaa authorization tunnel radius** command is configured for the context `TunnelsA`), and the actual configuration for the outgoing tunnel is obtained from RADIUS.

Note It is not required that the outgoing tunnels be in the same context as in the previous example. RADIUS support for DNIS-based tunnel switching works the same way when the outgoing tunnels are in different contexts.

Configuring L2TP Groups

An L2TP group is a group of LNSs among which PPP sessions are parceled out. You must configure all the group members (peers), and the group itself, in the same context. L2TP groups are created in context configuration mode and configured in L2TP group configuration mode.

Some considerations when configuring L2TP groups are as follows:

- L2TP group and peer names must be unique. Do not give a group a name that is already being used by a peer, a peer domain, a group domain, or even an L2F peer in the same context.
- An L2TP group name can be used in the following commands where an L2TP peer name can be used:

  tunnel name *tun-name* (subscriber configuration mode)
  bind session *tun-name context* (circuit configuration mode)

  A group name (or alias) cannot be used in the **bind l2tp-tunnel** command in circuit configuration mode.
- The peers do not have to be defined prior to inclusion in a group. This is important because the peers can be served by RADIUS (see the next section for special RADIUS considerations).
- PPP sessions are distributed among the peers in a group according to the algorithm specified in the **algorithm** command (see the **algorithm** command documentation in the "L2TP Commands" chapter in the *Access Operating System (AOS) Command Reference* publication).

The algorithm options are:

- Strict-priority

  Each peer is assigned a priority. At the CLI, the priorities correspond to the order in which the peers are listed, the highest priority peer being listed first. Sessions are directed to the highest priority peer until or unless connectivity to that peer is lost, the maximum number of sessions to the peer has been reached, or the peer is marked as dead, at which time sessions are directed to the next peer in line.
- Load balancing

  Each session is directed to the peer that has the fewest sessions at the moment so that sessions are distributed across peers in the group more or less equally. Peers can still have assigned priorities, but they are ignored.

Both algorithms are subject to the maximum number of tunnels and sessions configured for the peers that are members of the group. For example, if strict-priority is being used and the maximum number of sessions is reached on the highest priority peer, additional sessions are sent to the next highest priority peer.

When a peer is not reachable (regardless of the algorithm being used), it is marked with a deadtime (see the **deadtime** command documentation in the *Access Operating System (AOS) Command Reference* publication). There is no further attempt to reach a peer that is marked as dead until the deadtime has expired. A peer is also marked with a deadtime if it is not yet defined at the time that a connection attempt is made.

The following example creates a new L2TP group called `group1`, adds two members to the group, sets the algorithm to load balancing, and sets the deadtime to 15 minutes:

```
[local]RedBack(config-ctx)#l2tp-group name group1
[local]RedBack(config-l2tpgrp)#peer-name 1peer
[local]RedBack(config-l2tpgrp)#peer-name 2peer
[local]RedBack(config-l2tpgrp)#algorithm load-balance
```

```
[local]RedBack(config-l2tpgrp)#deadtime 15
```

The next example shows using the **show l2tp group** operator exec command to display the newly created group. Note that the asterisk in front of the peer called `2peer` indicates that the peer is dead:

```
[local]RedBack#show l2tp group group1

Group name:   group1          RADIUS: YES
Algorithm:    Load-balance    Deadtime: 15
Description:  (NO DESCRIPTION)
Peers:        1peer                    *2peer
Domains:      (NO DOMAINS)

                            Max    Tun   Max   Ses
Peer Name Local Name Med    Tuns   Cnt   Ses   Cnt   Stat  LAC   LNS   Named
--------- ---------- ---    ----   ---   ---   ---   ----- ---   ---   ------
1peer     tgrp3      PVC    4      1     65535 7     NO    YES   YES   YES
2peer     tgrp1      UDP    4      0     20    0     NO    YES   YES   YES
```

RADIUS Considerations for Configuring L2TP Groups

There are some significant considerations for RADIUS-based configurations of L2TP groups resulting from the requirement that both types of RADIUS servers be supported: those that support tunnel extensions (tunnel tags) and those that do not.

Servers That Do Not Support Tunnel Extensions

The following is an example of a RADIUS tunnel configuration for a server that does not support tunnel extensions. The L2TP group is named `isp` and the peer members are `peer1`, `peer2`, and `peer3`. The Tunnel-Preference attribute determines which tunnel has the highest priority for the case of strict priority. Lower preference numbers mean higher priority. If the Tunnel-Preference attribute is missing from all peers, the server-dependent order in which the peers are listed becomes the priority order. We highly recommend setting the priority explicitly. In the case that some peers have an explicit priority and some do not, the ones without priorities are considered of lower priority than those with explicit priorities. In the following example, `peer2` is the highest-priority peer because it has the lowest preference value. These examples represent a vendor-specific implementation (Merit server).

isp
Password = “Redback”,
Service-type = Outbound,
RedBack:Tunnel-Algorithm = 1
RedBack:Tunnel-Deadtime = 10,
RedBack:Tunnel-Group = TRUE,
Tunnel-Assignment-Id = “peer2”,
Tunnel-Assignment-Id = “peer3”,
Tunnel-Assignment-Id = “peer1”

peer1
Password = “Redback”,
Service-type = Outbound,
Tunnel-Medium-Type = IP,

Tunnel-Client-Endpoint =12.1.1.1,
Tunnel-Server-Endpoint = 12.1.1.5,
Tunnel-Password = “pass4me”,
Tunnel-Preference = 3,
Redback:Tunnel-Local-Name = “gr-atm1”

peer2
Password = “Redback”,
Service-type = Outbound,
Tunnel-Medium-Type = IP,
Tunnel-Client-Endpoint =22.1.1.1,
Tunnel-Server-Endpoint = 22.1.1.5,
Tunnel-Password = “pass4me”,
Tunnel-Preference = 1,
Redback:Tunnel-Local-Name = “gr-atm2”

peer3
Password = “Redback”,
Service-type = Outbound,
Tunnel-Medium-Type = IP,
Tunnel-Client-Endpoint =32.1.1.1,
Tunnel-Server-Endpoint = 32.1.1.5,
Tunnel-Password = “pass4me”,
Tunnel-Preference = 2,
Redback:Tunnel-Local-Name = “gr-atm3”

Servers That Do Support Tunnel Extensions

The following is an example of a RADIUS tunnel configuration for a server that does support tunnel extensions (tunnel tags). The Tunnel-Preference attribute determines which tunnel has the highest priority for the case of strict-priority. Lower preference numbers mean higher priority. In the following example, the tunnel with tag 1 is the highest-priority peer, because it has the lowest preference value. If the Tunnel-Preference attribute is missing from all peers, the tag value becomes the priority order (in other words, the lowest-tag-numbered peer becomes the highest-priority peer). We highly recommend setting the priority explicitly rather than overloading the tag field. In the case that some peers have a priority and some do not, the ones without priorities are considered of lower priority than those with explicit priorities.

isp
Password = “Redback”,
RedBack:Tunnel-Deadtime = 10,
Service-type = Outbound,
Tunnel-Medium-Type = 1:IP,
Tunnel-Client-Endpoint = 1:12.1.1.1,
Tunnel-Server-Endpoint = 1:12.1.1.5,
Tunnel-Password = 1:“pass4me”,
Tunnel-Preference = 1:3,
Redback:Tunnel-Local-Name = 1:“gr-atm1”
Tunnel-Medium-Type = 2:IP,
Tunnel-Client-Endpoint = 2:22.1.1.1,
Tunnel-Server-Endpoint = 2:22.1.1.5,
Tunnel-Password = 2:“pass4me”,

Tunnel-Preference = 2:3,
Redback:Tunnel-Local-Name = 2:“gr-atm2”
Tunnel-Medium-Type = 3:IP,
Tunnel-Client-Endpoint = 3:32.1.1.1,
Tunnel-Server-Endpoint = 3:32.1.1.5,
Tunnel-Password = 3:“pass4me”,
Tunnel-Preference = 3:3,
Redback:Tunnel-Local-Name = 3:“gr-atm3”

The names of the individual peers can be anonymous for UDP/IP tunnels. The names of the tunnels are assigned as groupname_tag. For example, the name for the first tunnel in the previous example would be assigned as isp_1.

However, in the case of PVC-based tunnels (Tunnel-Medium-Type = PVC), the above mechanism would not suffice, because the **bind l2tp-tunnel** command requires an explicit peer name. In such a case, use the Tunnel-Assignment-Id RADIUS attribute to associate a specific peer with the one named in the **bind l2tp-tunnel** command.

L2TP groups from RADIUS servers that support tunnel extensions are limited to 31 peers per group.

Tunnel Group Override

The default behavior of SMS devices is to create a tunnel group for multiple tunnels (grouped by tags in accordance with RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*) received in a RADIUS response. This is the default behavior of SMS devices.

Use the **no l2tp radius auto-group** command to override automatic tunnel group creation, allowing a RADIUS server to return a set of tunnels ordered by preference, using the Tunnel-Preference RADIUS attribute. The tunnel with the lowest preference value is attempted first. If tunnel creation fails, the system tries the tunnel with the next lowest preference value, and so on. This tunnel group override enables limited L2TP tunnel fail-over, and enables the RADIUS server to perform load-balancing of subscribers across tunnels.

To use the tunnel group override feature, the RADIUS server must respond with a full set of tunnel attributes, specifying client and server endpoints, and preference values grouped by tags.

In general, due to Point-to-Point Protocol (PPP) client timeouts and tunnel setup delay, we recommend returning no more than 3 tunnels in a RADIUS response.

The tunnel group override feature takes effect only if tunnel creation fails. If the tunnel is configured with a maximum session count, and the new PPP session would cause the maximum session count for the tunnel to be exceeded, the second tunnel is not be attempted.

Making Configuration Changes

Changing the configuration of an L2TP peer or L2TP group with one or more established tunnels does not take effect until you delete all tunnels to the peer (using the **clear tunnel** command) or until all the tunnels to the peer come down naturally. The configuration database is queried again to reestablish tunnels to the peer, thereby implementing the new configuration.

You can use the **show l2tp info** or **show l2tp group** command to find out if there are any active sessions before you clear the peers, because clearing the peers terminates any active sessions through the tunnels.

Changing L2TP Default Settings

When you create a new L2TP peer, there are factory default settings that are applied and remain in effect until you change them in the configuration of the individual peer. However, you can change the default settings that are applied to new peers by entering L2TP configuration mode via the **l2tp-peer default** command and then issuing L2TP configuration mode commands with the desired default values.

The factory default settings for new L2TP peers can be changed by entering L2TP configuration mode via the **l2tp-peer default** command; see Table 25-1.

Table 25-1 L2TP Factory Default Settings

Setting	L2TP Configuration Mode Command	Factory Default Value
Text description of the peer	**description**	None
Tunnel switching based on DNIS	**dnis**	Disabled
Ethernet encapsulation type	**ethernet encapsulation**	IP/bridging encapsulation
Authentication method for Ethernet session	**ethernet session**	None
LAC and/or LNS functionality	**function**	Both LAC and LNS
Local hostname for SCCRQ control messages	**local-name**	System hostname
Maximum number of sessions allowed per tunnel	**max-sessions**	65,355 sessions
Incoming rate and burst tolerance	**police**	None
Outgoing rate and burst tolerance	**rate-limit**	None
Number of times an unacknowledged control message is retransmitted before tunnel is brought down	**retry**	5 retransmissions
Authorization protocol for arriving sessions	**session-auth**	CHAP PAP protocol
Amount of time to wait for acknowledgment before retransmitting	**timeout**	6 seconds
L2TP password	**tunnel-auth**	None
Number of messages peer can send without receiving acknowledgment	**tunnel-window**	10 messages

Once the default settings have been changed, the new values are applied to any new peer that is created, unless the values are changed in the configuration for an individual peer.

Configuring Ethernet over L2TP

This section describes how to configure L2TP tunnels to carry Ethernet-encapsulated PPP over Ethernet (PPPoE) sessions. Before configuring your system for Ethernet over L2TP, be sure to read the preceding section called “Configuring an LNS.”

By allowing Ethernet sessions over L2TP tunnels, the AOS is able to provide the LNS full control over the advertisement of services. The following sections are included:

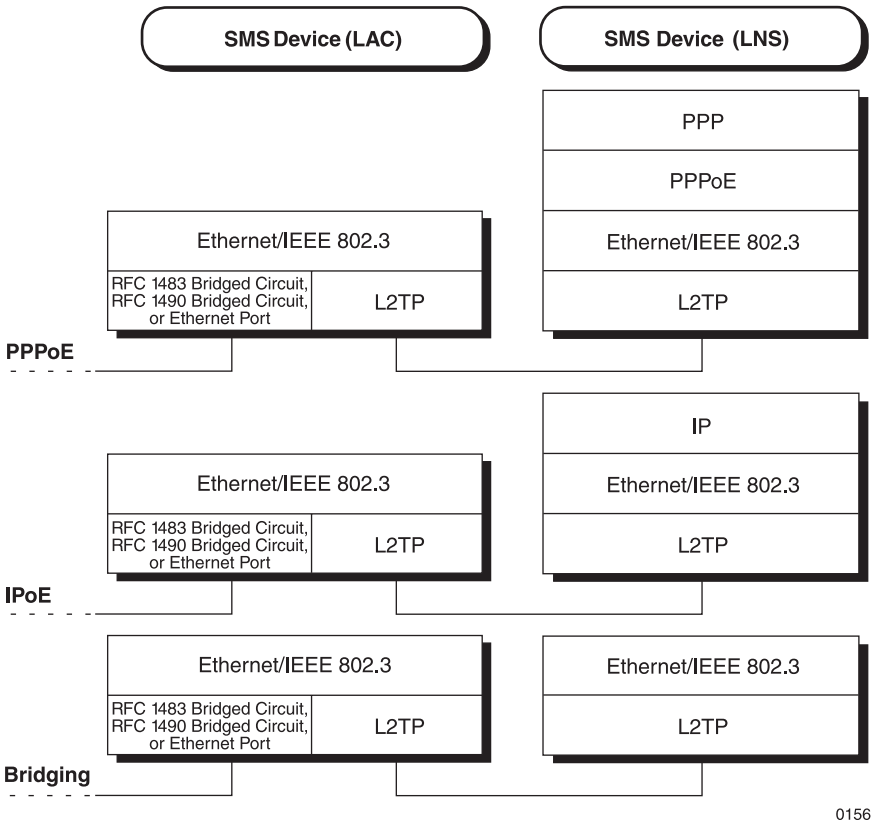
- Overview
- Configuration Tasks on the LAC Side
- Configuration Tasks on the LNS Side
- Configuration Examples

For a complete description of the commands related to Ethernet over L2TP, see the “L2TP Commands” chapter in the *Access Operating System (AOS) Command Reference* publication.

Overview

The AOS implementation of Ethernet over L2TP allows Ethernet sessions to be tunneled intact to the LNS. Given that PPPoE is a bridged protocol, the Ethernet over L2TP capability is useful in situations where it is desirable to advertise PPPoE services directly from the LNS rather than the LAC. Figure 25-4 shows the relationship of protocols/encapsulations between a LAC and an LNS in the three types of sessions that can be tunneled in this fashion: PPPoE, IP over Ethernet (IPoE), and Bridging.

Figure 25-4 Protocol Stack When Tunneling Ethernet over L2TP



If more than one Ethernet session is tunneled from one peer and bound to the same interface on the LNS, the AOS cannot use Address Resolution Protocol (ARP) to resolve the IP address to a physical Media Access Control (MAC) address. As an alternative, you can use Dynamic Host Configuration Protocol (DHCP) with secured-ARP as a way to map IP addresses to physical MAC addresses.

Configuration Tasks on the LAC Side

To configure Ethernet over L2TP on the LAC side, first set up the L2TP peers according to the instructions in the previous section, “Configuring a LAC.” Then, perform the tasks described in the following sections:

- Identify the Ethernet Ports or Bridge-Encapsulated Circuits
- Bind the Ports or Bridge-Encapsulated Circuits to the Peers
- Enable Ethernet Sessions Retry over L2TP
- Configure Ethernet Timeout

Identify the Ethernet Ports or Bridge-Encapsulated Circuits

The first step is to decide which Ethernet ports or bridge-encapsulated circuits you want tunneled over L2TP. This is necessary because the next step is to hard bind them to L2TP peers.

Bind the Ports or Bridge-Encapsulated Circuits to the Peers

To bind a port or circuit to an L2TP peer, enter the following command in port, circuit, or HDLC channel configuration mode as appropriate:

bind session *peer-name context* [**bridge-acl** *list-name*]

The *peer-name* argument is the name of the L2TP peer to which the circuit or port is to be bound and the *context* argument is the context in which that peer exists. The optional **bridge-acl** *list-name* construct allows you to include a bridge access control list. The access control list must already have been configured in the specified context. You can use this construct to filter packets so that only PPPoE traffic is allowed through an Ethernet over L2TP tunnel.

If you are binding an Ethernet port, the **bind session** command puts the port into promiscuous mode, which means that it ignores MAC addresses and tunnels everything to the LNS. The concept of promiscuous mode is implicit for RFC 1483 bridged and RFC 1490 bridged encapsulated circuits.

Enable Ethernet Sessions Retry over L2TP

To enable the creation of Ethernet sessions over L2TP, enter the following command in context configuration mode:

aaa authentication re-try *minutes*

The *minutes* argument is the number of minutes the system is to wait before re-attempting to connect after failure to establish a session to the tunnel peer. By default, no further attempts are made once an attempt to create a session has failed. No attempt is made to create a connection until data begins to come through over the circuit/port.

Configure Ethernet Timeout

To configure a timeout that is specific to Ethernet traffic (does not consider PPP traffic), enter the following command in global configuration mode:

l2tp eth-sess-idle-timeout *seconds*

The value of the *seconds* argument is the time between polls of the L2TP session statistics. If the inbound or outbound packet statistics show no change from the last poll, the session is considered idle and is terminated.

The timeout is typically configured in conjunction with using the **bridge-acl** *list-name* construct in the **bind session** command to filter packets so that only PPPoE traffic is allowed through an Ethernet over L2TP tunnel. If you configure an idle timeout, and you do not filter packets with a bridge access control list, any Ethernet traffic prevents the session from timing out.

Configuration Tasks on the LNS Side

To configure Ethernet over L2TP from the LNS side, you must first set up the L2TP peers as described in the previous section, “Configuring an LNS.” Then, perform the tasks in the following sections:

- Determine How Subscribers Are Terminated
- Bind the Sessions

Determine How Subscribers Are Terminated

You can terminate subscribers in one of two ways: IPoE or PPPoE. These two encapsulations are mutually exclusive and apply to all Ethernet-encapsulated sessions from the peer. This decision is made on the LNS side, because the LAC is intended to send everything, without evaluation.

If the client is using PPPoE, configure the peer as such by entering the following command in L2TP configuration mode:

ethernet encapsulation ppp over-ethernet

If the client is using IPoE, the default setting for this command (**ppp**) is what you want.

Bind the Sessions

To bind a session to a PPPoE client, enter the following command in L2TP configuration mode:

ethernet session auth {**pap** | **chap** | **chap pap**} [**maximum** *sessions*] [**context** *ctx-name* | **service-group** *group-name*]}

The **pap**, **chap**, and **chap pap** keywords are authentication method choices, the **maximum** *sessions* construct allows you to limit the number of PPPoE sessions allowed per L2TP session, the **context** *ctx-name* construct allows you to restrict the Ethernet-encapsulated PPPoE sessions to the named context, and the **service-group** *group-name* construct allows you to limit the services available to those permitted by the named service access list. If the peer is not first encapsulated as PPPoE (with the **ethernet encapsulation ppp over-ethernet** command), the **auth** construct is not available on the command line.

To bind a session to an IPoE client, enter the following command in L2TP configuration mode:

ethernet session interface *if-name ctx-name*

The *if-name* argument is the name of the interface to which the Ethernet session is to be bound and the *ctx-name* argument is the name of the context in which the interface exists.

Configuration Examples

The following shows example configurations on the LAC and LNS sides:

On the LAC side:

```
[local]RedBack(config-ctx)#l2tp-peer name lnsmain media pvc
[local]RedBack(config-l2tp)#local-name lacmain
.
.
.
[local]RedBack(config)#port ethernet 3/0
[local]RedBack(config-port)#bind session lnsmain local
```

On the LNS side:

```
[local]RedBack(config-ctx)#l2tp-peer name lacmain media pvc
[local]RedBack(config-l2tp)#local-name lnsmain
.
.
.
[local]RedBack(config-l2tp)#ethernet encapsulation ppp over-ethernet
```

```
[local]RedBack(config-l2tp)#ethernet session auth pap
```

Configuring L2F

This section describes how the Access Operating System (AOS) interoperates with legacy systems that are implementing Cisco's Layer 2 Forwarding (L2F) protocol. The L2F protocol supports the creation of secure virtual private dial-up networks over the Internet and is one of the predecessors to the Layer 2 Tunneling Protocol (L2TP).

For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "L2F Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

The following sections are included:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

The AOS implementation of L2F supports the following:

- Both Network Access Server (NAS) and home gateway functions. Consistent with the limitations of the L2F protocol, a peer can function as one or the other, but not both.
- Tunnel switching between L2F tunnels and between L2F and L2TP tunnels.
- User Datagram Protocol (UDP)/IP tunnel encapsulation.
- A tunnel can be defined in one context, while the sessions within that tunnel can be terminated or can be further tunneled (tunnel switch) in any contexts.
- L2F tunnel configurations can be configured locally (in the AOS configuration file) or they can be served by Remote Access Dial-In User Service (RADIUS).
- Multiple named tunnels to a given peer.
- The aggregate traffic in any given L2F tunnel can be policed or rate-limited to a specified speed and burst tolerance. This is independent of the optional per-Point-to-Point Protocol (PPP)-session, per-permanent virtual circuit (PVC), and per-port policing and rate-limiting functions also available in the AOS.

- In any given context, an individual PPP session can be terminated and routed or tunneled, based on the subscriber's configuration. The AOS implementation of L2F also allows tunnel selection via the Dialed Number Identification Service (DNIS) for sessions arriving over L2TP.

Configuration Tasks

To configure L2F, perform the tasks in the following sections:

- Create L2F Peers
- Configure Peers as NAS or Home Gateway
- Establish Aliases for the Peers
- Configure the Parameters
- Create the Circuits
- Bind the Circuits to the Peers
- Clear Tunnels or Sessions

Create L2F Peers

To create an L2F peer and enter L2F configuration mode, enter the following command in context configuration mode:

l2f-peer name *peer-name* **media udp-ip remote** *ip-address* **local** *ip-address*

The *peer-name* argument is the same name for the new peer that will be provided as the hostname in confirmation (CONF) packets. If that name is complex, you can create one or more aliases (using the **domain** command) once you enter L2F configuration mode. The *ip-address* argument is first the remote and then the local IP address for the tunnels.

This command puts you into L2F configuration mode such that all subsequent commands (until you exit this mode) apply to the configuration of the named L2F peer. You can also enter L2F configuration mode to change the configuration of an existing peer. Use the *peer-name* argument to specify the existing peer you want to change.

Configure Peers as NAS or Home Gateway

Each peer can be configured with either home gateway functionality (receiving end of the tunnels) or NAS functionality (originating end of the tunnels), but not both. Once a new peer is created, you must establish which functionality it will have. You are actually configuring your system's relationship to the peer. In other words, if your system receives messages from peer_1, configure peer_1 with home gateway functionality, because that is the function that your system has relative to peer_1. If your system sends messages to peer_2, configure peer_2 with NAS functionality, because that is the function your system has relative to peer_2.

To select the functionality of the peer, enter the following command in L2F configuration mode:

function {**nas** | **home-gateway**}

Establish Aliases for the Peers

Often, the name of the peer can be a fully qualified domain name, such as hssi_0_5.chi_core.isp.net. You can create shorter, easier-to-use aliases for the peer.

To create an alias for a peer, enter the following command in L2F configuration mode:

domain *dom-name*

A domain name can be used in any command that calls for the *peer-name* argument.

Configure the Parameters

The tunnel parameters that you configure for the peer serve as a template for all tunnels to that peer. Enter the configuration commands in L2F configuration mode. The configurable parameters include:

- Text description of the peer. Can be any alphanumeric string not longer than one line. Enter the following command:

  description *text*

- Local hostname for outbound CONF messages. By default, the system hostname is used, but you can change it by entering the following command:

  local-name *hostname*

- Maximum number of sessions per tunnel. The default maximum is 65,355—the largest value in the possible range. To limit the number of sessions per tunnel to a smaller number, enter the following command:

  max-sessions *maxses*

- Maximum number of tunnels allowed for the peer. New tunnels are opened as needed, until the maximum number of tunnels has been reached. The default maximum is one tunnel per peer. To increase this number (no more than 128 tunnels per peer are possible), enter the following command:

  max-tunnels *maxtun*

- Policing of incoming traffic. To limit the aggregate packet stream received over an L2F tunnel by rate and burst tolerance, enter the following command:

  police rate *rate* **burst** *size*

  The *rate* argument is a limiting rate in kbps and the *size* argument is a burst tolerance size in bytes.

- Rate limiting of outgoing traffic. To limit the aggregate packet stream transmitted over an L2F tunnel by rate and burst tolerance, enter the following command:

  rate-limit rate *rate* **burst** *size*

  The *rate* argument is a limiting rate in kbps and the *size* argument is a burst tolerance size in bytes.

- Amount of time to wait for an acknowledgment before a control message is retransmitted to a peer. To change the default of four seconds, enter the following command:

  timeout *seconds*

- Number of times an unacknowledged control message is retransmitted to a peer before the tunnel is brought down. To change the default of three retransmissions, enter the following command:

  retry *count*

- Method used by home gateways to authenticate arriving subscriber sessions. To change the default of Challenge Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP), or to add context or service group restrictions, enter the following command:

  session-auth {**pap** | **chap** | **chap pap**} [**context** *ctx-name* | **service-group** *group-name*]

  The **context** *ctx-name* and **service-group** *group-name* constructs are mutually exclusive and represent optional restrictions on authentication.

- Local and remote passwords to the tunnel peer. These are used for tunnel authentication. To set the passwords, enter the following command:

  tunnel-auth local *secret1* **remote** *secret2*

  The *secret1* argument is the password expected by the peer; the *secret2* argument is the password expected from the peer.

Create the Circuits

To create a subscriber circuit, enter the following command in port configuration mode:

atm pvc *vpi vci* **profile** *profile* **encapsulation ppp**

Note The appropriate encapsulation is PPP.

See the "ATM Commands" chapter in the *Access Operating System (AOS) Command Reference* publication for detailed syntax description and usage guidelines for this command.

Bind the Circuits to the Peers

To bind a circuit to a tunnel peer, enter the following command in circuit configuration mode:

bind session *peer-name context*

Both the name of the peer and the context must be specified. In this command, a domain name for the peer can be used for the *peer-name* argument. Dynamic tunnel selection is not available for L2F. The **bind session** command creates a hard binding to a specific peer in a specific context.

See the "Bind Commands" chapter in the *Access Operating System (AOS) Command Reference* for detailed syntax description and usage guidelines for this command.

Clear Tunnels or Sessions

You can clear (shut down) all or a specific tunnel to an L2F peer. You can also clear all or a specific session within a tunnel. To clear tunnels or sessions, enter the following command in administrator exec mode:

clear tunnel peer *peer-name* [**tunnel** *tunnel-id* [**session** *session-id*]]

If you use this command without any optional constructs, all the sessions in all the tunnels are cleared. If you specify a tunnel, you can also specify a session. If you do not specify a session, all the sessions in the specified tunnel are cleared.

Configuration Examples

The following example creates an L2F peer named `hssi_3_0.chi.core.isp.net`. The relationship with the peer is established as home gateway, and the peer is given an alias (domain name) of `isp.net`. The parameters for tunnels from the peer are then configured in L2F configuration mode. Finally, an Asynchronous Transfer Mode (ATM) circuit is created and bound to the peer using the peer's domain name as follows:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#l2f-peer name hssi_3_0.chi.core.isp.net media udp-ip local
10.0.0.1 remote 10.0.0.2
[local]RedBack(config-l2f)#function home-gateway
[local]RedBack(config-l2f)#domain isp.net
[local]RedBack(config-l2f)#session-auth pap
[local]RedBack(config-l2f)#max-sessions 2000
[local]RedBack(config-l2f)#police rate 1000 burst 15000
[local]RedBack(config-l2f)#retry 4
[local]RedBack(config-l2f)#timeout 6
[local]RedBack(config-l2f)#exit
[local]RedBack(config)#port atm 3/0
[local]RedBack(config-port)#atm pvc 0 1 profile ubr encapsulation ppp
[local]RedBack(config-pvc)#bind session isp.net local
```

Part 7

Security

Configuring IPSec

This chapter describes the tasks related to configuring IP Security (IPSec) features through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "IPSec Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Note You must have an IPSec/Compression Transform Engine (TE) module installed in your Subscriber Management System (SMS) device to configure and use IPSec features. The module may be configured to provide IPSec and Point-to-Point Protocol (PPP) compression service. The module has two chipsets onboard, each one of which may be used for either IPSec or PPP compression. These chipsets are modeled as internal "ports" in the AOS, and are configured the same way a traditional I/O card port is configured.

Overview

There are two approaches when configuring IPSec for an SMS device—the peer approach (which limits the routing of all the subscriber's traffic) and the policy approach (which limits the routing of only specified portions of the subscriber's traffic).

In the peer approach, all traffic coming from or going to the subscriber must be routed to the specified IPSec peer. The subscriber is not allowed to have both secure traffic routed to the specified IPSec peer and other traffic routed normally.

In the policy approach, traffic that matches any of the configured source/destination specifications must be routed to the IPSec peer. These specifications constitute a policy. All traffic to or from the subscriber that does not match the policy is routed normally.

An analysis of the advantages and disadvantages of each of these two approaches is beyond the scope of this document. Both strategies provide security. The difference is in how much responsibility and control the destination wants to have over the subscriber's traffic.

If you opt for the policy approach, you must configure at least one policy. Regardless of whether you choose policy or peer, you must configure at least one peer.

Once you have decided to use either the peer or policy approach, you must also consider the issue of manual versus Internet Key Exchange (IKE)-negotiated Security Associates (SA), because it makes a difference in the proposals that you must configure. Manual SAs require just one IPSec proposal with a corresponding key configuration. IKE-negotiated SAs require at least one IPSec proposal and at least one IKE proposal.

You can have both policy and peer elements in a single IPSec configuration database. The appropriate elements are used based on the subscriber configuration.

Features and Limitations

The Redback implementation of IPSec conforms to the standards specified in the following RFCs:

- RFC 2401, *Security Architecture for the Internet Protocol*
- RFC 2402, *IP Authentication Header (AH)*
- RFC 2406, *IP Encapsulating Security Payload (ESP)*
- RFC 2408, *Internet Security Association and Key Management Protocol (ISAKMP)*
- RFC 2409, *The Internet Key Exchange*

The Redback implementation supports the following:

- Subscriber binding (no other binding types are supported at this time)
- Manual and IKE-negotiated proposals (IKE is limited to pre-shared keys)
- One IPSec option—Perfect Forward Secrecy (PFS)
- Tunnel encapsulation mode for proposals (transport mode is not supported at this time)
- Authentication Header (AH) and Encapsulating Security Payload (ESP) hash algorithms
- ESP cipher algorithms
- Seven levels of IPSec peer debugging messages
- Ability to clear IPSec peers and bring them up again when needed with the latest configuration using a single command

There are five AOS configuration modes associated with IPSec; all of them accessible from context configuration mode. See Chapter 2, “Understanding the User Interface,” for information on the command-line interface (CLI) commands used to enter each configuration mode and the system prompt that is displayed in each mode.

Configuration Tasks

To configure IPSec, first determine whether you will use the peer approach or the policy approach, and then perform the appropriate tasks as described in the following sections:

- Configure the TE Port
- Configure an IPSec Policy
- Change the Default IPSec Peer Configuration
- Configure an IPSec Peer
- Configure an IPSec Proposal
- Configure an IKE Proposal
- Configure a Key Structure
- Configure the Subscribers
- Clear IPSec Peers
- Display IPSec Information
- Enable IPSec Debugging

Configure the TE Port

See Chapter 9, "Configuring Common Port, Circuit, and Channel Parameters" for information on providing a description for, and enabling each port on the IPSec/Compression TE module.

Configure an IPSec Policy

To create an IPSec policy and enter IPSec policy configuration mode, enter the following command in context configuration mode:

ipsec policy name *pol-name*

You can configure multiple IPSec policies within a context.

Once in IPSec policy configuration mode, enter the following command to define the IP address parameters of the policy:

tunnel ip *src-addr src-netmask dst-addr dst-netmask* **name** *peer-name*

The *src-addr* argument is the source IP address of packets arriving from and destined for the subscriber. The *src-netmask* argument is an indication of which bits in the *src-addr* argument are significant for purposes of matching. The *dst-addr* argument is the destination IP address of packets arriving from and destined for the subscriber. The *dst-netmask* argument is an indication of which bits in the *dst-addr* argument are significant for purposes of matching. The **name** *peer-name* construct is the name of the IPSec peer to be used when the IP source or destination address matches the specified criteria.

You can enter this command multiple times in a policy configuration, each with different IP address parameters and IPSec peer designations. Any traffic to or from the subscriber not included in any of the **tunnel ip** command entries for the policy to which the subscriber is bound is routed normally.

Change the Default IPSec Peer Configuration

When you create a new IPSec peer, there are factory default settings that are applied and remain in effect unless you change them in the configuration of the individual peer. However, you can change the default settings that are applied to new peers by entering IPSec peer configuration mode via the **ipsec peer default** command, and then issuing IPSec peer configuration mode commands with the desired default values.

Table 27-1 lists the factory default settings for new IPSec peers that can be changed by entering IPSec peer configuration mode via the **ipsec peer default** command.

Table 27-1 IPSec Peer Factory Default Settings

Setting	IPSec Peer Configuration Mode Command	Factory Default Value
Name of an IPSec proposal that can be used with the peer	**proposal crypto**	None
Name of an IKE proposal that can be used with the peer	**proposal ike**	None
IPSec options that can be used with the peer	**ipsec options**	PFS enabled
Mode to be used when negotiating the Internet Security Association and Key Management Protocol (ISAKMP)	**ipsec mode**	Main mode
ISAKMP group to be used to construct key material in a manual tunnel	**ipsec pfs-group**	Group 1 (768-bit MODP group)
Soft lifetime in kilobytes for a manual tunnel	**ipsec lifetime soft kbytes**	1,800 kilobytes
Soft lifetime in seconds for a manual tunnel	**ipsec lifetime soft seconds**	3,240 seconds
Hard lifetime in kilobytes for a manual tunnel	**ipsec lifetime hard kbytes**	2,000 kilobytes
Hard lifetime in seconds for a manual tunnel	**ipsec lifetime hard seconds**	3,600 seconds
ISAKMP group to be used to construct key material in an IKE-negotiated tunnel	**ike group**	Group 1 (768-bit MODP group)
Soft lifetime in kilobytes for an IKE-negotiated tunnel	**ike lifetime soft kbytes**	1,800 kilobytes
Soft lifetime in seconds for an IKE-negotiated tunnel	**ike lifetime soft seconds**	3,240 seconds
Hard lifetime in kilobytes for an IKE-negotiated tunnel	**ike lifetime hard kbytes**	2,000 kilobytes
Hard lifetime in seconds for an IKE-negotiated tunnel	**ike lifetime hard seconds**	3,600 seconds
Authentication method used when invoking IKE	**ike auth**	Pre-shared keys

Table 27-1 IPSec Peer Factory Default Settings

Setting	IPSec Peer Configuration Mode Command	Factory Default Value
Pre-shared key used in IKE authentication	**ike pre-shared-key**	None

Once the default settings have been changed, the new values are applied to any new peer that is created, unless the values are changed in the configuration for an individual peer. The configuration of an individual peer always overrides the default peer settings.

To enter IPSec peer configuration mode for purposes of changing the default settings, enter the following command in context configuration mode:

ipsec peer default

Any IPSec peer configuration commands listed in Table 27-1 that you subsequently enter will change the configuration of the default peer rather than changing the configuration of a specific peer.

Configure an IPSec Peer

To create an IPSec peer and enter IPSec peer configuration mode, enter the following command in context configuration mode:

ipsec peer name *peer-name*

Once in IPSec peer configuration mode, there are numerous parameters that you can configure, as described in the remainder of this section.

To define the local address of the peer, enter the following command:

ip-address local *local-ip-addr*

To define the remote address of the peer, enter the following command:

ip-address remote *remote-ip-addr*

To specify that a specific IPSec proposal can be used with the peer, enter the following command:

proposal crypto *prop-name*

To specify that a specific IKE proposal can be used with this peer, enter the following command:

proposal ike *ike-prop-name*

You can enter this command multiple times, once for each of the proposals that IKE can consider during negotiation.

To enable IPSec options, enter the following command:

ipsec options pfs

Note At this time, PFS is the only supported IPSec option.

To specify the mode to be used when negotiating Internet Security Association and Key Management Protocol (ISAKMP) for both IPSec and IKE, enter the following command:

ipsec mode {**main** | **aggressive**}

The **main** and **aggressive** keywords represent the two available modes.

To specify the ISAKMP group that is to be used by the Diffie-Hellman key exchange to construct key material for an IPSec Security Association (SA), enter the following command:

ipsec pfs-group *group-num*

The *group-num* argument is an integer from 1 to 5 representing one of the valid ISAKMP groups. See the full description of this command in the "IPSec Commands" chapter of the *Access Operating System (AOS) Command Reference* publication for more information on the valid groups.

To specify how long an SA can remain active before renegotiation for a new SA begins, enter either (or both) of the following commands. One of the commands allows you to specify the limit in terms of time and the other in terms of kilobytes of data:

ipsec lifetime soft seconds *seconds*

ipsec lifetime soft kbytes *kbytes*

When a soft lifetime is reached, that only means that renegotiation begins. The IPSec tunnel remains intact and secure traffic can continue to be transmitted.

To specify how long an SA can remain active before transmittal of secure traffic is shut down, enter either (or both) of the following commands. One of the commands allows you to specify the limit in terms of time and the other in terms of kilobytes of data:

ipsec lifetime hard seconds *seconds*

ipsec lifetime hard kbytes *kbytes*

When a hard lifetime is reached, renegotiation for a new SA continues, but secure traffic can no longer be transmitted.

To specify the ISAKMP group that is to be used by the Diffie-Hellman key exchange to construct key material for an IKE SA, enter the following command:

ike group *group-num*

The *group-num* argument is an integer from 1 to 5 representing one of the valid ISAKMP groups. See the full description of this command in the "IPSec Commands" chapter of the *Access Operating System (AOS) Command Reference* publication for more information on the valid groups.

To specify how long an IKE SA can remain active before renegotiation for a new SA begins, enter either (or both) of the following commands. One of the commands allows you to specify the limit in terms of time and the other in terms of kilobytes of data:

ike lifetime soft seconds *seconds*

ike lifetime soft kbytes *kbytes*

When a soft lifetime is reached, that only means that renegotiation begins. The IPSec tunnel remains intact and secure traffic can continue to be transmitted.

To specify how long an IKE SA can remain active before transmittal of secure traffic is shut down, enter either (or both) of the following commands. One of the commands allows you to specify the limit in terms of time and the other in terms of kilobytes of data:

ike lifetime hard seconds *seconds*

ike lifetime hard kbytes *kbytes*

When a hard lifetime is reached, renegotiation for a new SA continues, but secure traffic can no longer be transmitted.

To specify the authentication method used when invoking IKE, enter the following command:

ike auth pre-shared-keys

Note Currently, pre-shared keys is the only authentication method supported.

To define the pre-shared key, enter the following command:

ike pre-shared-key {**string** *string* | **hex** *binary*}

The key can be expressed as either an ASCII string or a binary expression.

To configure the source and destination addresses for a local Internet key exchange (IKE) Security Association (SA) subnet, enter the following command:

ike sa_subnet {*source source-wildcard* | **any**} {*destination destination-wildcard* | **any**}

Configure an IPSec Proposal

To create an IPSec proposal and enter IPSec proposal configuration mode, enter the following command in context configuration mode:

ipsec proposal crypto name *prop-name*

You can configure multiple proposals in a context.

Once in IPSec proposal configuration mode, you can set a number of parameters for the proposal as described in the remainder of this section.

To define the encapsulation mode for the proposal, enter the following command:

encapsulation-mode tunnel

Note Currently, tunnel mode is the only encapsulation mode supported.

To specify the hash algorithm to use for the authentication header (AH) in the proposal, enter the following command:

ah hash {**hmac-md5** | **hmac-md5-96** | **hmac-sha** | **hmac-sha-96** | **none**} [**key** *key-name*]

The **hmac-md5**, **hmac-md5-96**, **hmac-sha**, and **hmac-sha-96** keywords represent hash algorithms. You can also choose the **none** keyword if you do not want any hash algorithm applied. Use the optional **key** *key-name* construct for manual proposals only.

To specify the hash algorithm to use for Encapsulating Security Payload (ESP) in this proposal, enter the following command:

esp hash {**hmac-md5** | **hmac-md5-96** | **hmac-sha** | **hmac-sha-96** | **none**} [**key** *key-name*]

The **hmac-md5**, **hmac-md5-96**, **hmac-sha**, and **hmac-sha-96** keywords represent hash algorithms. You can also choose the **none** keyword if you do not want any hash algorithm applied. Use the optional **key** *key-name* construct for manual proposals only.

To specify the cipher algorithm to use for ESP in this proposal, enter the following command:

esp cipher {**des-cbc** | **3des-cbc** | **des-ecb** | **3des-ecb** | **none**} [**key** *key-name*]

The **des-cbc**, **3des-cbc**, **des-ecb**, and **3des-ecb** keywords represent cipher algorithms. You can also choose the **none** keyword if you do not want any cipher algorithm applied. Use the optional **key** *key-name* construct for manual proposals only.

Configure an IKE Proposal

To create an IKE proposal and enter IKE proposal configuration mode, enter the following command in context configuration mode:

ipsec proposal ike name *ike-name*

You can configure multiple IKE proposals per context. IKE uses the list of configured IKE proposals to negotiate an SA between the two ends of the IPSec tunnel.

Once in IKE proposal configuration mode, you can specify cipher and hash algorithms for the proposal. To specify the cipher algorithm to use for the IKE proposal, enter the following command:

cipher {**des-cbc** | **3des-cbc** | **des-ecb** | **3des-ecb** | **none**}

The **des-cbc**, **3des-cbc**, **des-ecb**, and **3des-ecb** keywords represent cipher algorithms. You can also choose the **none** keyword if you do not want any cipher algorithm applied.

To specify the hash algorithm to use for the IKE proposal, enter the following command:

hash {**md5** | **sha** | **none**}

The **md5** and **sha** keywords represent hash algorithms. You can also choose the **none** keyword if you do not want any hash algorithm applied.

Configure a Key Structure

All manual proposals must reference a configured key structure. To create a key structure and enter key configuration mode, enter the following command in context configuration mode:

ipsec key name *key-name*

You can have multiple key structures configured in a context.

Once in key configuration mode, you can define the parameters of the key structure. To define the Security Parameter Index (SPI) used for the inbound SA of a manual tunnel, enter the following command:

spi in *num*

To define the SPI used for the outbound SA of a manual tunnel, enter the following command:

spi out *num*

To define the key used for the inbound SA of a manual tunnel, enter the following command:

in [**string** *string* | **hex** *binary*]

The key can be expressed as either an ASCII string or a binary expression.

To define the key used for the outbound SA of a manual tunnel, enter the following command:

out [**string** *string* | **hex** *binary*]

The key can be expressed as either an ASCII string or a binary expression.

Configure the Subscribers

Subscribers can be bound to an IPSec policy. To bind a subscriber to an IPSec policy, enter the following command in subscriber configuration mode:

ipsec tunnel policy *pol-name*

When bound to a policy, the traffic that must go through an IPSec peer is defined by the terms of the policy. All other traffic to or from the subscriber is routed normally.

Clear IPSec Peers

To clear IPSec peers and bring them back up again using the latest configuration as soon as they are needed, enter the following command in administrator exec mode:

clear ipsec peer [**name** *peer-name* | **id** *tunnel-id*] [**no-restart**]

You can specify an individual peer by either the name of the peer or the tunnel identification, or you can clear all peers by not specifying a particular peer at all. The optional **no-restart** keyword allows you to specify that you do not want the peers brought back up after being cleared. In that case, a peer does not come back up until a new subscriber session requires it or until the **clear ipsec peer** command is issued again, without the **no-restart** keyword.

Display IPSec Information

Several **show** commands are available to display IPSec information. To display IPSec information, enter any of the following commands in administrator exec mode:

show hardware *slot*/*port*

show ipsec peer {**name** *peer-name* | **tunnel-id** *tunnel-id*}

show ipsec peer stats {**global** | **name** *peer-name* | **tunnel-id** *tunnel-id*}

show memory te

show te cpu

show te performance

show te ps

show te time

See the "IPSec Commands" and "System Monitoring and Testing Commands" chapters in the *Access Operating System (AOS) Command Reference* publication for information on and examples of the output of these **show** commands.

In addition, the following common port commands also display useful information:

show port diag *slot*/*port*

show port info [*slot*/*port*]

show port table

See the "Common Port, Circuit, and Channel Commands" chapter of the *Access Operating System (AOS) Command Reference* publication for information on and examples of the output of these **show** commands.

Enable IPSec Debugging

There are seven levels of IPSec peer debugging messages that can be enabled. The levels are numbered 1 to 7, with 7 being the most detailed and 1 being the least detailed. To enable IPSec peer debugging messages, enter the following command in administrator exec mode:

debug ipsec peer [*value*]

To enable IKE debugging messages, enter the following command in administrator exec mode:

debug ipsec ike

Configuration Examples

The following is an example using manually configured tunnels:

```
[local]RedBack(config)#system hostname hydro2@local
[local]RedBack(config)#aaa default-domain local username-format username @
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#ipsec peer name goodname
[local]RedBack(config-ipsec-peer)#ip-address local 10.13.16.58
[local]RedBack(config-ipsec-peer)#ip-address remote 155.53.36.12
[local]RedBack(config-ipsec-peer)#proposal crypto bestcrypto
[local]RedBack(config-ipsec-peer)#exit
[local]RedBack(config-ctx)#ipsec proposal crypto name bestcrypto
[local]RedBack(config-ipsec-prop)#esp hash hmac-md5 key keyname1
[local]RedBack(config-ipsec-prop)#esp cipher des-cbc key keyname2
[local]RedBack(config-ipsec-prop)#exit
[local]RedBack(config-ctx)#ipsec key name keyname1
[local]RedBack(config-ipsec-key)#spi in 512
[local]RedBack(config-ipsec-key)#spi out 512
[local]RedBack(config-ipsec-key)#in string 12345678
[local]RedBack(config-ipsec-key)#out string 12345678
[local]RedBack(config-ipsec-key)#exit
[local]RedBack(config-ctx)#ipsec key name keyname2
[local]RedBack(config-ipsec-key)#spi in 512
```

```
[local]RedBack(config-ipsec-key)#spi out 512
[local]RedBack(config-ipsec-key)#in string 12345678
[local]RedBack(config-ipsec-key)#out string 12345678
[local]RedBack(config-ipsec-key)#exit
[local]RedBack(config-ctx)#ipsec policy name policyname
[local]RedBack(config-ipsec-pol)#tunnel IP 12.1.1.1 0.255.255.255 155.53.36.12
0.0.0.255 name ike-name
[local]RedBack(config-ipsec-pol)#exit
[local]RedBack(config-ctx)#interface adm0
[local]RedBack(config-if)#ip address 10.13.16.58 255.255.248.0
[local]RedBack(config-if)#ip arp arpa
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#interface ppp
[local]RedBack(config-if)#ip address 12.1.1.2 255.255.255.0
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#interface eth7/0
[local]RedBack(config-if)#ip address 11.1.1.1 255.255.255.0
[local]RedBack(config-if)#ip arp arpa
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#interface ssh
[local]RedBack(config-if)#ip address 155.53.34.99 255.255.255.0
[local]RedBack(config-if)#ip arp arpa
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#subscriber name zt4
[local]RedBack(config-sub)#password two
[local]RedBack(config-sub)#outbound password one
[local]RedBack(config-sub)#ip address 12.1.1.1
[local]RedBack(config-sub)#ipsec tunnel policy policyname
[local]RedBack(config-sub)#exit
[local]RedBack(config-ctx)#subscriber name zt42
[local]RedBack(config-sub)#password two
[local]RedBack(config-sub)#outbound password one
[local]RedBack(config-sub)#ip address 12.1.1.3
[local]RedBack(config-sub)#ipsec tunnel policy policyname
[local]RedBack(config-sub)#exit
[local]RedBack(config-ctx)#ip route 0.0.0.0 0.0.0.0 10.13.23.254 adm0
[local]RedBack(config-ctx)#ip route 155.53.36.0 255.255.255.0 10.13.23.254 adm0
[local]RedBack(config-ctx)#exit
[local]RedBack(config)#atm profile ubr
[local]RedBack(config-atmpro)#shaping ubr
[local]RedBack(config-atmpro)#exit
[local]RedBack(config)#port ethernet 0/0
[local]RedBack(config-port)#no bind int
[local]RedBack(config-port)#exit
[local]RedBack(config)#port te 4/1
[local]RedBack(config-port)#no shut
[local]RedBack(config-port)#exit
[local]RedBack(config)#port atm 6/0
[local]RedBack(config-port)#atm pvc 1 1 profile ubr encapsulation ppp
[local]RedBack(config-pvc)#bind authentication chap pap
[local]RedBack(config-pvc)#exit
[local]RedBack(config-port)#no shut
```

```
[local]RedBack(config-port)#exit
[local]RedBack(config)#port atm 6/1
[local]RedBack(config-port)#atm pvc 1 1 profile ubr encapsulation ppp
[local]RedBack(config-pvc)#bind authentication chap pap
[local]RedBack(config-pvc)#exit
[local]RedBack(config-port)#no shut
[local]RedBack(config-port)#exit
[local]RedBack(config)#port ethernet 7/1
[local]RedBack(config-port)#bind interface adm0 local
[local]RedBack(config-port)#ip host 10.13.23.254 00:e0:2b:90:b6:00
[local]RedBack(config-port)#no shut
[local]RedBack(config-port)#end
```

The following is an example using IKE-negotiated tunnel configuration:

```
[local]RedBack(config)#system hostname hydro2@local
[local]RedBack(config)#aaa default-domain local username-format username @
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#ipsec peer name ike-name
[local]RedBack(config-ipsec-peer)#ip-address local 10.13.16.58
[local]RedBack(config-ipsec-peer)#ip-address remote 155.53.36.12
[local]RedBack(config-ipsec-peer)#proposal crypto ike-tunnel
[local]RedBack(config-ipsec-peer)#proposal ike ike-prop
[local]RedBack(config-ipsec-peer)#ike group 2
[local]RedBack(config-ipsec-peer)#ike lifetime soft kbytes 1000
[local]RedBack(config-ipsec-peer)#ike lifetime hard kbytes 2000
[local]RedBack(config-ipsec-peer)#ike lifetime soft seconds 1800
[local]RedBack(config-ipsec-peer)#ike lifetime hard seconds 3600
[local]RedBack(config-ipsec-peer)#ipsec lifetime soft kbytes 1000
[local]RedBack(config-ipsec-peer)#ipsec lifetime hard kbytes 2000
[local]RedBack(config-ipsec-peer)#ipsec lifetime soft seconds 1800
[local]RedBack(config-ipsec-peer)#ipsec lifetime hard seconds 3600
[local]RedBack(config-ipsec-peer)#ike auth pre-shared-keys
[local]RedBack(config-ipsec-peer)#ike pre-shared-key string ike-key
[local]RedBack(config-ipsec-peer)#exit
[local]RedBack(config-ctx)#ipsec proposal crypto name ike-tunnel
[local]RedBack(config-ipsec-prop)#encapsulation-mode tunnel
[local]RedBack(config-ipsec-prop)#ah hash none
[local]RedBack(config-ipsec-prop)#esp hash hmac-md5-96
[local]RedBack(config-ipsec-prop)#esp cipher des-cbc
[local]RedBack(config-ipsec-prop)#exit
[local]RedBack(config-ctx)#ipsec proposal ike name ike-prop
[local]RedBack(config-ike-prop)#hash md5
[local]RedBack(config-ike-prop)#cipher des-cbc
[local]RedBack(config-ike-prop)#exit
[local]RedBack(config-ctx)#ipsec policy name policyname
[local]RedBack(config-ipsec-pol)#tunnel IP 12.1.1.1 0.255.255.255 155.53.36.12
0.0.0.255 name ike-name
[local]RedBack(config-ipsec-pol)#exit
[local]RedBack(config-ctx)#interface adm0
[local]RedBack(config-if)#ip address 10.13.16.58 255.255.248.0
[local]RedBack(config-if)#ip arp arpa
```

```
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#interface ppp
[local]RedBack(config-if)#ip address 12.1.1.2 255.255.255.0
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#interface eth7/0
[local]RedBack(config-if)#ip address 11.1.1.1 255.255.255.0
[local]RedBack(config-if)#ip arp arpa
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#interface ssh
[local]RedBack(config-if)#ip address 155.53.34.99 255.255.255.0
[local]RedBack(config-if)#ip arp arpa
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#subscriber name zt4
[local]RedBack(config-sub)#password two
[local]RedBack(config-sub)#outbound password one
[local]RedBack(config-sub)#ip address 12.1.1.1
[local]RedBack(config-sub)#ipsec tunnel policy policyname
[local]RedBack(config-sub)#exit
[local]RedBack(config-ctx)#ip route 0.0.0.0 0.0.0.0 10.13.23.254 adm0
[local]RedBack(config-ctx)#ip route 155.53.36.0 255.255.255.0 10.13.23.254 adm0
[local]RedBack(config-ctx)#exit
[local]RedBack(config)#atm profile ubr
[local]RedBack(config-atmpro)#shaping ubr
[local]RedBack(config-atmpro)#exit
[local]RedBack(config)#port ethernet 0/0
[local]RedBack(config-port)#no bind int
[local]RedBack(config-port)#exit
[local]RedBack(config)#port te 4/1
[local]RedBack(config-port)#no shut
[local]RedBack(config-port)#exit
[local]RedBack(config)#port atm 6/0
[local]RedBack(config-port)#atm pvc 1 1 profile ubr encapsulation ppp
[local]RedBack(config-pvc)#bind authentication chap pap
[local]RedBack(config-pvc)#exit
[local]RedBack(config-port)#exit
[local]RedBack(config)#port atm 6/1
[local]RedBack(config-port)#atm pvc 1 1 profile ubr encapsulation ppp
[local]RedBack(config-pvc)#bind authentication chap pap
[local]RedBack(config-pvc)#exit
[local]RedBack(config-port)#exit
[local]RedBack(config)#port ethernet 7/1
[local]RedBack(config-port)#bind interface adm0 local
[local]RedBack(config-port)#ip host 10.13.23.254 00:e0:2b:90:b6:00
[local]RedBack(config-port)#exit
[local]RedBack(config)#logging console
[local]RedBack(config)#line console
[local]RedBack(config-line)#exit
[local]RedBack(config)#line tty 1
[local]RedBack(config-line)#exit
[local]RedBack(config)#line tty 2
[local]RedBack(config-line)#exit
[local]RedBack(config)#line tty 3
```

```
[local]RedBack(config-line)#exit
[local]RedBack(config)#line tty 4
[local]RedBack(config-line)#exit
[local]RedBack(config)#line tty 5
[local]RedBack(config-line)#exit
[local]RedBack(config)#line tty 6
[local]RedBack(config-line)#exit
[local]RedBack(config)#line tty 7
[local]RedBack(config-line)#exit
[local]RedBack(config)#line tty 8
[local]RedBack(config-line)#end
```

Part 8

IP Services

Configuring DNS

This chapter provides an overview of the Domain Name System (DNS) and describes the tasks involved in configuring DNS features through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "DNS Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

The DNS maps hostnames to IP addresses, enabling users to access devices using hostnames, instead of IP addresses. When an AOS command refers to a hostname, the Subscriber Management System (SMS) device consults the local host table for mappings. If the information is not in the host table, the SMS device generates a DNS query to resolve the hostname. DNS is enabled on a per-context basis, with one domain name allowed per context.

Configuration Tasks

To configure the DNS, perform the tasks described in the following sections:

- Enable DNS
- Provide a Domain Name
- Configure a Connection to a DNS Server
- Place Static Entries in the Local Host Table
- Show DNS Information
- Clear Hostname-to-IP Address Mappings
- Enable DNS Debugging Messages

Enable DNS

To enable the DNS, enter the following context configuration command:

ip domain-lookup

Provide a Domain Name

To configure a domain name for a context running DNS, enter the following context configuration command:

ip domain-name *dom-name*

The *dom-name* argument is the name by which the domain will be known. You may have only one domain name per context.

Configure a Connection to a DNS Server

To configure the DNS server IP address to which the context connects, enter the following command in context configuration mode. You can also configure a secondary (backup) DNS server.

ip name-servers *ip-address* [*ip-address*]

The first *ip-address* argument is the primary server and the second *ip-address* argument is the secondary server. DNS servers are queried in the order configured: primary followed by secondary.

Place Static Entries in the Local Host Table

Optionally, you can enter static hostname-to-IP address mappings in the local host table, in addition to having DNS perform dynamic resolution. To configure static mappings, enter the following command in context configuration mode:

ip localhost *hostname ip-address*

Up to 64 static entries can be included in the host table for the context. The SMS device always consults the host table prior to generating a DNS lookup query. Entries created with the **ip localhost** command are never aged out. Specifying a new IP address for an existing hostname removes the previously configured IP address for that hostname.

Show DNS Information

To view hostname-to-IP address mappings stored in the local host table in the current context, enter the following command in operator exec mode:

show ip localhosts [*hostname*]

When no arguments or keywords are specified, this command lists all hostname-to-IP address mappings stored in the local host table for the context. When a hostname is specified suing the *hostname* argument, only the single entry matching the hostname is displayed.

Clear Hostname-to-IP Address Mappings

To clear DNS hostname-to-IP address mappings, enter the following command in administrator exec mode:

clear ip localhosts [*hostname*]

Using this command with no arguments or keywords clears the entire local host table. When a hostname is specified using the *hostname* argument, only the single entry matching the hostname is deleted.

Enable DNS Debugging Messages

To enable DNS debugging messages, enter the following command in administrator exec mode:

debug ip dns

Configuration Examples

The following example configures the `redback.com` domain for the `local` context and configures a connection to a remote DNS server at IP address `155.53.130.200`. The DNS is enabled through the **ip domain-lookup** command.

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#ip domain-lookup
[local]RedBack(config-ctx)#ip domain-name redback.com
[local]RedBack(config-ctx)#ip name-server 155.53.130.200
```

Configuring DHCP

This chapter provides an overview of the Dynamic Host Configuration Protocol (DHCP) and describes the tasks involved in configuring DHCP features through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "DHCP Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

In some network installations, the DHCP is used to dynamically configure IP address information for subscriber hosts. The DHCP can assign a variety of information including the host's IP address, subnet mask, and default gateway address.

The Subscriber Management System (SMS) device acts as a DHCP relay (or a Bootstrap Protocol [BOOTP] relay), as a DHCP server, or both. The DHCP is typically used in conjunction with RFC 1483- or RFC 1490-encapsulated circuits, because Point-to-Point Protocol (PPP) circuits have other means by which to dynamically acquire IP addressing information. The Access Operating System (AOS) is compliant with both the Internet Draft, *DHCP Relay Agent Information Option*, draft-ietf-dhc-agent-options-12.txt, and RFC 2132, *DHCP Options and BOOTP Vendor Extensions*.

Configuration Tasks

To configure DHCP, perform the tasks described in the following sections:

- Enable DHCP Relay for a Context
- Enable DHCP Relay Options
- Make Interfaces Eligible for Relay of DHCP Packets

- Configure Hosts to Use DHCP
- Preserve DHCP State Information in Nonvolatile Memory
- Enable the DHCP Server
- Configure a Secondary DHCP Server
- Configure DHCP Server Parameters
- Display DHCP Information
- Display DHCP Server Information
- Enable DHCP Debugging Messages

Enable DHCP Relay for a Context

To enable the relay of DHCP messages and to configure the IP address of a DHCP server, enter the following command in context configuration mode:

dhcp relay server *server-address*

where the *server-address* argument is the IP address of the DHCP server.

The SMS device examines all responses from the server and notes the bindings between the assigned IP address, the requesting Ethernet Media Access Control (MAC) address, and the circuit from which the request was received. The result is a behavior similar to that of secured-Address Resolution Protocol (ARP). Because an entry is automatically placed in the SMS host table for this binding, the need to use ARP for the binding is eliminated. This ensures that the address cannot be spoofed and that traffic cannot be redirected.

To enable communications with more than one server (with a limit per context of five), you can enter the **dhcp relay server** command multiple times, each specifying the address of a different server. When multiple DHCP servers are configured, a DHCP request is sent to every configured server. There is no round-robin or backup implementation; all servers receive a request and can respond.

Enable DHCP Relay Options

To enable the sending of DHCP options in all DHCP packets that are relayed by the SMS device, enter the following command in context configuration mode:

dhcp relay option

When you enable this feature, the AOS adds relay options to all DHCP requests that are forwarded by the SMS device on behalf of a DHCP client. The DHCP relay options are described in the *DHCP Relay Agent Information Option* Internet draft.

The AOS can use DHCP relay options to help track DHCP requests. Some options can also enhance the DHCP server's function. For example, an **agent remote id** option contains the ASCII username associated with the circuit and the DHCP server can use this to make address allocation decisions. For AOS tracking purposes, the **agent circuit id** option contains a 32-bit number that identifies the circuit through which a subscriber has connected.

Make Interfaces Eligible for Relay of DHCP Packets

The SMS device does not relay DHCP messages received on an interface, unless the interface has been made eligible to relay DHCP packets. To make an interface eligible, enter the following command in interface configuration mode:

dhcp relay size *addresses*

where the *addresses* argument is the maximum number of unique IP addresses to be assigned by the DHCP server for hosts on the same subnet as the configured interface. This value is used by the SMS device for load balancing the use of addresses from multiple pools. It is not enforced as a strict limit. Strict limits cannot be imposed by a DHCP relay; they can only be applied by the DHCP server.

The **dhcp relay size** command does not apply to loopback interfaces.

Configure Hosts to Use DHCP

In the case of static IP addressing, the subscriber host IP address is configured directly into the corresponding subscriber records. When addressing is dynamic using DHCP, a DHCP configuration command is required in the subscriber record, rather than an IP address. Multiple addresses can be acquired by hosts associated with one subscriber circuit.

To configure hosts associated with a subscriber record to use DHCP to dynamically acquire address information, enter the following command in subscriber configuration mode:

dhcp max-addrs *num-addresses*

where the *num-addresses* argument is the maximum number of unique IP addresses expected to be assigned by the DHCP server to hosts associated with the subscriber circuit.

This command is also helpful for load balancing the use of addresses from multiple pools. It is not enforced as a strict limit. Strict limits cannot be imposed by a DHCP relay; they can only be applied by the DHCP server.

Note If for some reason a subscriber record contains both a **dhcp max-addrs** command and one or more static IP host addresses configured with the **ip address** command, the static IP addresses always take precedence. In other words, the associated circuit is bound to an interface on the basis of the static IP addresses. If no static addresses are configured and a **dhcp max-addrs** command is present in the record, then the associated circuit is bound to the first available interface with capacity for this subscriber.

Preserve DHCP State Information in Nonvolatile Memory

You can configure an SMS device to preserve the secured-ARP state for DHCP circuits across system restarts so that communications can be immediately restored to end stations. This involves storing DHCP secured-ARP state information to nonvolatile static RAM (SRAM).

If you do not configure your system to preserve DHCP secured-ARP state information in nonvolatile memory, you risk losing the information in the event of a system restart.

The secured-ARP table contains IP-address-to-circuit handle mappings that enable the AOS to direct each ARP request to the appropriate single circuit. An ARP request is not even processed, unless the source IP address and the circuit handle correspond to an entry in the secured ARP table.

With DHCP, the secured-ARP table is automatically populated by examining all DHCP server responses. Subscriber end stations begin using IP addresses assigned via DHCP. If the system is restarted, the secured ARP state can be lost. Those subscriber end stations cannot communicate, because the SMS device no longer has knowledge of the circuit binding or IP address-to-circuit mappings.

Note This issue does not exist if the subscriber lines are utilizing static addressing or are running Point-to-Point Protocol (PPP) or PPP over Ethernet (PPPoE).

Storing the secured-ARP state information in nonvolatile SRAM preserves it in the event of a system restart. If the system restarts while DHCP-added secured-ARP entries exist, the following recovery steps are performed:

1. When the **dhcp preserve-state** command is read from the configuration file, entries on SRAM are recovered for binding purposes.
2. When a circuit comes up, the SRAM contents are checked for an entry with a matching circuit handle. If a matching circuit handle is found, the circuit is bound to the interface containing the saved IP address within the saved context.
3. Once the bind is complete, a secured-ARP entry is added and authentication, authorization, and accounting (AAA) is updated with the IP address installed. Communication is then restored to the subscriber end station.

The DHCP secured-ARP information is also preserved when an I/O module is replaced (provided the I/O module supports hot swap).

To utilize this feature, the SMS device must have a PCMCIA slot available in an approved, battery backed-up SRAM PCMCIA card. A 2-MB module is sufficient to hold over 100,000 secured-ARP table entries.

To configure the SMS device to store DHCP secured-ARP state information to nonvolatile memory, perform the tasks in the following sections:

- Format the SRAM PCMCIA Card
- Enable DHCP Preserve-State

Format the SRAM PCMCIA Card

To format your SRAM PCMCIA card to function as nonvolatile storage for DHCP secured-ARP information, enter the following command in administrator exec mode:

format *device* **dhcp-secured-arp**

where the *device* argument is the name of the SRAM PCMCIA card you are formatting (/pcmcia0 or /pcmcia1). If the card already contains a format, you are prompted to confirm your request. If the PCMCIA card is in use as SRAM (for example, if DHCP preserve-state is already enabled for the module), the **format** command gives you an error. You also get an error if you attempt to format a flash memory card using the **dhcp-secured-arp** keyword.

Note If the formatted SRAM card is ever removed during operation, DHCP secured-ARP state information cannot be preserved until the SRAM card is reinserted. At that time, the current DHCP state is immediately written in its entirety to the SRAM card. This synchronizes the card with the running system.

Enable DHCP Preserve-State

Enabling DHCP preserve-state instructs the SMS device to store DHCP secured-ARP state information to the formatted SRAM card. To enable DHCP preserve-state, enter the following command in global configuration mode:

dhcp preserve-state

With this command in the configuration file and a properly formatted SRAM PCMCIA card available, subscriber end stations using DHCP can be successfully reconnected after system restart.

Enable the DHCP Server

To enable the internal DHCP server to dynamically assign IP addresses, enter the following command in interface configuration mode:

dhcp server range {**all** | *ip-address ip-address*}

Configure a Secondary DHCP Server

To configure a secondary DHCP server for load sharing or backup function to the primary DHCP server, enter the following command in interface configuration mode:

dhcp server next-server *ip-address*

Configure DHCP Server Parameters

You can configure several DHCP server parameters as described in this section.

Configure a Device's Boot File via the DHCP Server

For a device connected to the SMS device, you can configure that device's boot file via the DHCP server by entering the following command in interface configuration mode:

dhcp server filename *filename*

Set the Maximum Lease Time

To determine the maximum length of time an IP address is leased by the DHCP server, use the following command in interface configuration mode:

dhcp server max-lease-time *seconds*

Set the Default Lease Time

To determine the length of time an IP address is leased by the internal DHCP server when an explicit lease time is not requested, enter the following command in interface configuration mode:

dhcp server default-lease-time *seconds*

Configure DHCP Server Options

To configure DCHP server options, enter the following command in interface configuration mode:

dhcp server option *option*

See the "DHCP Commands" chapter in the *Access Operating System (AOS) Command Reference* publication for a list of the standard UNIX options. Options are fully described in RFC 2132, *DHCP Options and BOOTP Vendor Extensions*.

Display DHCP Information

To display DHCP information, enter the following command in operator exec mode:

show dhcp {**interface** [*name*] | **preserve-state** [**pre-bind** [**all**] | **secured-arp** [**all**]]}

When used with the **interface** keyword, the display includes the number of addresses that have been assigned by DHCP to the interface and the DHCP relay server size for the one or more interfaces. When used with the **preserve-state** keyword, the display includes the status of DHCP preserve-state and if enabled, the counters for the SRAM contents. Additional detailed information is included if the **pre-bind** or **secured-arp** options are used. The **all** keyword expands the display to include the secured-ARP or prebind entries for all contexts.

Display DHCP Server Information

To display DHCP server information, enter either of the following commands in operator exec mode:

show dhcp server lease [**all** | **circuit** {[*slot/port* {*vpi vci* | [*hdlc-channel*] *dlci*} | **lac** *vcn* | **lns** *vcn* | **pppoe** [*cm-slot-*] *session-id*} | **interface** *if-name* | **ip** *ip-address* | **mac** *mac-address*]

show dhcp server sram

Enable DHCP Debugging Messages

To enable debugging for DHCP so that log messages are written every time an event occurs, enter the following command in administrator exec mode:

debug dhcp {**packet** | **preserve-state** | **all**}

Configuration Examples

The following example assigns an IP address to each of two interfaces, and configures them to relay DHCP packets. Each interface expects a total of `200` host addresses to be assigned within its respective subnet:

```
[local]RedBack(config-ctx)#interface downstream1
[local]RedBack(config-if)#ip address 10.1.1.254 255.255.255.0
[local]RedBack(config-if)#dhcp relay size 200
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#interface downstream2
[local]RedBack(config-if)#ip address 10.1.2.254 255.255.255.0
[local]RedBack(config-if)#dhcp relay size 200
```

The following example creates two subscriber records, sub1 and sub2. The circuit associated with sub1 is implicitly bound to the downstream1 interface. The circuit associated with sub2 can be implicitly bound to either the downstream1 or downstream2 interface and is expected to consume four IP host addresses. The interface chosen is whichever interface has remaining capacity from its pool of 200 expected dynamic addresses.

```
[local]RedBack(config-ctx)#subscriber name sub1
[local]RedBack(config-sub)#ip address 10.1.1.1
[local]RedBack(config-sub)#subscriber name sub2
[local]RedBack(config-sub)#dhcp max-addr 4
```

The following example formats pcmcia0 for nonvolatile storage of DHCP secured-ARP state information and enables DHCP preserve-state:

```
[local]RedBack#format /pcmcia0 dhcp-secured-arp
[local]RedBack#config
[local]RedBack(config)#dhcp preserve-state
```

The following example configures a DHCP server using an IP address range of 10.0.154.100 10.0.154.199. The system log server is configured at IP address 10.0.154.200.

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#interface i1
[local]RedBack(config-if)#ip address 10.0.154.1 255.255.255.0
[local]RedBack(config-if)#ip arp arpa
[local]RedBack(config-if)#dhcp server range 10.0.154.100 10.0.154.199
[local]RedBack(config-if)#dhcp server default-lease-time 3600
[local]RedBack(config-if)#dhcp server max-lease-time 3600
[local]RedBack(config-if)#dhcp server option log-server 10.0.154.200
.
.
.
[local]RedBack(config)#port ethernet 6/1
[local]RedBack(port)#bind interface i1 local
```

Configuring NTP

This chapter provides an overview of the Network Time Protocol (NTP) and describes the tasks involved in configuring NTP through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "NTP Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

The AOS supports NTP as described in RFC 1305, *Network Time Protocol*. Although the default version is Version 3, the AOS also supports versions 1 and 2. The NTP on a Subscriber Management System (SMS) device operates in client mode only. The SMS device can be synchronized by a remote NTP server, but the remote server cannot be synchronized by the SMS device.

The NTP exchanges timekeeping information between servers and clients via the Internet and corrects errors, such as equipment or propagation failures, in an effort to synchronize clocks. In a hierarchy of servers and clients, each level in the hierarchy is identified by a stratum number indicating the accuracy of each server. The highest level (primary server) is 1 with each level downward (secondary servers) assigned a number that is one greater than the preceding level.

Extremely reliable sources, such as radio clocks and Global Positioning System (GPS) satellite timing receivers, typically act as primary servers. Company or campus servers can act as secondary time servers. To reduce the protocol overhead, secondary servers distribute time to attached local hosts.

The NTP makes estimates based on the following variables shared between a client and a server: network delay, dispersion of time packet exchanges (a measure of maximum clock error between the two hosts), and clock offset (the correction needed to synchronize clocks). Clock synchronization occurs at approximately 10 ms across WANs and at 1 ms across LANs.

Configuration Tasks

To configure NTP, perform the tasks in the following sections:

- Configure the SMS Device to Synchronize to an NTP Server
- Set NTP Parameters
- Display NTP Information
- Enable NTP Debugging Messages

Configure the SMS Device to Synchronize to an NTP Server

Configure the SMS device to synchronize to a remote NTP server using the following global configuration command:

ntp server *ip-address* [**context** *ctx-name*] [**prefer**] [**source** *if-name*] [**version** *ver-num*]

If the NTP server must be reached through a context other than local, you must specify the context. If multiple NTP servers are configured, you can mark one server as the first-choice server by using the **prefer** keyword. Use the **source** keyword to choose the SMS device interface that is used for NTP traffic. Use the **version** *ver-num* construct to modify the version of NTP that is used.

Set NTP Parameters

Optionally, you can enter NTP configuration mode and then set NTP parameters.

To enter NTP configuration mode, enter the following command in global configuration mode:

ntp mode

Then in NTP configuration mode, enter the following command:

slowsync

By default, the SMS device NTP daemon adjusts the local clock within a few minutes if the difference between the local clock and the remote NTP server is greater than five seconds. However, you can instead use the **slowsync** command, which changes the rate of the SMS device clock, so that it gradually converges with the NTP server clock—provided the initial difference in time between the two clocks is less than 16 minutes. If the time difference is more than 16 minutes, synchronization does not occur.

Display NTP Information

To list current associations with remote NTP servers and display NTP daemon statistics, enter the following operator exec command:

show ntp associations

To list current internal NTP parameter settings and synchronization status, enter the following operator exec command:

show ntp status

Enable NTP Debugging Messages

To enable NTP debugging messages, enter the following administrator exec command:

debug ntp

Configuration Examples

In the following example, the NTP client on the SMS device is configured to synchronize with a remote NTP server at IP address 10.1.1.1:

```
[local]RedBack(config)#ntp server 10.1.1.1
```

The following commands configure the NTP client on the SMS device to use multiple remote NTP servers as synchronization sources. In this case, the preferred server is at IP address `20.1.1.1`:

```
[local]RedBack(config)#ntp server 10.1.1.1
[local]RedBack(config)#ntp server 20.1.1.1 prefer
```

Part 9

Routing

Configuring Basic IP Routing

This chapter provides an overview of IP routing and describes the tasks involved in configuring static IP routing through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "Basic IP Routing Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

IP routing moves information across an internetwork from a source to a destination, typically passing through one or more intermediate nodes along the way. The primary difference between routing and bridging is that they access different levels of information to determine how to transport packets from source to destination—routing occurs at layer 3 (the network layer), while bridging occurs at layer 2 (the link layer) of the Open Systems Interconnection (OSI) reference model.

In addition to transporting packets through an internetwork, routing involves determining optimal paths to a destination. Routing algorithms use metrics, or standards of measurement, to establish these optimal paths, initializing and maintaining routing tables that contain all route information.

The AOS routing table stores routes to directly attached devices, static IP routes, routes configured in subscriber records, and routes learned dynamically from the Routing Information Protocol (RIP), the Open Shortest Path First (OSPF) protocol, and the Border Gateway Protocol (BGP).

In the routing table, next-hop associations specify that a destination can be reached by sending packets to a next-hop router located on an optimal path to the destination. When the Subscriber Management System (SMS) device receives an incoming packet, it checks the destination address and attempts to associate this address with a next-hop address and outgoing interface.

Routing algorithms must converge rapidly; that is, all routers must agree on optimal routes. When a network event causes routes either to go down or become unavailable, routers distribute routing update messages that are propagated across networks, causing a universally agreed recalculation of optimal routes. Routing algorithms that converge slowly can cause routing loops or network outages. Many algorithms can quickly select next-best paths and adapt to changes in network topology.

Static Versus Dynamic Routing

Static routing involves packet forwarding on the basis of static routes configured by the system administrator. Static routes work well in environments where network traffic is relatively predictable and network topology is relatively simple.

In contrast, dynamic routing algorithms adjust to changing network circumstances by analyzing incoming routing update messages. RIP, BGP, and OSPF all use dynamic routing algorithms. A dynamic routing algorithm can also be supplemented with static routes where appropriate. For example, a router of last resort (to which all unroutable packets are sent) can store information on such packets for troubleshooting purposes.

Some routing algorithms operate in a flat, hierarchy-free space, while others use routing hierarchies. In a flat routing system, such as RIP, all routers are peers of all other routers. As networks increase in size, flat routing systems encounter scaling limitations. To address this, some routing protocols allow the administrator to partition the network into hierarchical levels. Partitioning facilitates the summary of topology information for anyone located outside the immediate level or area. An example is the OSPF protocol, which supports a two-level hierarchy where area 0 is the backbone area that interconnects all other areas.

IGPs Versus EGPs

Another group of protocols that works to optimize network performance are the Interior Gateway Protocols (IGPs). These optimize the route between points within a network. Examples of commonly used IGPs are RIP, OSPF, and Intermediate System to Intermediate System (IS-IS).

Exterior Gateway Protocols (EGPs) support route information exchange between different networks. An example of a commonly-used EGP is BGP-4. The choice of an optimal path is made based on the cost of the path measured by metrics associated with each link in the network.

IGPs and EGPs have slightly differing administrative designs. IGP typically runs in an area under a single administrative control; this area is referred to as an autonomous system (AS) or a routing domain. In contrast, an EGP allows two different autonomous systems to exchange routing information and send data across the AS border. Policy decisions in EGPs can be shaped to decide which routing information crosses the border between the two autonomous systems.

Equal-Cost Multipath Forwarding

Equal-cost multipath forwarding utilizes the full bandwidth of multiple routes to a destination. Without equal-cost multipath forwarding, only one of many equal-cost paths between a source and its destination can be utilized.

Equal-cost multipath forwarding uses a hash threshold algorithm to spread session traffic equally among as many as six equal-cost paths to a destination. The Subscriber Management System (SMS) device chooses among equal-cost paths by examining the IP and protocol headers.

Note Though the system default configures six paths, the administrator can set any number up to six using the AOS **ip maximum-paths** command in context configuration mode.

Depending on the specific transport protocol used, the equal-cost hash algorithm functions differently:

- When the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are used, the algorithm reads the source and destination address and the source and destination ports.
- If the generic routing encapsulation (GRE) protocol is active, the hash algorithm makes use of GRE keys to identify tunnels (logical connections between private networks over the Internet) and specific virtual private networks (VPNs) associated with those tunnels.
- For Internet Control Message Protocol (ICMP) traffic and all other forms of traffic, the algorithm uses the source and destination addresses.

Equal-cost multipath forwarding is supported by OSPF routes and by static IP routes. As many as six equal-cost routes can be installed using these two protocols. Once the equal-cost hash algorithm selects a path, packets between a given source and destination are forwarded along that path. The path is altered only if network topology changes. This mechanism reduces the chance of out-of-order packet delivery for a specific flow.

ICMP Router Discovery Protocol

The Internet Control Message Protocol (ICMP) Router Discovery Protocol (IRDP) enables hosts to learn their default route via router-transmitted advertisement packets instead of through manual configuration of the hosts. IRDP sends advertisements on a regular basis and in response to host solicitations, which are typically generated when hosts boot up.

IP Routing Protocols

Redback currently supports the following IP routing protocols:

- RIP is a distance-vector IGP that uses hop count as its metric. Each router sends all or some of the portion of its routing table, but only to its neighbors. The RIP is widely used for routing traffic in the global Internet; see Chapter 32, “Configuring RIP.”
- OSPF is a link-state IGP that uses link-state advertisements (LSAs) to inform other routers of the state of the sender’s links. Each router sends only the portion of the routing table that describes the state of its own links to all nodes in the internetwork. LSAs are used to build a complete picture of the network topology, enabling other routers to determine optimal routes to destinations.

  In OSPF, the autonomous system can be hierarchically organized by partitioning it into areas. Each area contains a group of contiguous networks and hosts. An area border router (ABR) communicates routing information between the areas; see Chapter 33, “Configuring OSPF.”
- BGP-4 is a distance-vector EGP, and uses the Transmission Control Protocol (TCP) as its transport protocol. With BGP, a TCP connection is established over which two BGP peers exchange routing information. Routers that belong to the same autonomous system run internal BGP (I-BGP), while routers that belong to different autonomous systems run external BGP (E-BGP); see Chapter 34, “Configuring BGP.”

Route Selection Process

When determining a single optimal route among multiple routes within a single routing protocol, the AOS selects the route that has the lowest cost. When deciding a best path among routes originating from multiple protocols, the system uses a more complex methodology. The AOS routing table stores direct, RIP, static, subscriber, OSPF, and BGP routes.

The SMS device selects the best route to a destination by choosing:

1. The lowest cost route within a protocol, such as OSPF.
2. The best protocol.

When each protocol submits a best path candidate to the route table manager, the route table manager determines the choice of active route to be installed in the forwarding table as follows:

1. The route with the lowest precedence value is selected first.

   Each route is associated with a default precedence value. This precedence value is assigned based on the source of the route information; see Table 31-1.
2. For BGP, the route with the lowest preference value is selected.
3. For BGP, the route configured with the highest local preference value is selected.
4. If a route includes an autonomous system path:
   a. The route with the fewest AS numbers listed in its path is preferred.
   b. The route with the lowest origin code is preferred. Routes learned from an IGP have a lower origin code than those learned from an EGP. Both routes learned from an IGP or an EGP have lower origin codes than routes with the origin code unknown.
   c. If routes are received from the same AS, the route with the lowest Multi-Exit Discriminator (MED) value is selected. A missing metric is treated as a MED value of 0.

5. Internal (IGP) routes are selected.
6. External (EGP) routes are selected.
7. For BGP, the route with a next hop that is resolved through the IGP route with the lowest metric is selected.
8. The route with the lowest router ID (ORIGINATOR_ID) is selected.
9. The route with the lowest next-hop IP address is selected.

Table 31-1 lists the default values for routes learned through various protocols.

Table 31-1 Protocol Precedence Defaults

Protocol	Precedence Value
Directly connected	0
Static IP	10
Subscriber record	15
OSPF—Internal to the autonomous system	60
RIP	100
OSPF—External to the autonomous system	150
BGP	170

Configuration Tasks

To configure and monitor basic IP routing, perform the tasks described in the following sections:

- Enable Equal-Cost Multipath Forwarding
- Configure Static IP Routes
- Enable IRDP on Interfaces
- Display Static IP Routes
- Display IP Route Information
- Enable IP Route Debugging Messages
- Enable IRDP Debugging Messages

Enable Equal-Cost Multipath Forwarding

To enable equal-cost multipath forwarding, enter the following command in context configuration mode:

ip maximum-paths *maximum*

You can configure up to six equal-cost paths. By default, the AOS sets the equal-cost multipath forwarding to 1 (disabled). This command is independent of any routing protocol and applies only to outbound traffic forwarding.

Configure Static IP Routes

Rather than dynamically selecting the best route to a destination, you can configure one or more static routes to the destination. Once configured, a static route stays in the routing table indefinitely. When multiple static routes are configured for a single destination and the outbound interface of the current static route goes down, a backup route is activated, improving network reliability.

Up to six routes can be statically configured for a single destination. Each route is assigned a default precedence value and cost value. Modifying these values allows you to set a preference for one route over the next.

Note If static routes are redistributed through dynamic routing protocol, only the active static route to a destination is advertised.

To configure a default static IP route, use 0.0.0.0 for the network number and mask. A valid next-hop IP address and interface is required. To attach static routes to a subscriber, enter the **ip route** command in subscriber configuration mode.

To configure one or more static IP routes to the same destination, enter the following command in context configuration mode:

ip route {*ip-address netmask ip-address if-name*} [**precedence** *value*] [**cost** *value*]

The *netmask* argument specifies the IP address and netmask of the target network. The *ip-address* argument specifies the IP address of the next-hop router on the path to the target. The *if-name* argument specifies the name of the outgoing interface.

Optionally, you can change the default precedence value for the route, assign a cost to the route, or both. When equal-cost multipath forwarding is enabled, the system selects a subset of routes to install in the forwarding table. The maximum number of routes installed depends on the current maximum path configuration. For example, if you set the number of maximum paths to two using the **ip maximum-paths** command and configure three static routes of equal cost and precedence to the same destination, only two of these routes are submitted to the route table manager. The third is held in reserve as a floating route.

Among multiple routes with the same destination, preferred routes are selected in the following order:

1. The route with the lowest precedence value is preferred first.
2. If there are two or more routes with the same precedence value, the route with the lowest cost value is preferred.
3. If there are two or more routes with the same precedence and cost values, the route with the lowest IP address is preferred.
4. When redistributing static routes, routing protocols ignore the cost value assigned to those static routes. If static routes are redistributed through dynamic routing protocols, only the active static route to a destination is advertised.

Enable IRDP on Interfaces

To enable IRDP on an interface, enter the following command in interface configuration mode:

ip irdp [**broadcast**] [**maximum** *value*] [**minimum** *value*] [**lifetime** *value*] [**preference** *value*]

Use the **show ip interface** command in operator exec mode to display IRDP information; see Chapter 7, "Configuring Interfaces."

Display Static IP Routes

To display static IP routes, enter the following command in operator exec mode:

show ip static-route [*ip-address* [*netmask*]]

This command displays the destination address, next-hop interface, the interface cost, and its time-to-live value. If there are multiple routes to the same destination, the preferred route is indicated by an asterisk (*). If multiple equal cost paths to a destination are available, all are displayed.

Display IP Route Information

To display IP route information, enter the following command in operator exec mode:

show ip route [*ip-address* [*netmask*] | **detail**]

You can display the entire routing table used for IP data forwarding in the current context, display information about a specific network destination, and view detailed information on protocol-specific metrics attached to routes. All multiple equal-cost routes to a destination are shown.

Enable IP Route Debugging Messages

To enable the logging of IP routing debug messages, enter the following command in administrator exec mode:

debug ip route

Enable IRDP Debugging Messages

To enable the logging of IRDP debugging messages, enter the following command in administrator exec mode:

debug ip irdp [**circuit** {*slot/port* {*vpi vci* | *hdlc-channel dlci*} | **lac** *vcn* | **lns** *vcn* | **pppoe** *cm-slot-session-id*}]

Configuration Examples

The following example routes packets for network 10.10.0.0/16 via interface enet1 to the device at 10.3.2.1:

```
[local]RedBack(config-ctx)#ip route 10.10.0.0 255.255.0.0 10.3.2.1 enet1
```

The following example has an IP default route with a cost of 2, and uses atm5 as the outgoing interface and 10.1.1.1 as the next-hop router:

```
[local]RedBack(config-ctx)#ip route 0.0.0.0 0.0.0.0 10.1.1.1 atm5 2
```

In the following example, the first static route, atm5 has a default cost of 0 and, therefore, is used as the active route. Both e6 and atm6 have the same cost (2). In the event that atm5 goes down, atm6 becomes the interface with the preferred route, because its IP address is lower than that of e6.

```
[local]RedBack(config-ctx)#ip route 0.0.0.0 0.0.0.0 10.1.1.1 atm5
[local]RedBack(config-ctx)#ip route 0.0.0.0 0.0.0.0 172.21.200.254 e6 2
[local]RedBack(config-ctx)#ip route 0.0.0.0 0.0.0.0 10.1.1.1 atm6 2
```

The following example configures two equal-cost routes to the same destination 1.0.0.0:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#ip maximum-paths 2
[local]RedBack(config-ctx)#ip route 1.0.0.0 255.0.0.0 3.3.3.3 nhop3 cost 5
[local]RedBack(config-ctx)#ip route 1.0.0.0 255.0.0.0 4.4.4.4 nhop4 cost 5
```

The following example displays the two routes configured in the previous example:

```
[local]RedBack#show ip route

Destination       Nexthop     Protocol    Precedence   Cost   Ttl
1.0.0.0/8         3.3.3.3     static      10           5      infinity
                  4.4.4.4     static      10           5      infinity
```

The following example configures an additional route to the same destination 1.0.0.0:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#ip route 1.0.0.0 255.0.0.0 2.2.2.2 nhop2 cost 5
```

The following example displays the routing table, which has been updated with the addition of the route configured in the previous example. Because the next-hop IP addresses of 2.2.2.2 and 3.3.3.3 are lower than the 4.4.4.4 IP address, and because the maximum paths to the destination is set to 2, the 4.4.4.4 next-hop IP address is removed from the routing table.

```
[local]RedBack#show ip route

Destination      Nexthop       Protocol     Precedence   Cost  Ttl
1.0.0.0/8        2.2.2.2       static       10           5     infinity
                 3.3.3.3       static       10           5     infinity
```

The following example configures a new route that supersedes all previously configured routes because it has a lower cost (0 versus 5):

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#ip route 1.0.0.0 255.0.0.0 5.5.5.5 nhop5
```

The following example displays the updated routing table. Previously configured routes are held in floating status in case this route becomes unavailable.

```
[local]RedBack#show ip route

Destination      Nexthop       Protocol     Precedence   Cost  Ttl
1.0.0.0/8        5.5.5.5       static       10           0     infinity
```

Configuring RIP

This chapter provides an overview of the Routing Information Protocol (RIP) and describes the tasks involved in configuring RIP features through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed under the "Configuration Tasks" section, see the "RIP Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

Redback supports RIP versions 1 and 2. RIP is a distance-vector protocol that uses a hop count as its metric. RIP is widely used for routing traffic in the global Internet and is an Interior Gateway Protocol (IGP); it performs routing within a single autonomous system.

Configuration Tasks

To configure RIP, perform the tasks described in the following sections:

- Enable RIP
- Configure RIP Interfaces
- Configure the Precedence for RIP-Learned Routes
- Redistribute Routes Learned via Other Protocols into RIP
- Disable Automatic Network Number Summarization
- Modify the RIP Version

- Display IP Routes
- Enable the Logging of RIP Debug Messages

Enable RIP

1. To enable the RIP routing process, enter the following command in context configuration mode:

   router rip

2. Then specify network or networks for which directly connected interfaces automatically receive and send RIP updates by entering the following command in RIP configuration mode:

   network *network*

Autosummarization is enabled by default. To disable autosummarization, see the "Disable Automatic Network Number Summarization" section.

By default, the Subscriber Management System (SMS) device receives RIP version 1 and 2 packets, but sends only version 1 packets. To modify the RIP version, see the

Configure RIP Interfaces

You can enable interfaces to send or receive RIP packets, modify the version of RIP being sent or received, and enable split-horizon processing.

Enable an Interface to Send or Receive RIP Packets

To enable an interface to send RIP packets, enter the following command in interface configuration mode:

ip rip supply

To enable an interface to receive RIP packets, enter the following command in interface configuration mode:

ip rip listen

See the "Enable RIP" section to understand how the **network** command in RIP configuration mode can enable RIP packets to be sent and received on a set of directly connected interfaces.

Modify the RIP Version an Interface Sends or Receives

To modify the RIP version that an interface sends, enter the following command in interface configuration mode:

ip rip send version {**1** | **2**}

To modify the RIP version that an interface receives, enter the following command in interface configuration mode:

ip rip receive version {**1** | **2**}

See the "Enable RIP" section to understand how the **version** command in context configuration mode applies the RIP version used by all interfaces within a context.

Disable Split-Horizon Processing

Split-horizon processing prevents routing loops in distance-vector routing protocols, such as RIP. It blocks route information from being advertised out any interface from which the information originated. Split-horizon processing is enabled by default.

To disable split-horizon processing, enter the following command in interface configuration mode:

no ip rip split-horizon

Configure the Cost Value

To modify the cost value of an interface, enter the following command in interface configuration mode:

ip rip interface-cost *cost*

By default, the cost is 0. The cost value is used by RIP as a metric for route selection. The lower the cost, the more likely an interface is to be used to forward data traffic.

Configure the Precedence for RIP-Learned Routes

The SMS device assigns a default precedence value to each route that the routing table receives. A lower value indicates a more-preferred route. The default value depends on the source of the route. For RIP, the default value is 100.

To modify the precedence value for RIP, enter the following command in RIP configuration mode:

precedence *precedence*

Redistribute Routes Learned via Other Protocols into RIP

To redistribute routes learned through protocols other than RIP into the RIP routing process, enter the following command in RIP configuration mode:

redistribute {**bgp** | **direct** | **ospf** | **static** | **subscriber**} [**metric** *metric*]

Routes learned through the Border Gateway Protocol (BGP), directly attached networks, the Open Shortest Path First (OSPF) protocol, static routes, and routes learned through subscriber records can all be redistributed into a RIP network. Use the optional **metric** keyword to set a metric value for the redistributed route.

Disable Automatic Network Number Summarization

RIP automatic network number summarization (autosummarization) is on by default. With autosummarization, the SMS device summarizes subprefixes to the Class A, Class B, and Class C network boundaries when class network boundaries are crossed.

To disable RIP autosummarization, enter the following command in RIP configuration mode:

{**no** | **default**} **auto-summary**

Modify the RIP Version

Modify the RIP version for the routing process by entering the following command in RIP configuration mode:

version {**1** | **2**}

Display IP Routes

To display IP route information, enter the following command in operator exec mode:

show ip route [*ip-address* [*netmask*] | **detail**]

You can display the entire routing table used for IP data forwarding in the current context, display information about a specific network destination, and view detailed information on protocol-specific metrics attached to routes.

See the "Basic IP Routing Commands" chapter in the *Access Operating System (AOS) Command Reference* for detailed information on this command.

Enable the Logging of RIP Debug Messages

To enable the logging of RIP debug messages, enter the following command in administrator exec mode:

debug ip rip

Configuration Examples

The following example creates a RIP process in the local context:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#router rip
[local]RedBack(config-router)#network 10.0.0.0
```

The process is further configured to listen to and supply RIP updates for any interfaces within the context that have a network prefix of 10.0.0.0/8.

Configuring OSPF

This chapter provides an overview of the Open Shortest Path First (OSPF) protocol and describes the tasks involved in configuring OSPF features through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "OSPF Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

OSPF is an Interior Gateway Protocol (IGP) that uses link-state advertisements (LSAs) to inform other routers of the state of the sender's links. In a link-state routing protocol, each router distributes information about its interfaces and neighbor relationships. The collection of the link states of individual routers forms a database that describes the autonomous system (AS) topology. As OSPF routers accumulate link-state information, they use the Shortest Path First (SPF) algorithm to calculate the shortest path to each node, which forms the basis for developing routing information for that autonomous system.

Redback's implementation of RFC 2328, *Open Shortest Path First (OSPF) Version 2* and Internet Draft *The OSPF NSSA Option*, draft-ietf-ospf-nssa-update-09.txt, supports:

- One instance of OSPF routing per context
- Configuration of the Subscriber Management System (SMS) device as an internal router, an autonomous system boundary router (ASBR), an area border router (ABR), or a designated router
- Running OSPF on interfaces connected to a point-to-point network (a single pair of routers) or a broadcast network (several routers addressing one message to all attached routers)
- Multiples areas per context and interarea route summarization.
- Stub areas, not-so-stubby-areas (NSSAs), and NSSA route summarization
- Default route injection into stub areas and NSSAs

- Simple and MD5 authentication
- Redistribution of routes learned through other protocols into OSPF and interautonomous system route summarization

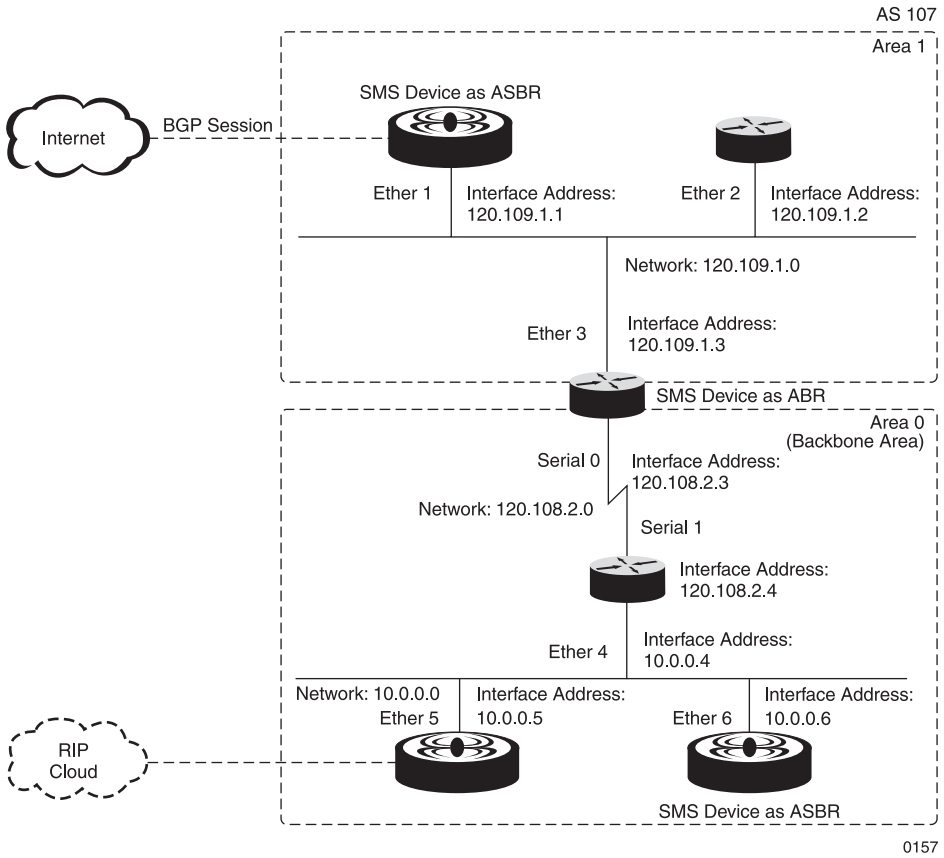
OSPF Hierarchy

In OSPF, the AS can be hierarchically organized by partitioning it into areas. Externally derived routes, also called AS-external routes, are routes learned from other routing protocols that are redistributed into the OSPF domain (or AS). These AS-external routes are advertised to all areas, except for stub areas and NSSAs. AS-external routes can also be forwarded out to another AS through routers on its boundary.

Areas

Each area can contain a group of contiguous networks and hosts. An ABR communicates routing information between the areas; see Figure 33-1.

Figure 33-1 OSPF Hierarchy



Because routers within the same area share the same information, they have identical topological databases. An area's topology is invisible to entities outside the area. By keeping area topologies separate, OSPF passes less routing traffic than it would if an autonomous system were not partitioned.

Area partitioning creates two different types of OSPF routing, depending on whether the source and destination are in the same or different areas. Intra-area routing occurs when the source and destination are in the same area; interarea routing occurs when they are in different areas.

Normal and Backbone

A normal OSPF area, including the backbone area, is distinguished by the fact that it can carry transit traffic, allowing LSAs from outside the autonomous system (type 5 AS-external-LSAs) to be flooded throughout the area. Type 5 AS-external-LSAs can be originated both by routers internal to the area or by ABRs.

Hierarchical organization of an OSPF autonomous system requires one of the areas to be configured as the backbone area. The backbone area is configured with an identity of 0 and must be contiguous, contain all area border routers, and be responsible for distributing routing information to all other nonbackbone areas.

Stub

OSPF also allows some areas to be configured as stub areas. Type 5 AS-external LSAs are *not* flooded into a stub area, thereby reducing the link state database size and the processor and memory usage of routers inside stub areas. While a stub area cannot propagate routes external to the autonomous system in which it resides, it can propagate a default route, intra-area routes, and interarea routes. A stub area relies on default routing to forward traffic addressed to external destinations. The backbone area *cannot* be configured as a stub area.

NSSA

Not-so-stubby-areas (NSSAs) are an extension of OSPF stub areas. Their intent is to preserve the properties of a stub area, while allowing limited import of external routes from other routing domains. These routes are imported as Type 7 NSSA-external LSAs, which are flooded only within the NSSA. For propagation of these routes to other areas, type 7 LSAs must be translated into type 5 external LSAs by the NSSA ABR. NSSA ABRs can also advertise a type 7 default route into the NSSA, and can be configured to summarize and to filter the translation of type 7 NSSA-external LSAs into Type 5 external LSAs.

Router Functions

Depending on its location in the OSPF hierarchy, an OSPF router can provide one or more of the following functions:

- Internal router

  A router with all directly connected networks belonging to the same area. An internal router maintains a single topological database.

- Backbone router

  A router that has one or more interfaces to the backbone area. An OSPF backbone is responsible for distributing routing information between areas.

- ABR

  A router that attaches to multiple areas. ABRs maintain a separate topological database for each attached area and summarize the information for distribution to the backbone. The backbone in turn distributes the information to the other areas.

- ASBR

  An ASBR exchanges routing information with routers belonging to other autonomous systems, and advertises external routing information throughout its internal autonomous system. The paths to each AS boundary router are known by every router in the autonomous system.

  AS boundary routers can be internal or area border routers, and may or may not participate in the backbone. ASBRs cannot be part of a stub area unless they are also ABRs; that is, connected to other non-stub areas.

- Designated router and backup designated router

  On networks with more than one router, a designated router is responsible for generating the LSAs for the network. The designated router is elected by the Hello protocol. Designated routers allow a reduction in network traffic and in the size of the topological database. Backup designated routers provide a failsafe in case the designated router is not operational.

Route Selection Process

A routing table contains all the information necessary to forward an IP packet to a destination. When forwarding an IP data packet, the routing table entry providing the best match for the packet's IP destination is located. In the case of OSPF, the best path to a destination is determined via the SPF computation performed on the link-state database.

From the link-state database, the router uses the Dijkstra algorithm to construct a tree of shortest paths with itself as root. This shortest-path tree gives the route to each destination in the autonomous system. A separate SPF computation is performed and a different tree is constructed for each area in which the router belongs. Externally derived routing information appears on the tree as leaves. Intra-area paths are preferred over inter-area paths.

Packet Types

OSPF runs directly on top of IP (protocol 89). There are five types of packets specified in OSPF:

- *Hello*

  The SMS or router sends Hello packets to its neighbors and receives their Hello packets. In this manner, adjacencies between neighbors are established. (Not all neighboring routers are adjacent.)

- *Database description*

  Sent by adjacent routers when an adjacency is initialized, database description packets describe the contents of the respective database to synchronize the two neighboring databases.

- *Link-state request*

  Requests pieces of the topological database from neighbor routers. These messages are sent after a router discovers (by examining database-description packets) that parts of its topological database are out of date.

- *Link-state update*

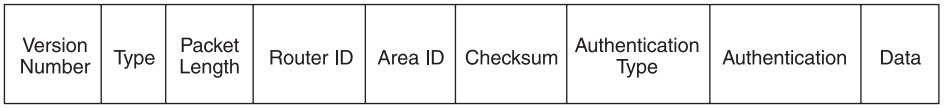
  Responds to a link-state request packet. These messages are also used for the regular flooding of LSAs. Several LSAs can be included within a single link-state update packet.

- *Link-state acknowledgment*

  Acknowledges link-state update packets.

Each packet includes a common header as illustrated in Figure 33-2.

Figure 33-2 OSPF Packet Header



The OSPF packet header contains the following fields:

- Version Number

  Identifies the OSPF version.

- Type

  Identifies the OSPF packet type; for example, Hello, database description, link-state request, link-state update, and link-state acknowledgement.

- Packet Length

  Specifies the packet length, including the OSPF header, in bytes.

- Router ID

  Identifies the source of the packet.

- Area ID

  Identifies the area to which the packet belongs. A packet is associated with a single area.

- Checksum

  Checks the entire packet contents for any damage that may have occurred in transit.

- Authentication Type

  Contains the authentication type. All OSPF protocol exchanges are authenticated. The authentication type is configurable on a per-area basis.

- Authentication

  Contains authentication information.

- Data

  Contains packet data.

LSAs

Table 33-1 describes the LSAs types:

Table 33-1 LSA Types and Descriptions

ID	Type	Description
1	Router-LSA	Originated by all routers. Describes the collected states of the router's interfaces to an area. Flooded throughout a single area only.
2	Network-LSA	Originated by the designated router. Contains the list of routers connected to the network. Flooded throughout a single area only.
3	Summary-LSA (networks)	Originated by ABRs. Describes routes to networks. Each summary-LSA describes a route to a destination outside the area, but still inside the autonomous system.
4	Summary-LSA (routers)	Originated by ABRs. Describes routes to ASBRs. Each summary-LSA describes a route to a destination outside the area, but still inside the autonomous system.
5	AS-external-LSAs	Originated by ASBRs and flooded throughout the autonomous system. Each AS-external-LSA describes a route to a destination in another autonomous system. Default routes for the AS can also be described by AS-external-LSAs.
7	NSSA-external-LSAs	Originated by ASBRs that connect the NSSA to the network outside the OSPF routing domain. Type 7 LSAs are advertised only within an NSSA. When forwarded outside the NSSA to nonstub areas, type 7 LSAs are converted into type 5 LSAs by an ABR configured to perform translation, or by the ABR with the highest router ID. ABRs can be configured to summarize and filter type 7 LSAs.

Configuration Tasks

To configure OSPF routing for a context, perform the tasks in the following sections:

- Enable OSPF Routing
- Configure Global Parameters
- Configure Area Parameters
- Configure OSPF Interface Characteristics
- Display OSPF Information
- Enable OSPF Debugging Messages
- Configuration Examples

Enable OSPF Routing

To enable OSPF routing:

1. Set the router ID using the following command in context configuration mode:

   router-id *ip-address*

   This command establishes the IP address that is used to uniquely identify the OSPF router in the autonomous system.

2. Enable OSPF routing and enter OSPF configuration mode using the following command in context configuration mode:

   router ospf

   You can configure only one OSPF routing process per context.

3. Configure one area and enter OSPF area configuration mode using the following command in OSPF configuration mode:

   area {*id* | *ip-address*}

   The area command configures a normal area. To configure the area as a stub type, see the "Configure Area Parameters" section.

4. Enable OSPF on at least one interface and enter OSPF interface configuration mode using the following command in OSPF area configuration mode:

   ospf-interface *ip-address* {**broadcast** | **p2p** | **loopback**}

   OSPF interfaces can be attached to broadcast or point-to-point (P2P) networks, or to a loopback interface. The IP address of the OSPF interface must match the address of an IP interface that has already been configured using the **interface** command in context configuration mode.

Configure Global Parameters

Optionally, you can define parameters that affect the operation of OSPF within the context.

Note All of the commands described in this section are entered in OSPF configuration mode.

Set the Precedence for OSPF-Learned Routes

The SMS device assigns a default value to each routing protocol process that the routing table receives.

To modify the precedence for routes learned through OSPF, enter the following command:

precedence *internal external*

Configure a Route Address Range for Inter-AS Route Summarization

To summarize AS external routes for redistribution into an OSPF domain, enter the following command:

as-sumrange *ip-address netmask* [**not-advertise**]

This command is only valid when the SMS device is configured as an ASBR. Summarization of routes reduces the size of the OSPF routing table. To suppress the specified route from being advertised in route summarizations, use the optional **not-advertise** keyword.

Enable an ASBR to Originate A Default Route

To enable an ASBR to originate a default route into an OSPF domain, enter the following command:

default-originate [**always**] [**metric** *metric*]

Use the optional **always** keyword to enable the system to originate the default route regardless of whether or not there is an active default route in the routing table. When this keyword is not specified, a default route is only originated if there is an active default route in the routing table. Use the optional **metric** keyword to assign a metric value to the default route.

Redistribute Routes Learned via Other Protocols into OSPF

You can redistribute routes from BGP networks, directly connected networks, RIP networks, networks running static IP, and from routes that are applied to subscriber records into OSPF. You can alter the metric value of each type of route redistribution. The metric is always specified as a Type 2 external route metric.

To redistribute routes learned via other protocols into OSPF, enter the following command:

redistribute {**bgp** | **direct** | **rip** | **static** | **subscriber** [**metric** *metric*]}

This command does not cause the SMS device to redistribute a default route into the OSPF domain unless it is used in conjunction with the **default-originate** command in OSPF configuration mode for external-capable areas, or with the **defaultroute** command in OSPF configuration mode for NSSAs.

Modify SPF Calculation Times

To modify SPF calculation timers, enter the following command:

spf-timers *delay holdtime*

Configure Area Parameters

You can configure an area as a stub type or as an NSSA, control the summarization of routes sent out an NSSA, configure route address ranges for interarea route summarization, and set the metric for default routes injected into a stub area or NSSA.

Note All commands in this section are entered in OSPF area configuration mode.

Configure an Area as a Stub or NSSA

To designate an area as a stub area or a not-so-stubby-area (NSSA) type, enter the following command:

areatype {**nssa** [**always-translate** | **noredistribute** | **nosummary**] | **stub** [**nosummary**]}

Control the Summarization of Routes Sent out an NSSA

To control the summarization of routes that are translated into type 5 AS-external LSAs when sent out from a NSSA by an ABR, enter the following command:

nssa-sumrange *prefix netmask* [**not-advertise**]

To suppress the translation into type 5 AS-external LSAs, use the **not-advertise** keyword.

Configure a Route Address Range for Interarea Route Summarization

To specify the address range of a route for the purpose of interarea route summarization, enter the following command:

area-sumrange *ip-address netmask* [**not-advertise**]

This command is only relevant when the SMS device is configured as an ABR. To suppress the specified route from being advertised in route summarizations, use the **not-advertise** keyword.

Configure a Default Route Metric for a Stub Area or NSSA

To specify the metric for a default route to be injected into a stub area or NSSA, enter the following command:

defaultroute [**metric** *metric*]

This command is only applicable when the SMS device is configured as an ABR. The metric value allows the SMS device to determine the distance between itself and the sending router.

Configure OSPF Interface Characteristics

To change OSPF interface default settings, perform any or all tasks in the following sections.

Note All commands in this sections are entered in OSPF interface configuration mode.

Modify the Router Priority

Modify the preference of the SMS device to act as the designated router for a network:

router-priority *priority*

By default, the value is 1. The value must be greater than or equal to 1 to indicate that the SMS device can act as a designated router. The router with the highest router priority is used as the designated router for the network, if there is not a previously-elected designated router already on the network.

Set an Authentication Password

To set an authentication password, enter the following command:

authentication {**simple** *password* | **md5 keyid** *keyid password*}

Modify the Routing Cost

The OSPF metric is advertised as the cost of an interface or link. The cost for a route is the sum of the costs for all the links in the route. The lower the cost, the more likely an interface is to be used to forward data traffic. Assign only one cost per interface.

To configure the cost of the interface, enter the following command:

cost *cost*

Modify the Interval Between Hello Packets

Devices send Hello packets at a fixed interval on all interfaces to establish and maintain neighbor relationships. The smaller the Hello interval, the faster topological changes are detected, but more routing traffic ensues.

To modify the interval between Hello packets, enter the following command:

hello-interval *interval*

Modify the Interval Between LSA Retransmissions

When a router sends link-state advertisements to its neighbors, the router expects to receive an acknowledgment packet within a certain amount of time. If the router does not receive an acknowledgment, it retransmits the advertisement.

To modify the interval between LSA retransmissions, enter the following command:

retransmit-interval *interval*

Modify the Router Dead Interval

To modify the interval the SMS device waits without receiving a Hello packet before declaring its neighbor is nonoperational, enter the following command:

routerdead-interval *interval*

Modify the Transmit Delay Value

Before a link-state update packet is propagated out of an interface, the router must increase the age of the packet by the specified transmit delay.

To modify the transmit delay value, enter the following command:

transmit-delay *delay*

Display OSPF Information

To obtain information about OSPF, use any or all of the **show ip ospf** commands described here. All of the commands described in this section are entered in operator exec mode.

Show Global Information

To display OSPF top-level session information, enter the following command:

show ip ospf

Show OSPF Areas

To display OSPF area information, enter the following command:

show ip ospf area [[*id* | *ip-address*] | [**detail**]]

Show OSPF ABRs and ASBRs

To display information about OSPF ABRs and ASBRs, enter the following command:

show ip ospf border-router

Show OSPF Database Information

To display OSPF database information, enter the following command:

show ip ospf database [*id* | *ip-address*] [**database-summary** | {**external** | **network** | **nssa-ext** | **router** | **sum-asbr** | **sum-net**} *linkid linkadvrt*]

Show OSPF Interface Information

To display OSPF interface information, enter the following command:

show ip ospf interface [*ip-address* | **detail**]

Show OSPF Neighbor Information

To display OSPF neighbor information, enter the following command:

show ip ospf neighbor [*id* | **detail**]

Show OSPF Route Summarization

To display OSPF route summarization information, enter the following command:

show ip ospf summary-range [**area** [*area-id*] | **as** | **nssa** [*area-id*]] [*ip-address netmask*]]

Enable OSPF Debugging Messages

You can view information on OSPF state transitions, SPF computations, designated router election, link-state database computations, OSPF routes summarized by ABRs, OSPF policy changes and information on OSPF packets.

To debug OSPF problems, enter the following command in administrator exec mode:

debug ip ospf {**database** | **packet** {**ack** | **all** | **dd** | **hello** | **lsr** | **lsu**}| **policy** | **spf** | **state**}

Configuration Examples

The following example provides a basic OSPF configuration:

```
[local]RedBack#config
[local]RedBack(config)#context a
[local]RedBack(config-ctx)#interface enet60
[local]RedBack(config-if)#ip address 10.1.1.1 255.255.255.0
[local]RedBack(config-if)#ip arp arpa
.
.
.
[local]RedBack(config-ctx)#router-id 10.1.1.1
[local]RedBack(config-ctx)#router ospf
[local]RedBack(config-ospf)#spf-timers 5 10
[local]RedBack(config-ospf)#precedence 15 150
[local]RedBack(config-ospf)#area 0
[local]RedBack(config-ospf-area)#ospf-interface 10.1.1.1 broadcast
[local]RedBack(config-ospf-interface)#hello-interval 20
[local]RedBack(config-ospf-interface#routerdead-interval 80
[local]RedBack(config-ospf-interface)#retransmit-interval 10
[local]RedBack(config-ospf-interface)#transmit-delay 2
[local]RedBack(config-ospf-interface)#authentication simple test123
[local]RedBack(config-ospf-interface)#cost 2
.
.
.
[local]RedBack(config)#port ethernet 6/0
[local]RedBack(config-port)bind interface enet60 a
```

With this configuration, OSPF routing is enabled for context `a` and the context contains an Ethernet interface (`enet60`) with an IP address of `10.1.1.1` and bound to Ethernet slot 6/port 0. Context `a` is also assigned a router ID of that IP address (`10.1.1.1`). Within OSPF `area 0`, an OSPF-enabled interface `10.1.1.1` is configured as a `broadcast` type with several OSPF parameter settings, including a `simple` password (`test 123`) and a `cost` of `2`.

Configuring BGP

This chapter provides an overview of the Border Gateway Protocol (BGP) and describes the tasks involved in configuring BGP features through the Access Operating System (AOS).

For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "BGP Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

Redback supports BGP-4 as described in RFC 1771, *Border Gateway Protocol 4 (BGP-4)*. BGP is an Exterior Gateway Protocol (EGP) based on distance-vector algorithms, and uses the Transmission Control Protocol (TCP) as its transport protocol. BGP is a protocol between exactly two BGP nodes, or BGP speakers. First, the TCP connection is established and then the two BGP speakers exchange dynamic routing information over the connection. The exchange of messages is a BGP session between BGP peers.

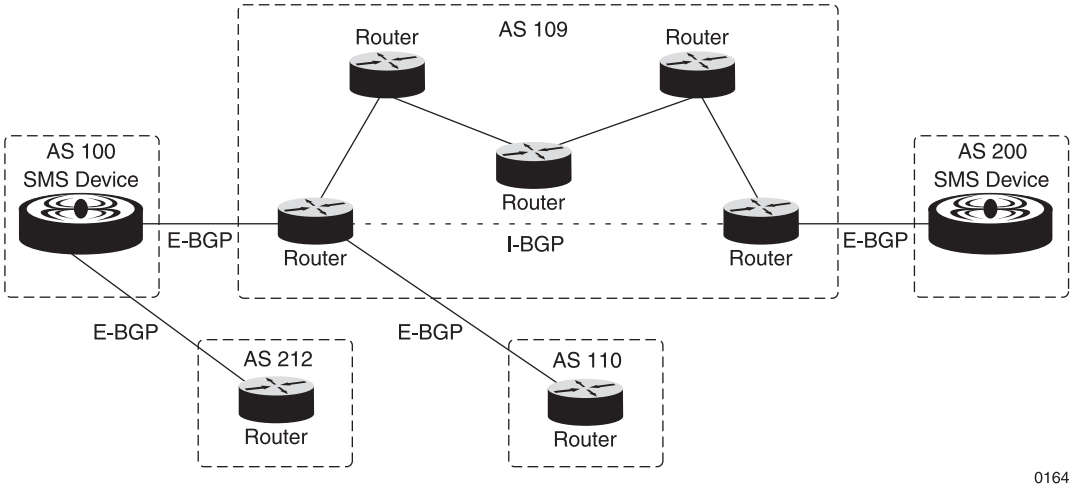
Routers that belong to the same autonomous system (AS) and exchange BGP updates are running internal BGP (I-BGP), and routers that belong to different autonomous systems and exchange BGP updates are running external BGP (E-BGP).

With Redback's implementation:

- Both I-BGP and E-BGP are supported
- Confederations can be set up
- Route reflection is supported
- Only one instance of BGP routing can be configured per context.

Figure 34-1 illustrates the concept of autonomous systems and I-BGP versus E-BGP.

Figure 34-1 I-BGP and E-BGP Networks



BGP Messages

BGP systems send four types of messages:

- Open
- Update
- Keepalive
- Notification

Open

After a TCP connection is established between two BGP systems, they exchange BGP open messages to create a BGP connection between them. The open message consists of the following fields:

- Version

  Specifies the BGP version that the sender of the open message is using. If the version of the two BGP nodes does not match, the TCP connection is closed. The BGP node that initiated the connection can try again by opening a new connection and then behaving according to the correct BGP version.

- Local autonomous system

  Provides the autonomous system number (ASN) of the sender of the open message. A BGP node is configured with its own ASN and the ASN of each BGP neighbor. If the ASN value between nodes matches, BGP processing continues; otherwise, the TCP connection is closed.

- Hold time

  Specifies maximum length of time, in seconds, that the sender of the open message waits to hear either an update or keepalive message from the other node before assuming the BGP session is down.

- BGP identifier

  Contains a value used to identify the BGP speaker. The value is typically one of the IP addresses assigned to the BGP speaker.

Update

Update messages are exchanged between BGP systems to determine the reachability and relationships of all known autonomous systems. Update messages contain the following fields:

- Withdrawn routes length

  The length of the withdrawn route field in octets.

- Withdrawn routes

  A list of prefixes for which the sender of the update message no longer forwards packets due to reconfiguration or inoperability.

- Total path attributes field

  A single set of BGP attributes that apply to all of the prefixes listed in the network layer reachability information field. Attributes describe how prefixes were routed by BGP, the path of autonomous systems through which prefixes have been advertised until this point, and metrics determining the degree of preference for the prefixes.

- Network layer reachability information

  The list of prefixes advertised. To advertise multiple prefixes in a single update message, the BGP speaker must advertise only prefixes that share all of the attributes in the total path attributes field.

Notification

When an error occurs during a BGP session, a notification message can signal the presence of the error before the TCP connection is closed, allowing the administrator to determine why the session failed. Immediately after a notification message is sent, the TCP connection is closed.

Keepalive

BGP neighbors send keepalive messages to confirm that the connection between them is still active. The interval between messages is determined by the hold timer and the frequency at which update messages are sent.

Best AS Path Determination

When advertising a network prefix, the complete path to the prefix is included. This path consists of the sequence of autonomous systems, called the AS path, that are traversed as traffic is forwarded from the advertising router to the destination prefix. The first number in the AS path is the AS closest to the local BGP speaker; the last number is the AS farthest from the local BGP speaker, typically the origin of the path.

Path attributes include:

- AS path

  Whenever an update message passes through an AS, BGP adds its ASN to the update. The AS path attribute, then, is the list of autonomous systems that an update message has traversed to reach its destination. When aggregates are used, the more specific routes are combined to form an unordered set called the AS-SET.

- Origin

  The origin of a route can have one of three values—IGP (a route interior to the originating AS), EGP (a route exterior to the originating AS), or incomplete (unknown route or a route learned via another way). An incomplete route could indicate a route redistributed from another protocol; for example, Open Shortest Path First (OSPF) or Routing Information Protocol (RIP).

- Next hop

  The BGP next-hop attribute is the IP address of the next hop that is used to reach a certain destination.

- Multi-Exit Discriminator (MED)

  The MED attribute enables the SMS device to select the optimal exit point (among multiple points) to a remote AS. If all other factors in determining an exit point are equal, the exit point with the lowest MED metric is preferred. If a MED is received over an external BGP link, it is propagated over internal links within the AS. When the update is sent on to another AS, the MED is reset to 0.

- Local-pref

  The local-pref attribute can be used to select among multiple paths (possibly learned from peers in different autonomous systems) to the same prefix. The local-pref attribute allows preferences to be set through configuration. Any AS that is assigned a local-pref value higher than any other AS becomes the preferred path.

- Atomic-aggregate

  The atomic-aggregate attribute ensures that BGP speakers with overlapping routes do not advertise these routes twice. When a BGP speaker receives a prefix with the atomic-aggregate attribute set, the BGP speaker cannot deaggregate the prefix into more-specific entries in the routing table.

- Aggregator

  An aggregator attribute specifies the AS and BGP speaker that performed the aggregation of routes.

I-BGP Route Reflectors

Typically, I-BGP speakers must be fully meshed. Any BGP speaker that receives messages from an external router must advertise the routes it receives to all BGP speakers in its autonomous system. However, if a route reflector is configured, while it must have connections to all other BGP speakers in the AS, not all other BGP speakers must be fully meshed. When a BGP speaker in the AS receives messages from an external router, it is sufficient to advertise these routes only to the route reflector, which then re-advertises the routes to all other BGP speakers in the AS.

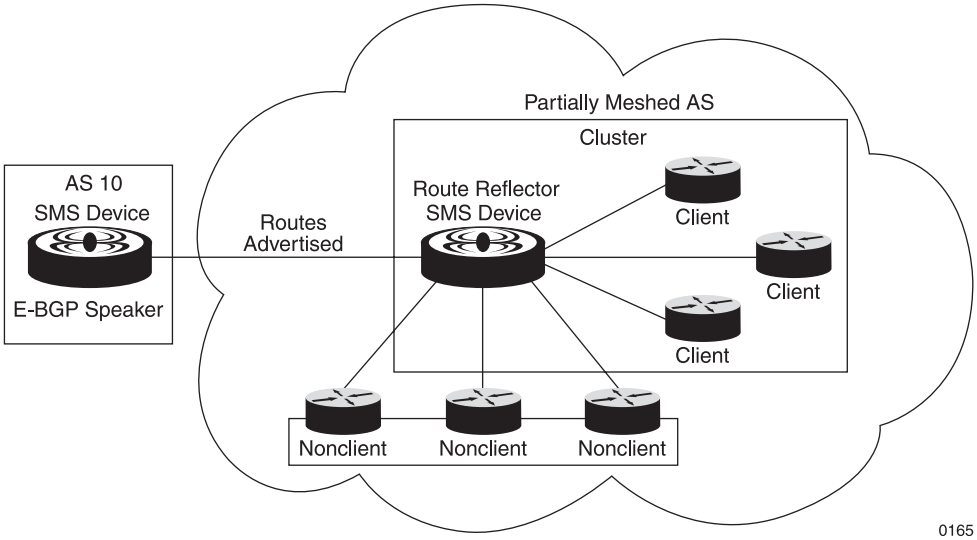
Internal peers of the route reflector are divided into two groups: client peers and non-client peers. A route reflector reflects routes between these two groups. The route reflector and its client peers form a *cluster*. Non-client peers must be fully meshed with each other. Client peers are not required to be fully meshed and do not communicate with BGP speakers outside their cluster. In cases where it is required, peer client-to-peer client route reflection can be disabled.

When the route reflector receives an advertised route:

- Any route from an external BGP speaker is advertised to all peers.
- Any route from a non-client peer is advertised to all client peers.
- Any route from a client peer is advertised to all peers.

Figure 34-2 shows an example I-BGP networking using route reflection.

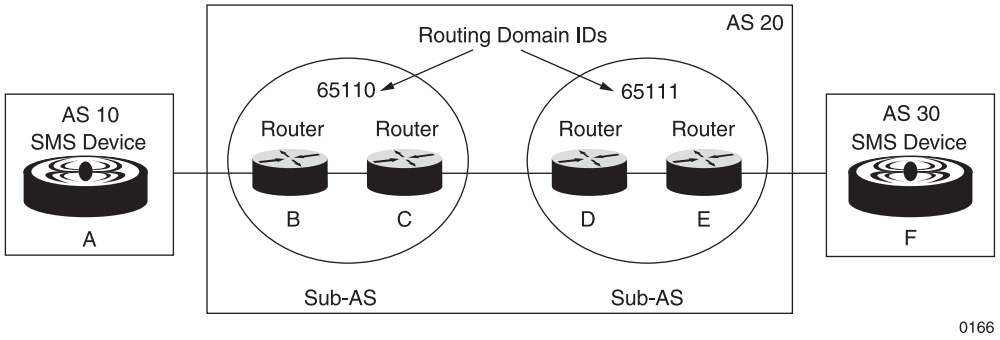
Figure 34-2 I-BGP Network Using Route Reflection



I-BGP Confederations

Another way to reduce I-BGP mesh is to divide an AS into subautonomous systems grouped by a routing domain identifier. The AS and its subautonomous systems are part of the same confederation. To the outside world, the confederation looks like a single AS. Each subautonomous system is fully meshed within itself and has a few connections to other subautonomous systems in the confederation. Neighbors from other subautonomous systems are treated as special E-BGP peers. Even though peers in different subautonomous systems engage in E-BGP sessions, they exchange routing information as if they were I-BGP peers. Specifically, the next-hop, the MED, and local preference information is preserved, so that a single IGP is used for all of the subautonomous systems; see Figure 34-3.

Figure 34-3 I-BGP Confederation



Route Aggregation

BGP4 supports Classless InterDomain Routing (CIDR). With CIDR, routers use the network prefix to determine the dividing point between the network number and the host number. For example, the range of addresses 128.186.1.0 to 128.186.1.255 can be represented as the network prefix 128.186.1.0/24; the 24 indicates that all addresses in the segment agree in their first 24 bits.

In addition, CIDR does not require a network to be of standard size, as is the case in classful addressing, which provides 8-bit (Class A), 16-bit (Class B), and 24-bit (Class C) network deployment. This flexibility in CIDR enables the creation of arbitrarily-sized networks.

Of particular importance is CIDR's ability to lend itself to the concept of route aggregation. The Internet is divided into addressing domains. Within a domain, detailed information is available about all of the networks that reside in the domain. Outside of an addressing domain, however, only the common network prefix is advertised. By allowing a single routing table entry to specify a route to many individual network addresses, aggregation minimizes the size of the routing table. A router cannot aggregate an address if it does not have a more specific route of that address in the BGP routing table. More-specific routes can be injected in the BGP routing table by incoming updates from other autonomous systems.

Configuration Tasks

To configure BGP, perform the tasks in the following sections:

- Enable BGP Routing
- Define Global BGP Parameters
- Configure BGP Group Characteristics
- Configure BGP Peer Characteristics
- Display BGP Information
- Clear Entries in the BGP Routing Table
- Enable IP BGP Debugging Messages

Enable BGP Routing

You can enable one BGP routing process per context. To enable BGP, you must first set the router ID and specify an ASN, and, optionally, a routing domain, for the BGP routing process. Then, configure a BGP group, and populate the group with peers belonging to the same autonomous system, or to the same confederation.

To enable BGP routing:

1. Set the BGP identifier using the following command in context configuration mode:

   router-id *ip-address*

2. Configure BGP routing using the following command in context configuration mode:

   router bgp *asn* [**routing-domain** *id*]

   Each context running BGP must be configured with its own ASN. For a topology with BGP confederations, this ASN is the confederation identifier (externally visible AS), whereas the routing domain identifier is the internally visible sub-AS.

3. Configure a BGP group and enter BGP group configuration mode using the following BGP configuration mode command:

   group *group-name* [**confederation**] **remote-as** *asn*

   Give the group a name and use the **remote-as** *asn* construct to assign the group an ASN. Or, give the group a name, include the optional **confederation** keyword, and assign a routing domain identifier using the **remote-as** *asn* construct.

4. Configure a BGP peer and enter BGP peer configuration mode using the following command in BGP group configuration mode:

   neighbor *ip-address*

5. Enable the peer using the following command in BGP peer configuration mode:

   enable-peer

Define Global BGP Parameters

Optionally, you can define parameters that affect the overall operation of BGP within the context.

Note All of the commands in this section are found in BGP configuration mode.

Configure Aggregate Entries in the BGP Routing Table

To create an aggregate entry in the routing table, enter the following command:

aggregate-address *address netmask* [**summary-only**] [**as-set**]

The optional **summary-only** keyword suppresses advertisements of more specific routes to neighbors. The optional **as-set** keyword generates AS path information.

Enable MED Comparisons Between Different Autonomous Systems

To enable MED comparison for paths from peers in different autonomous systems, enter the following command:

always-compare-med

Enable the Export of Nonactive Routes

To configure the SMS device to export locally inactive BGP routes to peers in the scenario where the active non-BGP route is prevented from export by policy, enter the following command:

export-non-active

Modify the Precedence for BGP-Learned Routes

To set the precedence for routes learned through BGP, enter the following command:

precedence *pref*

The SMS device assigns a default value to each routing protocol process that the routing table receives. The default value depends on the source of the route.

Redistribute Routes Learned via Other Protocols into BGP

To redistribute routes learned from other protocols into BGP, enter the following command:

redistribute {**direct** | **ospf** | **rip** | **rip** | **static** | **subscriber**} [**route-map** *map-name*] [**metric** *metric*]

Routes from protocols, such as OSPF and RIP, can be redistributed into BGP domains. Routes can be filtered through the application of a route map. You can also specify a MED value to the route—this value is sent to BGP peers.

Configure a Cluster ID for Route Reflection

To configure a route reflector cluster ID, enter the following command:

cluster-id *id*

If a route reflection cluster has more than one route reflector, all route reflectors in the cluster must be configured with the same 4-byte cluster ID. The common cluster ID allows one route reflector to recognize updates from other route reflectors in the same cluster.

Configure BGP Group Characteristics

Any characteristic configured via a BGP group configuration mode command will apply to a peer that resides in the group, unless that peer has specifically been configured with a different value using the identical command in BGP peer configuration mode.

Note All of the commands in the this section are found in BGP group configuration mode.

Enable Route Reflector Clients

The SMS device acts as a route reflector when peers in a group are configured as route reflector clients:

route-reflector-client

When a route reflector receives an advertised route:

- Any route from an external BGP speaker is advertised to all peers
- Any route from a non-client peer is advertised to all client peers
- Any route from a client peer is advertised to all peers

Disable Client-to-Client Route Reflection

In the case where route reflection is configured and client-to-client peering may already be configured, you can ensure that routes learned from a client are not reflected to other clients by entering the following command:

no client-to-client

Accept a MED Value Offered by a Peer

To accept the MED value offered by an external peer, enter the following command:

accept-med

Set the MED Value Sent to External Peers

To set the MED value to send to external peers, enter the following command:

metric-out *metric*

Allow Sending Default Route to Peers

To allow the sending of default route (0.0.0.0) to peers belonging to the BGP group, enter the following command:

default-originate

Set the Interval Required Before a Route Can Be Exported to BGP

To set the amount of time a route must be present in the routing table before the route can be exported to the BGP routing process, enter the following command:

out-delay *delay*

Set the Maximum Hold Time Interval

To set the maximum interval allowed between successive keepalive or update messages sent by a remote peer before the SMS device drops the BGP session, enter the following command:

hold-time *holdtime*

Set the Maximum Number of Allowed Prefixes

To set the maximum number of network prefixes the SMS device accepts from a peer before dropping the BGP session, enter the following command:

maximum-prefix *max-prefix*

To set the maximum number of network prefixes the SMS device accepts from a peer before logging a warning message, enter the following command:

maximum-prefix-warn *threshold*

Prevent Sending Third-Party Next-Hop Information

To ensure the SMS device does not send third-party next-hop information to peers, enter the following command:

nexthop-self

Prevent Creation of Aggregate Routes Using Different ASNs

To prevent BGP devices within the same AS from creating aggregate routes that contain different AS paths, enter the following command:

no-aggregator-id

Configure the SMS to Wait for a Peer to Initiate a Connection

To configure the SMS device to not send open messages to a peer for initiation of a BGP connection, enter the following command:

passive

In this case, the SMS device waits for a peer to send it open messages for initiation of a BGP connection.

Modify the Precedence for BGP-Learned Routes

To modify the precedence for routes learned through peers belonging to the group, enter the following command:

precedence *prec*

Modify the Preference Value

In the case where the *value* argument of the **precedence** command among two or more routes is equal, break the tie by entering the following command:

preference *pref*

Strip the Private ASN from BGP Updates

To strip the private AS number from BGP updates sent to external peers, enter the following command:

remove-private-AS

Apply a Route Map to BGP Updates

To apply a route map to incoming or outgoing BGP updates sent or from to peers in the group, enter the following command:

route-map *map-name* [**in** | **out**]

The name of route map applied is be configured through the **route-map** command in context configuration mode. See the "Create Route Maps" section in Chapter 35, "Configuring Routing Policies."

Modify the BGP Update Message Rate

To modify the rate at which BGP update messages are sent to peers, enter the following command:

throttle *rate*

Modify the Time-to-Live Value

To modify the time-to-live (TTL) value for IP packets containing BGP messages when communicating with peers, enter the following command:

ttl *seconds*

Configure BGP Peer Characteristics

If a peer belongs to a group and no characteristics have been configured for the peer, the peer will have the same values that have been assigned to the group. However, any values configured in BGP peer configuration mode will override values set using identical commands in BGP group configuration mode.

Note All of the commands in this section are found in BGP peer configuration mode.

Accept a MED Value Offered by a Peer

To accept the MED value offered by an external peer, enter the following command:

accept-med

Allow BGP Sessions with Peers that Have Invalid Router IDs

To allow BGP sessions with peers that have invalid router IDs, enter the following command:

allow-bad-routerid

Set the Interval Required Before a Route Can Be Exported to BGP

To set the amount of time a route must be present in the routing table before the route can be exported to the BGP routing process, enter the following command:

out-delay *delay*

Set the Maximum Hold Time Interval

To set the maximum interval allowed between successive keepalive or update messages sent by a remote peer before the SMS device drops the BGP session, enter the following command:

hold-time *holdtime*

Set the Maximum Number of Allowed Prefixes

To set the maximum number of network prefixes the SMS device accepts from a peer before dropping the BGP session, enter the following command:

maximum-prefix *max-prefix*

To set the maximum number of network prefixes the SMS device accepts from a peer before logging a warning message, enter the following command:

maximum-prefix-warn *threshold*

Prevent the Sending of Third-Party Next-Hop Information

To configure the SMS device to not send third-party next-hop information to peers, enter the following command:

nexthop-self

Prevent the Creation of Aggregate Routes Using Different ASNs

To prevent BGP devices within the same AS from creating aggregate routes that contain different AS paths, enter the following command:

no-aggregator-id

Configure the SMS to Wait for a Peer to Initiate a Connection

To configure the SMS device to not send open messages to a peer for initiation of a BGP connection, enter the following command:

passive

In this case, the SMS device waits for a peer to it send open messages for initiation of a BGP connection.

Modify the Precedence for BGP-Learned Routes

To modify the precedence for BGP routes learned through peers, enter the following command:

precedence *prec*

Modify the Preference Value

In the case where the *value* argument of the **precedence** command among two or more routes is equal, break the tie by entering the following command:

preference *pref*

Strip the Private ASN from BGP Updates

To strip the private ASN from BGP updates sent to external peers, enter the following command:

remove-private-AS

Apply a Route Map to BGP Updates

To apply a route map to incoming BGP updates sent to the peer, enter the following command:

route-map *map-name* **in**

The name of route map applied is be configured through the **route-map** command in context configuration mode. See the "Create Route Maps" section in Chapter 35, "Configuring Routing Policies."

Modify the TTL Value

To modify the TTL value for IP packets containing BGP messages when communicating with peers, enter the following command:

ttl *seconds*

Display BGP Information

To obtain information about BGP, use any or all of the **show ip bgp** commands described in this section. All of these commands are available in operator exec mode.

Show Global Information

To display global BGP information, enter the following command:

show ip bgp [*ip-address*]

Show BGP Groups

To display BGP group information, enter the following command:

show ip bgp groups [*group-name*]

Show BGP Neighbors

To display BGP neighbor information, enter the following command:

show ip bgp neighbors [*ip-address*]

Show BGP AS Paths

To display information about BGP AS paths, enter the following command:

show ip bgp paths

Show BGP Summary Information

To display a summary of BGP information, enter the following command:

show ip bgp summary

Clear Entries in the BGP Routing Table

To reset a single BGP connection, all connections, or BGP connections for all members of a group, enter the following command in administrator exec mode:

clear ip bgp {*ip-address* | **all** | **group** *group-name* [**soft** [**in** | **out**]]}

Enable IP BGP Debugging Messages

To allow debugging of BGP events and packets, enter the following command in administrator exec mode:

debug ip bgp [**all** | **events** | **keepalives** | **misc** | **packets** | **updates**]

Configuration Examples

The following example configures a router ID of `1.1.1.1` for the local context. The local ASN is `1`. A peer session is established with an external peer at IP address `20.1.1.2` in ASN `2`.

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#router-id 1.1.1.1
[local]RedBack(config-ctx)#router bgp 1
[local]RedBack(config-bgp)#group as2 remote-as 2
[local]RedBack(config-group)#neighbor 20.1.1.2
[local]RedBack(config-peer)#enable-peer
```

In following example, the local autonomous system used by the BGP process is `2`. A peer session will be established with internal peers at IP addresses `30.1.1.2` and `35.1.1.2`, respectively.

```
[local]RedBack(config)#context corpA
[local]RedBack(config-ctx)#router-id 1.1.1.1
[local]RedBack(config-ctx)#router bgp 2
[local]RedBack(config-bgp)#export-non-active
[local]RedBack(config-bgp)#group as2 remote-as 2
[local]RedBack(config-group)#route-reflector-client
[local]RedBack(config-group)#neighbor 30.1.1.2
[local]RedBack(config-peer)#enable-peer
[local]RedBack(config-peer)#exit
[local]RedBack(config-group)#neighbor 35.1.1.2
[local]RedBack(config-peer)#enable-peer
[local]RedBack(config-peer)#exit
[local]RedBack(config-bgp)#group as4 remote-as4
[local]RedBack(config-group)#neighbor 10.1.1.2
[local]RedBack(config-group)#enable-peer
```

Configuring Routing Policies

This chapter provides an overview of and describes the tasks involved in configuring routing policies through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed under the "Configuration Tasks" section, see the "Routing Policy Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

Routing policies allow network administrators to enforce various routing policy decisions onto incoming, outgoing, and redistributed routes. The tools used to configure routing policies include autonomous system (AS) access control lists, community lists, route access control lists, and the application of match criteria and set actions used in conjunction with route maps.

Configuration Tasks

To configure routing policies, perform the tasks in the following sections:

- Configure AS Path Access Control Lists
- Configure Community Lists
- Configure Route Access Control Lists
- Create Route Maps
- Determine Route Map Match Criteria
- Determine Route Map Set Actions
- Display Routing Policies

Configure AS Path Access Control Lists

To configure an AS path access control list, enter the following command in context configuration mode:

as-path access-list *list-num* **seq** *seq-num* {**deny** | **permit**} *as-reg-exp*

The sequence number indicates the position this AS path access control list has with respect to other AS path access control lists using the same *list-num* argument. The AS path access control list with the lowest sequence number is looked at first by the system. A **deny** statement causes any route matching the criteria to be dropped. A **permit** statement causes any route matching the criteria to be accepted. The optional *wildcard* argument allows corresponding bits in the *ip-address* argument to be included in the permit or deny criteria.

Each filter is an access control list based on regular expressions. If the regular expression matches the representation of the AS path of the route, the **permit** or **deny** condition applies.

When configuring AS path access control lists:

1. Use the **as-path access-list** command to define the AS path access list.
2. Create a route map using the **route-map** context configuration mode command and apply the AS path access list to the route map using the **match as-path** command in route map configuration mode.
3. Use the **route-map** command in BGP group or BGP peer configuration mode to apply the route map to a Border Gateway Protocol (BGP) group or peer.

Note You can apply a route map to incoming and outgoing routes in BGP group configuration mode. However, you can only apply a route map to incoming routes in BGP peer configuration mode.

Configure Community Lists

A community is a group of destinations that share some common attributes. Each destination can belong to multiple communities. To configure a community list, enter the following command in context configuration mode:

community-list *list-num* **seq** *seq-num* {**deny** | **permit**} {*community-num* | **internet** | **local-AS** | **no-advertise** | **no-export**}

The sequence number indicates the position this community list has with respect to other community lists with the same community list number. The community list with the lowest sequence number is looked at first by the system. A **deny** statement causes any route matching the criteria to be dropped. A **permit** statement causes any route matching the criteria to be accepted.

By default, the *community-num* argument is an unsigned decimal value. If the **ip bgp-community** command is enabled, you can use the format, *aa:nn*, where *aa* is the autonomous system number (ASN) and *nn* is a 2-byte number.

When configuring community lists:

1. Use the **community-list** command to define the BGP community list.
2. Create a route map using the **route-map** context configuration mode command and apply the AS path access list to the route map using the **match community** command in route map configuration mode.
3. Use the **route-map** command in BGP group or BGP peer configuration mode to apply the route map to the BGP group or peer.

Note You can apply a route map to incoming and outgoing routes in BGP group configuration mode. However, you can only apply a route map to incoming routes in BGP peer configuration mode.

Configure Route Access Control Lists

To configure a route access control list that allows or prevents acceptance of routes from specified sources or advertisement of routes to specified destinations, enter one of the following commands in context configuration mode:

route-access-list *extended-access-list-num* **seq** *seq-num* {**deny** | **permit**} {*ip-address* | **any**} [*wildcard*] [*netmask* | **any**] [*netmask wildcard*]

route-access-list *standard-access-list-num* **seq** *seq-num* {**deny** | **permit**} {*ip-address* | **any**} [*wildcard*]

The sequence number indicates the position this route access list has with respect to other route access control lists with the same access list number. The route access control list with the lowest sequence number is looked at first by the system. A **deny** statement causes any route matching the criteria to be dropped. A **permit** statement causes any route matching the criteria to be accepted. The optional *wildcard* argument allows corresponding bits in the *ip-address* argument to be included in the permit or deny criteria.

An extended access list number adds the capability of filtering any set of prefix lengths combined with any set of network numbers.

When configuring route access control lists:

1. Use the **route-access-list** command to define the route access control list.
2. Create a route map using the **route-map** context configuration mode command and apply the AS path access list to the route map using the **match ip address** command in route map configuration mode. The specified route access control list number must be matched in order for the route to be allowed or denied distribution.
3. Use the **route-map** command in BGP group or BGP peer configuration mode to apply the route map to the BGP group or peer.

Note You can apply a route map to incoming and outgoing routes in BGP group configuration mode. However, you can only apply a route map to incoming routes in BGP peer configuration mode.

Create Route Maps

Route maps allow administrators to have detailed control over which incoming and outgoing routes are permitted or denied.

1. To configure a route map, enter the following command in context configuration mode:

   route-map *map-name* [**deny** | **permit**] [*seq-num*]

   You can create several route maps with the same name, but with different conditions applied to each. A sequence number indicates the position this route map has with respect to other route maps with the same name. The route map with the lowest sequence number is looked at first by the system.

2. Use the route map configuration mode commands, **match** and **set**, to specify the conditions under which distribution is allowed for routes, and to dictate the actions to perform on those routes if the conditions are met.

   See the "Determine Route Map Match Criteria" and "Determine Route Map Set Actions" sections for a list of **match** and **set** command configuration tasks.

   If the criteria set by the **match** command are met and **deny** is specified, the route is not distributed. No further route-map sequences that share the same *map-name* argument are examined.

   If the criteria set by the **match** command are met for this route map, and the **permit** keyword is specified, the route is distributed according to the criteria specified by the **set** command.

   If the match criteria are not met and the **permit** keyword is specified, the next route-map sequence with the same *map-name* argument is tested.

   If a route passes none of the match criteria for a set of route-map sequences that share the same *map-name* argument, it is not distributed.

3. Apply the route map to either or both of the following:

   a. A BGP group or peer by using the **route-map** command in BGP group or BGP peer configuration mode. Only incoming routes can be applied at the BGP peer level.

   b. Routes that are redistributed into the BGP routing process by using the **route-map** keyword with the **redistribute** command found in BGP configuration mode.

Determine Route Map Match Criteria

Use **match** commands to specify conditions under which incoming or outgoing routes are distributed. There must be at least one **match** statement associated with a route map. All of the commands described in this section are entered in route map configuration mode.

Distribute Routes that Pass the AS Path Access List Conditions

To match the autonomous system path access list, enter the following command:

match as-path *list-num* [*...list-num*]

Distribute Routes with a Matching BGP Community List

To distribute routes with a matching BGP community list, enter the following command:

match community-list *list-num* [*...list-num*]

Distribute Routes Connecting to a Next Hop via a Matching Interface

To distribute routes that connect to a next hop via the named interface, enter the following command:

match interface *if-name* [*...if-name*]

Distribute Routes with a Permitted Destination IP Address

To distribute routes that have a destination IP address permitted by the specified route access list or lists, enter the following command:

match ip address *list-num* [*...list-num*]

Distribute Routes with a Permitted Next-Hop IP Address

To distribute routes with a next-hop IP address that is permitted by the specified route access list or lists, enter the following command:

match ip next-hop *list-num* [*...list-num*]

Distribute Routes with a Matching Metric Value

To distribute routes with a matching metric (MED) value, enter the following command:

match metric *metric*

Distribute Routes with a Matching Type

To distribute routes that match the type specified, enter the following command:

match route-type {**local** | **internal** | **external** [**type-1** | **type-2**]}

Distribute Routes with a Matching Tag

To distribute routes that match the specified tag value, enter the following command:

match tag *tag*

Determine Route Map Set Actions

Use **set** commands to determine the action that is performed on routes that have met **match** command criteria. All of the commands described in this section are entered in route map configuration mode.

Set the AS Path for BGP Routes

To modify an AS path for BGP routes, enter the following command:

set as-path prepend *asn*

Set the BGP Community Attribute

To set the BGP community attribute, enter the following command:

set community {*community-num* [**additive**] | **none**}

Set the Next-Hop IP Address for Packet Forwarding

To determine the next-hop IP address for forwarding packets, enter the following command:

set ip next-hop {*ip-address* [*...ip-address*] | **peer-address**}

Set the AS Path Preference

To set the degree of preference for the AS path, enter the following command:

set local-preference *pref*

Modify the Metric Value for the Destination Routing Protocol

To modify the metric value for the destination routing protocol, enter the following command:

set metric [+ | -] *metric*

Set the BGP Origin Code

To set the BGP origin code, enter the following command:

set origin {**egp** | **igp** | **incomplete**}

Set the Degree of Preference for BGP-Learned Routes

To set the degree of preference for BGP-learned routes, enter the following command:

set preference *pref*

Display Routing Policies

Use the commands described in this section to display information about configured routing policies. All commands are entered in operator exec mode.

Display Route Maps

To display all configured route maps for the current context, enter the following command:

show route-map [*map-name*]

Display AS Path Access Lists

To display configured AS path access lists, enter the following command:

show as-path-access-list [*list-num*]

Display Community Lists

To display configured community lists, enter the following command:

show community-list [*list-num*]

Display Route Access Lists

To display configured route access lists, enter the following command:

show route-access-list [*list-num*]

Configuration Examples

The following example configures a BGP routing process in `context A` with autonomous system 11 talking to two BGP peers, one via interface `enet60` and one via interface `enet40`.

Static routes in `context A` are advertised to the remote peers subjecting to filtering by `route-map rdist-a`. Announced routes from peer `21.1.1.2` are accepted and are advertised to peer `12.1.1.2` without any filtering. Routes advertised to peer `21.1.1.2` are subject to another filtering by `route-map permit-all-routes`. Announced routes from peer `12.1.1.2` are filtered based on the community attribute before acceptance.

```
[local]RedBack(config)#context A
[local]RedBack(config-ctx)#router-id 11.11.11.11
[local]RedBack(config-ctx)#interface enet60
.
.
.
[local]RedBack(config-if)#ip address 21.1.1.1 255.255.255.0
[local]RedBack(config-if)#ip arp arpa
.
.
.
[local]RedBack(config-ctx)#interface enet40
[local]RedBack(config-if)#ip address 12.1.1.1 255.255.255.0
[local]RedBack(config-if)#ip arp arpa
.
.
.
[local]RedBack(config-ctx)#ip route 28.0.0.0 255.0.0.0 12.1.1.6 enet40
[local]RedBack(config-ctx)#ip route 77.0.0.0 255.0.0.0 21.1.1.9 enet60
[local]RedBack(config-ctx)#ip route 81.1.10.0 255.255.255.0 21.1.1.9 enet60
[local]RedBack(config-ctx)#ip route 81.1.20.0 255.255.255.0 21.1.1.9 enet60
[local]RedBack(config-ctx)#ip route 82.0.0.0 255.0.0.0 12.1.1.6 enet40
[local]RedBack(config-ctx)#router bgp 11
[local]RedBack(config-bgp)#redistribute static route-map rdist-a
[local]RedBack(config-bgp)#group as3 remote-as 3
[local]RedBack(config-group)#neighbor 21.1.1.2
[local]RedBack(config-peer)#route-map permit-all-routes out
[local]RedBack(config-peer)#enable-peer
.
.
.
```

```
[local]RedBack(config-bgp)#group as12 remote-as 12
[local]RedBack(config-group)#neighbor 12.1.1.2
[local]RedBack(config-peer)#route-map filter-on-comm in
[local]RedBack(config-peer)#enable-peer
.
.
.
[local]RedBack(config-ctx)#route-map filter-on-comm permit 5
[local]RedBack(config-route-map)#match community 2
[local]RedBack(config-route-map)#match as-path 5
.
.
.
[local]RedBack(config-ctx)#route-map filter-on-comm permit 15
[local]RedBack(config-route-map)#match community 3
[local]RedBack(config-route-map)#set as-path prepend 11 11
.
.
.
[local]RedBack(config-ctx)#route-map permit-all-routes permit 10
[local]RedBack(config-route-map)#match ip address 50
[local]RedBack(config-route-map)#set metric 1500
.
.
.
[local]RedBack(config-ctx)#route-map rdist-a permit 10
[local]RedBack(config-route-map)#match ip address 8
[local]RedBack(config-route-map)#set metric 2000
.
.
.
[local]RedBack(config-ctx)#as-path access-list 5 seq 10 permit .* 27 89
[local]RedBack(config-ctx)#community-list 2 seq 5 permit local-AS
[local]RedBack(config-ctx)#community-list 3 seq 5 permit 65012:10
[local]RedBack(config-ctx)#route-access-list 8 seq 5 deny 81.1.10.0 0.0.0.255
[local]RedBack(config-ctx)#route-access-list 8 seq 10 permit any
[local]RedBack(config-ctx)#route-access-list 50 seq 100 permit any
.
.
.
[local]RedBack(config)#port ethernet 4/0
[local]RedBack(config-pvc)#bind interface enet40 A
.
.
.
[local]RedBack(config)#port ethernet 6/0
[local]RedBack(config-pvc)#bind interface enet60 A
```

Configuring IGMP Proxy

This chapter provides an overview of interfaces and describes the tasks involved in configuring Internet Group Management Protocol (IGMP) proxy through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "IGMP Proxy Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

The IGMP proxy feature allows the Subscriber Management System (SMS) device to forward IP multicast traffic without running a multicast routing protocol. The SMS device acts as a proxy for hosts directly attached to it, periodically sending IGMP reports for all multicast group members on any subnet to the multicast router. In this manner, the SMS device protects the multicast router from having to communicate directly with thousands of subscribers. The AOS IGMP proxy features support IGMP versions 1 and 2

IGMPv2 is defined in RFC 2236, *Internet Group Management Protocol, Version 2*. Redback Networks supports the following implementation:

- A separate instance of IGMP proxy is required per context.
- Each context in the SMS device requires the configuration of only one interface to which the true multicast router is attached. This interface is designated as the multicast router interface.
- The multicast router interface must be bound to a single circuit. Multicast data is forwarded out of the multicast router interface, regardless of whether a multicast router is detected on that interface or not.

- Because the SMS device simply forwards multicast traffic and is not actually running a multicast routing protocol, any context in which IGMP proxy is configured only be deployed on stub networks; that is, in network topologies with a single multicast-forwarding path originating from the multicast router interface on the SMS device toward a backbone network.

  A network with multiple forwarding paths from the SMS device to the backbone is not supported. Having the SMS device communicate with more than one multicast router can result in the formation of multicast loops in the network. In networks where multiple multicast routers are required, multicast traffic must be tunneled (IP-in-IP). Traveling through the SMS device, tunneled multicast traffic is indistinguishable from regular unicast IP traffic.

- Generated IGMP proxy-system log messages include unexpected queries received and number of groups exceeded.
- Simple Network Management Protocol (SNMP) information is provided through the IGMP Management Information Base (MIB).
- If Remote Authentication Dial-In User Service (RADIUS) authorization is used, all multicast related attributes are obtained from the RADIUS server. Configuration information can be specified in the subscriber default record or in an individual subscriber record. Vendor-specific attributes (VSAs) required for RADIUS are listed in Appendix C, "RADIUS Attributes."

Multicast Groups

Traditional IP communication allows a host to send packets to a single host (unicast transmission) or to all hosts (broadcast transmission). IP multicast provides a third scheme, allowing a host to send packets to a subset of all hosts (group transmission). These hosts are known as group members.

Membership in a multicast group is dynamic; hosts can join and leave at any time. There is no restriction on the location or number of members in a multicast group. A host can be a member of more than one multicast group at a time. How active a multicast group is and what members it has can vary from group-to-group and from time-to-time. A multicast group can be active for a long time, or it can be very short-lived. Membership in a group can change constantly. A group that has members can have no activity.

Routers use IGMP to *query*, or learn, whether members of a group are present on their directly attached subnets. IP hosts *report* their group membership to directly connected multicast routers. Packets delivered to group members are identified by a single multicast-group address. Senders use that address as the destination address of a datagram to reach all members of the group.

IP multicast uses group addresses, which are Class D IP addresses, to send and receive multicast data. Unlike Class A, B, and C IP addresses, the last 28 bits of a Class D address have no structure. The multicast group address is the combination of the high-order 4 bits of 1110 and the multicast group ID. Host group addresses can be in the range of 224.0.0.1 to 239.255.255.255.

Note Multicast addresses in the range of 224.0.0.0 to 224.0.0.255 are considered link-local addresses and are not forwarded by the SMS device.

Routers use multicast routing protocols to successfully forward multicast datagrams to other routers. Examples of commonly used multicast protocols are the Distance-Vector Multicast Routing Protocol (DVMRP), used on the multicast backbone (MBONE) of the Internet, and Protocol Independent Multicast (PIM).

IGMP Proxy Event Sequence

IGMP proxy events occur in the following order:

1. The SMS device receives an IGMP query from the backbone multicast router and responds with an IGMP report for each multicast group on attached subscriber circuits. The true multicast router then uses this membership information in conjunction with a multicast routing protocol to make multicast forwarding decisions.
2. To detect the presence of multicast groups on subscriber circuits, the SMS device periodically sends an IGMP query to each subscriber circuit that has IGMP proxy enabled.
3. Each multicast-capable host receiving the IGMP query waits for a random time interval to see if there is an IGMP report from another host on the same subnet. The host specifically looks for IGMP reports for the same multicast groups that it is attempting to report. If the host does not see this report, it generates an IGMP report of its own.
4. On receipt of the IGMP report, the SMS device makes note of the report's origin and the group to which the report is addressed. The SMS device then ensures that the report is not forwarded to other hosts on the same subnet. In this manner, the SMS device forces at least one station on each attached circuit to send an IGMP report, thereby providing the required membership information.
5. The SMS device uses the information from IGMP reports received to create a table mapping multicast groups to circuits. The mapping table ensures that the multicast traffic is forwarded only on the appropriate circuits. Traffic is forwarded to hosts with membership in the multicast group indicated by the destination IP address of the multicast packet. If a multicast router has been identified, the SMS joins the group on the circuit bound to the interface to which the true multicast router is attached.
6. Data originating on a local subscriber network is forwarded to the multicast router and to any of the SMS device circuits that have members in the targeted multicast group (regardless of subnet).

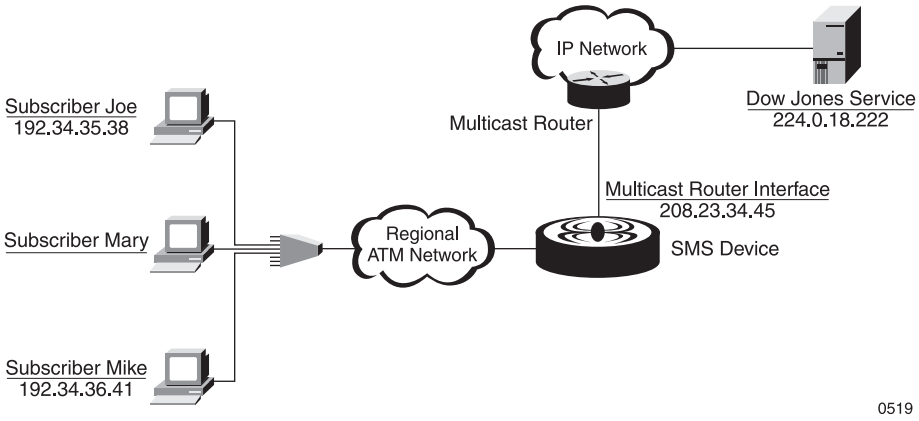
Network Examples

The following examples illustrate a stub network with a context attached to a single multicast router; how service providers can forward multicast traffic on a separate path from unicast traffic; and how multicast hosts can communicate with the SMS device running IGMP proxy.

Stub Network Attached to a Single Multicast Router

Figure 36-1 illustrates a stub network with a context attached to a single multicast router.

Figure 36-1 Single Multicast Router



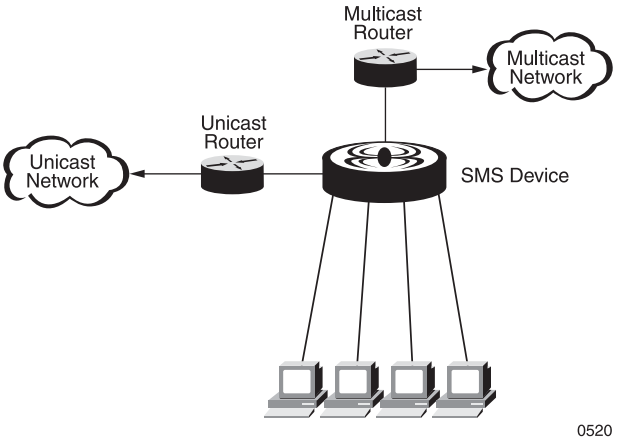
The following steps describe the IGMP proxy interaction between subscribers connected to the SMS device and a Dow Jones multicast service:

1. Hosts residing on different subnets, those of subscriber `Joe` and subscriber `Mike`, start the same ticker tape program.
2. The ticker tape program joins the Dow Jones multicast service group at address `224.0.18.222`.
3. The newly formed multicast group sends an IGMP report.
4. The SMS device sends an IGMP report through its multicast router interface to a multicast router.
5. The Dow Jones multicast service sends multicast data to the group.
6. The multicast router receives the multicast data and forwards it on to the subnet (`208.23.34.45`) connecting the router to the SMS device.
7. The SMS device forwards the multicast data to subscribers `Joe` and `Mike`.
8. The ticker tape programs receive data from the Dow Jones multicast service.

Separate Multicast Router and Unicast Router Paths

In some cases, service providers want to forward multicast traffic on a separate path from unicast traffic. Figure 36-2 illustrates this topology.

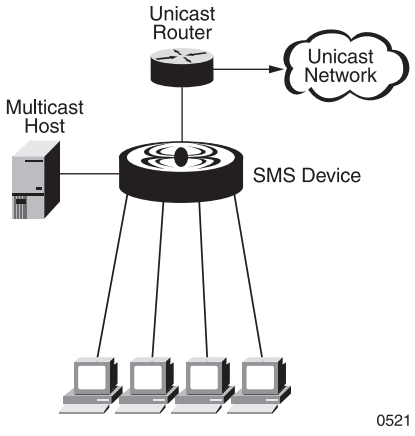
Figure 36-2 Separate Multicast Router and Unicast Router Paths



No Multicast Router but Multicast Hosts

There may be no multicast router on the network, but multicast hosts can communicate with the SMS device running IGMP proxy; Figure 36-3 illustrates this setup.

Figure 36-3 Multicast Host and Unicast Router



Configuration Tasks

To configure IGMP, perform the tasks in the following sections:

- Enable IGMP Proxy (and Limit the Number of Groups Per Context)
- Configure the Interface That Connects to the Multicast Router
- Add or Remove Circuits in Multicast Groups
- Limit the Number of Groups a Subscriber Can Join
- Use Access Control Lists to Filter IGMP Query Types
- Modify IGMP Interface Parameters
- Show IGMP Proxy Statistics
- Enable the Logging of IGMP Debug Messages

Enable IGMP Proxy (and Limit the Number of Groups Per Context)

You must enable IGMP proxy for a context and for designated interfaces within that context. In addition, subscribers must be given permission to send or receive IP multicast traffic.

To enable IGMP proxy, perform the following tasks:

1. Enable IP multicast routing so that the SMS device generates IGMP queries, maintains multicast forwarding information, and receives data originating from subscribers. Optionally, you can limit the number of multicast groups allowed within the context.

   Enter the following command in context configuration mode:

   ip multicast-routing [*max-groups*]

2. On designated interfaces, originate IGMP queries and use IGMP responses from hosts on the same subnet to build multicast forwarding tables.

   Enter the following command in interface configuration mode:

   ip igmp

3. Permit or deny subscriber records or default subscriber records to send or receive IP multicast traffic.

   Enter either or both of the following commands in subscriber configuration mode:

   ip multicast send {**permit** [**unsolicit**] | **deny**}

   ip multicast receive {**permit** | **deny**}

Configure the Interface That Connects to the Multicast Router

You can configure only one interface per context as the interface that connects to the true multicast router on the network.

To designate the multicast router interface, perform the following tasks:

1. Access IGMP proxy router configuration mode by entering the following command in context configuration mode:

   router igmp-proxy

2. Identify the interface connected to the multicast router by entering the following command in IGMP proxy router configuration mode:

   router-igmp-interface *if-name*

All multicast data and IGMP reports are sent out on the circuit associated with this interface

Note You must also bind the multicast router interface to only one circuit.

Add or Remove Circuits in Multicast Groups

To statically add circuits to multicast groups, enter the following command:

ip igmp join-group circuit {*slot/port* {*vpi vci* | *hdlc-channel dlci*} | **lac** *vcn* | **lns** *vcn* | **pppoe** *cm-slot-session-id*} *multicast-IP-address*

Use this command in context configuration mode if you want a circuit to retain membership even after a system reset; use this command in administrator exec mode if you do not want an entry to carry across a system reset.

To remove circuits from multicast groups, enter the following command in administrator exec mode:

ip igmp leave-group circuit {*slot/port* {*vpi vci* | *hdlc-channel dlci*} | **lac** *vcn* | **lns** *vcn* | **pppoe** *cm-slot-session-id*} *multicast-IP-address*| **all**

or enter the following command in context configuration mode:

no ip igmp join-group circuit {*slot/port* {*vpi vci* | *hdlc-channel dlci*} | **lac** *vcn* | **lns** *vcn* | **pppoe** *cm-slot-session-id*} *multicast-IP-address*

Caution In administrator exec mode, the **ip igmp leave-group** command drops the specified circuits from the multicast group immediately. In context configuration mode, the **no ip igmp join-group** command ensures that there are no current members in the multicast group before dropping the circuits.

Limit the Number of Groups a Subscriber Can Join

By default, subscribers can join an unlimited number of multicast groups. To set a limit, enter the following command in subscriber configuration mode:

ip multicast max-groups *max-count*

Use Access Control Lists to Filter IGMP Query Types

To allow or deny IGMP query types on an interface or a subscriber basis, enter the following command in access control list configuration mode:

{**permit** | **deny**} **igmp** {*source source-wildcard* | **any** | **host** *source*} {*destination destination-wildcard* | **any** | **host** *destination*} [*igmp-type*]

For details on this command, see the "IP Access Control List Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

Modify IGMP Interface Parameters

You can modify IGMP parameters on a per-interface basis.

Enter IGMP Interface Configuration Mode

To enter IGMP interface configuration mode, enter the following command in interface configuration mode:

ip igmp mode

Note All of the commands in the following sections are configured in IGMP configuration mode.

Modify the IGMP Version

To modify the IGMP version (1 or 2), enter the following command:

def-version {**1** | **2**}

Modify Query Intervals

You can modify intervals between IGMP queries, group-specific queries, and queries sent when a circuit comes up.

Between IGMP Queries

To modify the amount of time that passes between IGMP queries, enter the following command:

query-interval timer *interval*

Between Group-Specific Queries

To modify the amount of time between IGMP group-specific query messages, enter the following command:

last-member-query-interval {**count** *packets* [**timer** *interval*] | **timer** *interval*}

Between Queries Sent By the SMS Device When a Circuit Comes Up

To modify the length of time between queries sent by the SMS when a circuit comes up, enter the following command:

startup-query-interval {**count** *packets* [**timer** *interval*] | **timer** *interval*}

Modify the Maximum Time Allowed for a Host to Respond to a Query

To modify the maximum time allowed for a host to send a response to an IGMP query, enter the following command:

query-response-interval timer *interval*

Modify the Expected Packet Loss Value

To modify the expected packet loss value, enter the following command:

robustness *value*

Modify the Unsolicited Report Interval

To modify the interval between the initial unsolicited IGMPv2 membership reports sent by the SMS device to the IP multicast router, enter the following command:

unsolicited-report-interval timer *interval*

Modify the Version 1 Router Interval

To modify the period of time that the SMS device must wait after hearing an IGMPv1 query before sending an IGMPv2 message, enter the following command:

version1-router-interval timer *interval*

Show IGMP Proxy Statistics

To view IGMP proxy interface parameters, circuit specifications, subscriber names and more, enter the following command in operator exec mode:

show ip igmp [**circuit** {*slot/port* {*vpi vci* | *hdlc-channel dlci*}| **lac** *vcn* | **lns** *vcn* |
pppoe *cm-slot-session-id*} *multicast-IP-address* | **group** [*multicast-IP-address* [**verbose**]] |
interface *if-name* [**verbose**] | **params** [**interface** *if-name*] | **subscriber** [**name** *sub-name*]]

Enable the Logging of IGMP Debug Messages

To troubleshoot IGMP proxy, you can view and save information on multicast group formations and deletions, IGMP reports and queries, and more. To enable the logging of IGMP debug messages, enter the following command in administrator exec mode:

debug ip igmp

Configuration Examples

The following examples provide a snapshot of IGMP proxy configuration tasks.

1. Enable IGMP proxy for the local context:

```
[local]RedBack(config-ctx)#ip multicast-routing
```

2. Enable receive permissions in the default subscriber record. (By default, the privilege of sending or receiving multicast traffic is denied.)

```
[local]RedBack(config-ctx)#subscriber default
[local]RedBack(config-sub)#ip multicast receive permit
```

3. Enable send permissions for a specific subscriber in the local context. Due to the default subscriber permission set in step 2, subscriber `mike` can receive multicast traffic. With the following command, `mike` can also send multicast traffic:

```
[local]RedBack(config-ctx)#subscriber name mike
[local]RedBack(config-sub)#ip address 192.34.35.2 255.255.255.0
[local]RedBack(config-sub)#ip multicast send permit
```

4. Enable IGMP proxy on an interface in the local context where IGMP proxy is needed. (The IP address of the subscriber must fall within the range of the interface.)

```
[local]RedBack(config-ctx)#interface atm41
[local]RedBack(config-if)#ip address 192.34.35.1 255.255.255.0
[local]RedBack(config-if)#ip arp arpa
[local]RedBack(config-if)#ip igmp
```

5. Bind subscriber `mike` to a circuit in the local context:

```
[local]RedBack(config)#port atm 4/1
[local]RedBack(config-port)#atm pvc 1 40 profile fast encapsulation route1483
[local]RedBack(config-pvc)#bind subscriber mike@local
```

6. Designate the multicast router interface—this is an interface on the subnet to which the true multicast router is attached. There can only be one multicast router interface for a given context. In this case, the designated interface is `backbone`.

```
[local]RedBack(config-ctx)#router igmp-proxy
[local]RedBack(config-router-igmp)#router-igmp-interface backbone
```

7. Bind the multicast router interface `backbone` to only one circuit in the local context:

```
[local]RedBack(config)#port atm 5/0
[local]RedBack(config-port)#atm pvc 1 50 profile fast encapsulation route1483
[local]RedBack(config-pvc)#bind interface backbone local
```

Part 10

Access Control Lists

Configuring IP Access Control Lists

This chapter provides an overview of IP access control lists and describes the tasks involved in configuring them through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "IP Access Control List Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

IP access control lists are lists of packet filters used to control whether packets are forwarded, dropped, or redirected. The system examines each packet to determine whether to forward, drop, or redirect the packet, based on the criteria specified in the access control list associated with a given circuit or interface.

The AOS access control lists support access control at the transport, network, and data-link layers of the seven-layer Open System Interconnection (OSI) reference model.

The following features apply to all IP access control lists:

- A list can contain multiple entries and the order is significant. Each entry is processed in the order it appears in the configuration file. As soon as an entry matches, the corresponding action is taken and no further processing takes place.
- Each list has an implicit **deny any** statement at the end. If a packet does not match any filter statement in the list, it is dropped.
- All packets dropped as a result of an access control list are counted as dropped packets.
- Filter types include IP (basic and extended), Internet Control Message Protocol (ICMP), Transmission Control Protocol (TCP), and User Datagram Protocol (UDP).
- Both inbound and outbound filters are supported.
- All access lists are defined within a context.

- You can apply access groups directly to an interface or indirectly to a circuit through a subscriber record. If you configure an access control list for both a circuit and the interface to which it is bound, a packet traverses both filters in order. That is, for input filters—circuit then interface, and for outbound filters—interface then circuit.
- If you configure an interface or circuit with a nonexistent access control list, the default behavior is for the list to be treated as an implicit "permit any" access control list. Packets are forwarded with no filtering. However, using the **access-list undefined** command, you can specify that a nonexistent access control list be treated with "deny any" functionality. Either way, once the access control list is defined, the list's definition is then applied to the subscriber or interface.
- An empty access control list is treated with an implicit "deny any" functionality.
- In addition to the **permit** and **deny** commands, the **redirect** command is a supported filter action. Packets matching the filter are always sent to the specified interface (and next-hop address for media that can use Address Resolution Protocol [ARP]) irrespective of any forwarding table information.

Multiple administrators should not simultaneously edit the configuration file. This is especially the case with access lists. Doing this can have unpredictable results.

Once in access control list configuration mode, each command creates a statement in the access control list. When the access control list is applied (to a context, subscriber, interface, or to the Subscriber Management System [SMS] device itself), the action performed by each statement is one of the following:

- A **permit** statement causes any packet matching the criteria to be accepted.
- A **deny** statement causes any packet matching the criteria to be dropped.
- A **redirect** statement causes any packet matching the criteria to be forwarded to the specified next-hop through the specified interface, regardless of the contents of the forwarding table.

All access control lists have an implicit **deny any** command at the end. A packet that does not match the criteria of the first statement is subjected to the criteria of the second statement, and so on, until the end of the access control list is reached, at which point, the packet is dropped.

When used without a prefix, each **deny**, **permit**, or **redirect** command creates a new statement in the access control list. When used with the **before**, **after**, or **no** prefix, each command identifies an existing statement in the access control list.

The **before** and **after** prefixes are positioning prefixes. They indicate where in the access control list you want to insert additional statements. For example, if your access control list already consists of five statements and you want to insert more statements between the third and fourth, you would first use the **after** prefix, specifying the third statement (or the **before** prefix, specifying the fourth statement). The next new statement you create is then inserted between the original third and fourth statements. The next new statement is inserted after that one, and so on, until you provide a different positioning command. Without the instruction provided by a positioning command, each new statement you create is appended after the statement you created before it. Without any positioning commands at all, each new statement is appended to the end of the access control list.

The **no** form of an access control list configuration command identifies and removes an individual statement from the access control list. To delete an entire access control list, enter context configuration mode, and use the **no** form of the **ip access-list** command, naming the access list to be deleted. To disassociate an access list from the context, interface, or subscriber to which it was applied, enter the appropriate mode, and use the **no** form of the **ip access-group** command.

Administrative Access Control Lists

Administrative access control lists are IP access control lists applied to a context rather than to an interface or a subscriber circuit within a context. This kind of access control list allows you to control administrative access to an SMS device through the data path of any of the AOS contexts. Administrative access control lists can be used to permit and deny network access to an SMS device for more than just Telnet; they can be used to control access for Simple Network Management Protocol (SNMP), HTTP, and any other higher-layer protocol.

In effect, an administrative access control list is applied between the IP layer and the transport layer. That is, it is applied to packets that are either generated by or received by the AOS and are specifically not applied to transit traffic. The benefit of access control lists in such a position is that they are applied only at the point at which an IP packet is determined to be passed to the higher-layer protocol in the SMS device; they are not applied within the forwarding path.

Because the AOS model is that of *n* multihomed hosts, where *n* is the number of contexts, and because IP addresses are context-specific, administrative access control lists are per-context entities. An administrative access control list is enabled by applying an IP access control list to the context itself, rather than to an interface or a subscriber record.

Caution Be careful when applying administrative access control lists to one or more contexts; you can disable access to higher-layer protocols that are essential to the operation of a network, such as Internet Control Message Protocol (ICMP), Remote Authentication Dial-In User Service (RADIUS), or Layer 2 Tunneling Protocol (L2TP).

Although it is tempting to disable all higher-layer protocol access to an SMS device through the data path of certain contexts, consider the following before doing so:

- ICMP should be enabled under almost all circumstances.
- RADIUS authentication or accounting does not function unless UDP ports 1812 and 1813 are enabled, respectively (or the alternative, AOS-selectable ports being used for RADIUS).
- L2TP does not function unless UDP port 1701 is enabled.
- Dynamic Host Configuration Protocol (DHCP) does not function unless UDP ports 67 and 68 are enabled.

Due to the complexity of configuring administrative access control lists, we recommend enabling all higher-layer protocols and specifically disabling the ones you want to deny, such as Telnet, SNMP, or HTTP.

Reflexive Access Control Lists

Reflexive access control lists are applied only to subscribers. Reflexive access control lists are different in that they have the ability to watch both directions of traffic and apply filtering dynamically, based on the configured criteria. A reflexive access control list defines the traffic to be watched in one direction to determine if and how traffic in the opposite direction should be allowed to pass.

If the traffic being watched in one direction matches the configured criteria, a corresponding access control list is dynamically installed for the return trip. This access control list exists only for the duration of the session that matched the configured criteria. When no activity in the original direction has occurred for the amount of time specified in the **ip reflexive timeout** command, the connection in the return direction is dropped. This is also referred to as a stateful firewall, meaning that the firewall is dynamically defined for the period of an individual session.

This type of access control list is typically used to protect subscribers from unauthorized access when they initiate File Transfer Protocol (FTP) or Trivial File Transfer Protocol (TFTP) connections.

There is a limit of 20 simultaneous reflexive connections per subscriber.

Dynamic Redirects

Dynamic redirects are created when traffic is detected that matches criteria specified in the **watch** construct in a **redirect** command. Redirect commands that contain a **watch** construct are referred to as redirect/watch access control list entries. The **redirect** instructions do not take effect until traffic that matches the criteria specified in the **watch** construct is detected. At that time, traffic is redirected according to the instructions in the **redirect** command until the time period specified in the **ip dynamic-acl timeout** command has elapsed.

The **watch** construct can direct the AOS to watch for Internet Protocol (IP), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Internet Group Management Protocol (IGMP), or Internet Control Message Protocol (ICMP) traffic, and can be added to the end of any of the **redirect** commands supported by the AOS.

Although IP access control lists can be applied to interfaces, contexts, and subscribers, redirect/watch entries are only applicable when applied to subscribers.

Configuration Tasks

To configure an IP access control list, perform the tasks in the following sections:

- Map Out the Goals of the List
- Create the IP Access Control List
- Create the Statements in the List
- Display the Completed List
- Apply the IP Access Control List
- Set the Reflexive Timeout Period
- Set the Reflexive Timeout Period

- Specify the Handling of Undefined Access Control Lists
- Enable Access Control List Downloading
- Display Active Reflexive Access Control Lists
- Display Active Dynamic Redirects

Map Out the Goals of the List

Before you begin entering the commands that create and configure the IP access control list, carefully consider what you want to achieve with the list; whether it is better to deny specific accesses and permit all others or to permit specific accesses and deny all others.

Create the IP Access Control List

To create an IP access control list and enter access control list configuration mode, enter the following command in context configuration mode:

ip access-list *list-name*

Entering this command puts you into access control list configuration mode, where you can enter the individual statements that make up the access control list.

Create the Statements in the List

The following access control list configuration mode commands are available to you for building an IP access control list:

{**permit** | **deny**} {*source* [*source-wildcard*] | **any** | **host** *source*}

{**permit** | **deny**} **icmp** {*source source-wildcard* | **any** | **host** *source*} {*destination destination-wildcard* | **any** | **host** *destination*} [*icmp-type* [*icmp-code*]]

{**permit** | **deny**} **igmp** {*source source-wildcard* | **any** | **host** *source*} {*destination destination-wildcard* | **any** | **host** *destination*} [*igmp-type*]

{**permit** | **deny**} **ip** {*source source-wildcard* | **any** | **host** *source*} {*destination destination-wildcard* | **any** | *host destination*}

{**permit** | **deny**} {**tcp** | **udp**} {*source source-wildcard* | **any** | **host** *source*} [**eq** *port* | **gt** *port* | **lt** *port* | **neq** ***port*** | **range** *port endport*] {*destination destination-wildcard* | **any** | **host** *destination*} [**eq** *port* | **gt** *port* | **lt** port | **neq** *port* | **range** *port endport*] [**established**]

redirect *interface next-hop* {*source* [*source-wildcard*] | **any** | **host** *source*} [**watch** *construct*]

redirect *interface next-hop* **icmp** {*source source-wildcard* | **any** | **host** *source*} {*destination destination-wildcard* | **any** | **host** *destination*} [*icmp-type* [*icmp-code*]]

redirect *interface next-hop* **ip** {*source source-wildcard* | **any** | **host** *source*} {*destination destination-wildcard* | **any** | **host** *destination*}

redirect *interface next-hop* {**tcp** | **udp**} {*source source-wildcard* | **any** | **host** *source*} [**eq** *port* | **gt** *port* | **lt** *port* | **neq** *port* | *range port endport*] {*destination destination-wildcard* | **any** | **host** *destination*} [**eq** *port* | **gt** *port* | **lt** *port* | **neq** *port* | **range** *port endport*] [**established**]

reflexive {**ftp** | **tftp**} {*source source-wildcard* | **any** | **host** *source*} {*destination destination-wildcard* | **any** | **host** *destination*}

reflexive {**tcp** | **udp**} {*source* [*source-wildcard*] | **any** | **host** *source*} [**eq** {*port* | **learned**} | **gt** {*port* | **learned**} | **lt** {*port* | **learned**} | **neq** {*port* | **learned**} | **range** *port endport*] {*destination destination-wildcard* | **any** | **host** *destination*} [**eq** {*port* | **learned**} | **gt** {*port* | **learned**} | **lt** {*port* | **learned**} | **neq** {*port* | **learned**} | **range** *port endport*] [**watch** {**dest-port eq** *port* | **source-port eq** *port* | **dest-port eq** *port* **source-port eq** *port*}]

To create redirect/watch entries in an IP access control list, use any of the following syntax structures for the **watch** *construct* construct at the end of a **redirect** command:

watch {*source source-wildcard* | **any** | **host** *source*}

watch ip {*source source-wildcard* | **any** | **host** *source*} {*destination destination-wildcard* | **any** | *host destination*}

watch {**tcp** | **udp**} {*source source-wildcard* | **any** | **host** *source*} [**eq** *port* | **gt** *port* | **lt** *port* | **neq** ***port*** | **range** *port endport*] {*destination destination-wildcard* | **any** | **host** *destination*} [**eq** *port* | **gt** *port* | **lt** port | **neq** *port* | **range** *port endport*] [**established**]

watch igmp {*source source-wildcard* | **any** | **host** *source*} {*destination destination-wildcard* | **any** | **host** *destination*} [*igmp-type*]

watch icmp {*source source-wildcard* | **any** | **host** *source*} {*destination destination-wildcard* | **any** | **host** *destination*} [*icmp-type* [*icmp-code*]]

Enter these IP access control list commands as many times as necessary to build an access control list that filters packets appropriately on the basis of some combination of IP (basic and extended), ICMP, TCP, and UDP filter criteria.

Note Remember that there is an implicit **deny any** command at the end of every list. That means that anything that does not match a statement in the access list is denied. If, instead, you want anything that does not match a statement to be allowed, insert a **permit any** command as the last explicit statement in your list.

Display the Completed List

It can be useful to see all the statements in the list to ensure that you have achieved the goals of the list. To display the access list, enter the following command in administrator exec mode:

show ip access-list [*list-name*]

The optional *list-name* argument allows you to limit the display to a particular access control list. If you omit *list-name*, the display includes all access control lists that have been configured (both bridge and IP access control lists).

For an access control list called `WebCacheACL`, the resulting display might look like this:

```
ip access-list WebCacheACL
redirect WebCacheIntf 10.0.0.2 tcp any any eq 80
permit ip any any
```

If in displaying the list, you find that you want to add a statement, return to access control list configuration mode, and use the **before** or **after** positioning prefix to indicate where in the list you want to insert an additional statement. For example, to add a statement that says `permit udp 10.2.2.0 0.0.0.255 host 10.3.3.41 eq 1813` before the statement that says `permit ip any any`, enter the following commands:

before permit ip any any

permit udp 10.2.2.0 0.0.0.255 host 10.3.3.41 eq 1813

Displaying the list again shows the added statement in the correct position:

show ip access-list WebCacheACL

```
ip access-list WebCacheACL
redirect WebCacheIntf 10.0.0.2 tcp any any eq 80
permit udp 10.2.2.0.0.0.0.255 host 10.3.3.41 eq 1813
permit ip any any
```

You can also use the **show ip access-list** command without the optional *list-name* argument to show summary information for all the IP access lists in the context. In that case, any access list that was downloaded from a RADIUS server is indicated by the word `downloaded` in parentheses following the name of the list:

show ip access-list

```
IP access list 101
 redirect radius 155.53.197.100 tcp any any eq 80
 permit tcp any any
 permit ip any any

IP access list 201 (downloaded)
 permit udp any host 10.10.20.30
 deny tcp any any
```

Apply the IP Access Control List

Once the access control list is created and its conditions have been set, you can use the **ip access-group** command to apply the list to an interface, a subscriber (or default subscriber), or a context, depending on the configuration mode in which you enter the command. Applying the list to a context makes it an administrative access control list. See the "Administrative Access Control Lists" section for more information about this type of access control.

Note Although you can apply IP access control lists to interfaces, contexts, and subscribers, redirect/watch entries are only functional when you apply them to subscribers.

Enter the following command in interface, subscriber, or context configuration mode:

ip access-group *group-name* {**in** | **out**}

The **in** and **out** keywords specify whether you want the access control list applied to incoming or outgoing traffic. The named access list can be locally configured or it can be defined remotely via RADIUS. Access lists that are defined remotely and downloaded via RADIUS are called downloadable access control lists. These lists can only be downloaded if the feature is enabled using the **aaa authorization access-list radius** command in context configuration mode.

Set the Reflexive Timeout Period

To set the amount of time a reflexive access control list is kept installed after traffic has stopped flowing from the destination direction, enter the following command in global configuration mode:

ip reflexive timeout *seconds*

The *seconds* argument can have a value from 0 to 600. This timeout period is then applied to all reflexive connections on the SMS device.

Set Dynamic Access Control List Timeout Period

To set the amount of time a dynamic redirect is kept installed after traffic has stopped flowing from the destination direction, enter the following command in global configuration mode:

ip dynamic-acl timeout *seconds*

The *seconds* argument can have a value from 0 to 600. This timeout period is then applied to all connections created as a result of a dynamic redirect access control list entry on the SMS device.

Specify the Handling of Undefined Access Control Lists

In each context, you can specify how packets are to be handled (forwarded or dropped) when an undefined access control list is applied to a subscriber or to an interface.

This feature is helpful in cases where an access control list that has not yet been configured is applied to an interface or subscriber, or in cases where an incorrectly named access control list is applied. You can determine whether traffic intended for the interface or subscriber in such an instance is forwarded or dropped. Once a defined access control list is applied to the interface or subscriber, traffic can be transmitted according to the parameters of that access control list.

To specify whether packets are to be forwarded or dropped in cases where an undefined access control list is applied, perform the following command in context configuration mode:

access-list undefined {**permit-all** | **deny-all**}

The **permit-all** keyword causes all traffic to be forwarded and the **deny-all** keyword causes all traffic to be dropped.

Note In the case of an access control list that is downloaded from a RADIUS server, the behavior that you establish with the **access-list undefined** command is applied to subscribers in the (usually) brief interim between authorization and downloading of the list, and between clearing a downloaded access list (with the **clear access-list** command) and downloading a new version.

Enable Access Control List Downloading

You can configure IP access control lists locally, or you can configure them remotely and download them via RADIUS. You must enable the ability to download access lists on a per-context basis. When this feature is enabled, if a requested access list does not appear in the local configuration, the AOS looks for the list in the RADIUS database and downloads it from there. The list stays resident as long as there are subscribers referencing it. When there are no more subscribers referencing a list, the list is deleted from the system.

To enable access lists in a context, enter the following command in context configuration mode:

aaa authorization access-list radius

The following is an example of an access control list named `general` that is defined remotely using RADIUS attributes:

```
general
Password = “Redback”
Service-Type = Access-Control-List
Redback:ACL-Definition = “redirect to_subs 10.1.1.1 any”,
Redback:ACL-Definition = “deny icmp 12.1.1.1 12.3.4.23 any 100 200”,
Redback:ACL-Definition = “permit ip any any”,
Redback:ACL-Definition = “deny ip 121.24.234.12 121.3.34.129 23.4.34.0 25.23.56.12”
```

The ACL-Definition vendor-specific attribute (VSA) uses the same syntax as the AOS IP access control list commands (see the “IP Access Control List Commands” chapter in the *Access Operating System (AOS) Command Reference* publication).

To dereference a downloaded access control list from bound subscribers, and download the access control list again from the RADIUS server (such as when the list has been updated), enter the following command in administrator exec mode:

clear access-list *context* [*list-name*]

The *context* argument is the name of the context in which the access list is defined and the *list-name* argument is the name of the access list. You can also clear all downloaded access lists in the context by omitting the optional *list-name* argument.

Display Active Reflexive Access Control Lists

A reflexive entry in an access control list is considered active when traffic matches the criteria defined by the **reflexive** command. Until there is traffic that matches those criteria, the reflexive entry is considered passive. To display reflexive entries that are currently active for a specific subscriber, enter the following command in administrator exec mode:

show ip reflexive-acl subscriber *sub-name*

The *sub-name* argument is the name of the subscriber whose information you want to view. The name must be in the default structured username format (*name*@context) or other configured custom format. The output of this command shows any configured reflexive entries that are currently active along with the number of matched packets, if any.

Display Active Dynamic Redirects

A redirect/watch entry in an access control list is considered active when traffic matches the criteria defined by the **watch** *construct* in the **redirect** command. Until there is traffic that matches those criteria, the redirect/watch entry is considered passive. To display redirect/watch entries that are currently active for a specific subscriber, enter the following command in administrator exec mode:

show ip dynamic-acl subscriber *sub-name*

The *sub-name* argument is the name of the subscriber whose information you want to view. The name must be in the default structured username format (*name*@context) or other configured custom format. The output of this command shows any configured redirect/watch entries that are currently active along with the number of matched packets, if any.

Configuration Examples

Basic IP Access Control List Example

In the following example, an access list is created whose purpose is to redirect all web traffic (TCP port 80) to a web cache that has an IP address of `10.0.0.2`.

First, we create an interface named `WebCacheIntf` through which the web cache is attached to the SMS device:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#interface WebCacheIntf
[local]RedBack(config-if)#ip address 10.0.0.1 255.255.255.0
```

Next, we create an interface named `Downstream` to which the subscriber circuits is bound and to which we apply the access list to input packets:

```
[local]RedBack(config-ctx)#interface Downstream
[local]RedBack(config-if)#ip address 176.16.0.1 255.255.255.0
[local]RedBack(config-if)#ip access-group WebCacheACL in
```

Finally, we create the access list itself. The **permit ip any any** command is required. Without it, the implicit **deny any** command, present at the end of every access list, would cause all non-web traffic to be dropped in this example, which is not the desired behavior:

```
[local]RedBack(config-ctx)#access-list undefined deny-all
[local]RedBack(config-ctx)#ip access-list WebCacheACL
[local]RedBack(config-acl)#redirect WebCacheIntf 10.0.0.2 tcp any any eq 80
[local]RedBack(config-acl)#permit ip any any
```

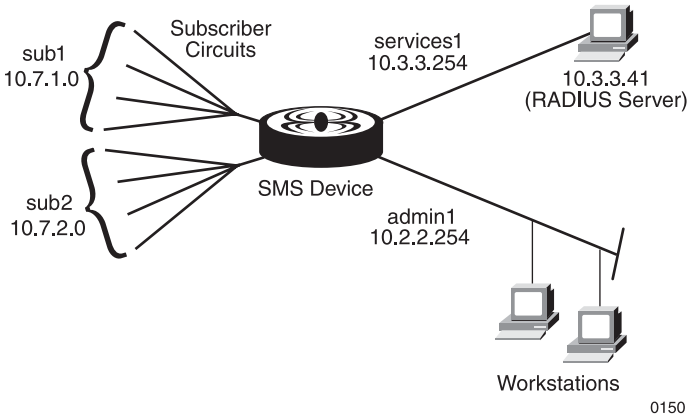
If an administrator configures a subscriber, intends to apply the access control list just created, and types the name of the access control list incorrectly, the use of the **access-list undefined deny-all** command would prevent any packets from going through until the error is corrected.

Advanced IP Access Control List Examples

In each of the examples in this section, an access control list is defined to control access to a RADIUS server attached to an Ethernet segment connected to the SMS device.

Figure 37-1 is the basis for all the examples in this section.

Figure 37-1 Access Control List Example Network



The following assumptions are common to all of the following examples:

- The administrative subnet is 10.2.2.0/24—bound to interface `admin1`.
- The services subnet is 10.3.3.0/24—bound to interface `services1`.
- The subscribers are assigned addresses in the 10.7.1.0/24 and 10.7.2.0/24 subnets—bound to interfaces `sub1` and `sub2`.
- The RADIUS server's address is 10.3.3.41.
- The interface address of `admin1` is 10.2.2.254.
- The interface address of `services1` is 10.3.3.254.
- RADIUS client/server communication is carried out on UDP sockets 1812 and 1813.
- The web server IP address is 10.3.3.51.

Advanced Example 1

In this example, we create an access list whose purpose is to prevent any user coming in on a subscriber circuit from having access to a RADIUS server on an Ethernet segment off the SMS device.

Purpose

The purpose of the access control list in this example is to:

- Prevent all packets not originating on the administrative subnet, or from the SMS device, from being forwarded to the services subnet.
- Allow access for UDP packets originating on the administrative subnet with a destination IP of 10.3.3.41 (the RADIUS server) and to socket 1812 or 1813.

Definition

The access control list in this example is defined as follows:

- Interface on which the access control list is placed: services1
- Direction: outgoing
- IP destination address of outgoing packets: 10.3.3.41
- Destination ports: 1812 and 1813
- IP source address of outgoing packets: anyone on subnet 10.2.2.0
- Source ports: any
- Packet type: UDP
- Default action: deny any

Syntax

The syntax to implement the access control list in this example is as follows:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#interface services1
[local]RedBack(config-if)#ip address 10.3.3.254 255.255.255.0
[local]RedBack(config-if)#ip access-group acl1 out
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#ip access-list acl1
[local]RedBack(config-acl)#permit udp 10.2.2.0 0.0.0.255 10.3.3.41 0.0.0.0 eq1812
[local]RedBack(config-acl)#permit udp 10.2.2.0 0.0.0.255 host 10.3.3.41 eq 1813
[local]RedBack(config-acl)#permit udp 10.3.3.0 0.0.0.255 host 10.3.3.41 eq 1812
[local]RedBack(config-acl)#permit udp 10.3.3.0 0.0.0.255 host 10.3.3.41 eq 1813
[local]RedBack(config-acl)#exit
```

Result

All UDP packets from any host on the administrative subnet `admin1` being sent to port 1812 or 1813 on RADIUS server 10.3.3.41 are forwarded. All other packets are dropped.

Advanced Example 2

In this case, the access control list from Example 1 is made more restrictive. An incoming filter is placed on `services1` to allow only packets to addresses on `admin1` and to port 1812 or 1813.

Purpose

The purpose of the access control list in this example is to:

- Prevent all packets not originating on the administrative subnet or the SMS device from being forwarded to the services subnet.
- Allow access for all UDP packets originating on the admin1 subnet or on the SMS device with a destination IP of 10.3.3.41 (the RADIUS server) and to socket 1812 or 1813.

- Allow administrators on the services subnet to access TCP-based services on other subnets.
- Allow the RADIUS server on the `services1` subnet to communicate with RADIUS clients on the `admin1` subnet.

Definition

The access control list in this example is defined as follows:

- Interface on which the access control list is placed: services1
- Direction: outgoing
- IP destination address of outgoing packets: 10.3.3.41
- Destination ports: 1812 and 1813
- IP source address of outgoing packets: anyone on subnet 10.2.2.0
- Source ports: any
- Packet type: UDP
- Default action: deny any

Syntax

The syntax to implement the access control list in this example is as follows:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#interface services1
[local]RedBack(config-if)#ip address 10.3.3.254 255.255.255.0
[local]RedBack(config-if)#ip access-group acl2a out
[local]RedBack(config-if)#ip access-group acl2b in
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#ip access-list acl2a
[local]RedBack(config-acl)#permit udp 10.2.2.0 0.0.0.255 10.3.3.41 0.0.0.0 eq1812
[local]RedBack(config-acl)#permit udp 10.2.2.0 0.0.0.255 host 10.3.3.41 eq1813
[local]RedBack(config-acl)#permit udp 10.3.3.0 0.0.0.255 host 10.3.3.41 eq1812
[local]RedBack(config-acl)#permit udp 10.3.3.0 0.0.0.255 host 10.3.3.41 eq1813
[local]RedBack(config-acl)#permit tcp any 10.3.3.0 0.0.0.255 established
[local]RedBack(config-acl)#exit
[local]RedBack(config-ctx)#ip access-list acl2b
[local]RedBack(config-acl)#permit tcp 10.3.3.0 0.0.0.255 any
[local]RedBack(config-acl)#permit udp 10.3.3.41 0.0.0.0 10.2.2.0 0.0.0.255 eq 1812
[local]RedBack(config-acl)#permit udp host 10.3.3.41 10.2.2.0 0.0.0.255 eq 1813
[local]RedBack(config-acl)#permit udp host 10.3.3.41 10.3.3.0 0.0.0.255 eq 1812
[local]RedBack(config-acl)#permit udp host 10.3.3.41 10.3.3.0 0.0.0.255 eq 1813
[local]RedBack(config-acl)#exit
```

Result

All UDP packets from any host on the administrative subnet `admin1` being sent to ports 1812 or 1813 on RADIUS server 10.3.3.41 are forwarded. All TCP packets from the `admin1` subnet are forwarded to the `services1` subnet, except for packets that do not have the SYN bit and ACK set; that is, no TCP connections can be initiated from outside the `services1` subnet. All TCP packets from the `services1` subnet are forwarded to any location.

Advanced Example 3

Purpose

The purpose of the access control list in this example is to:

- Prevent all packets not originating on the `admin1` subnet from being forwarded to the `services1` subnet.
- Allow access for all IP packets originating on the `admin1` subnet only to the RADIUS server.

Definition

The access control list in this example is defined as follows:

- Type of application: outgoing
- Interface on which the access control list is placed: services1
- IP destination address of outgoing packets: 10.3.3.41
- Destination port: any
- IP source address of outgoing packets: anyone on subnet 10.2.2.0
- Source port: any
- Packet type: IP (all IP packets)
- Default action: deny any

Syntax

The syntax to implement the access control list in this example is as follows:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#interface services1
[local]RedBack(config-if)#ip address 10.3.3.254 255.255.255.0
[local]RedBack(config-if)#ip access-group acl3 out
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#ip access-list acl3
[local]RedBack(config-acl)#permit ip 10.2.2.0 0.0.0.255 10.3.3.41 0.0.0.0
[local]RedBack(config-acl)#exit
```

Result

All IP packets from any host on the administrative subnet `admin1` are forwarded to the host running the RADIUS server 10.3.3.41. All other packets are dropped.

Advanced Example 4

Purpose

The purpose of the access control list in this example is to:

- Provide access to any machine on the services subnet for any packets originating on the `admin1` subnet.
- Reject all other packets.

Definition

The access control list in this example is defined as follows:

- Interface on which access control list is placed: services1
- Direction: outgoing
- IP destination address of outgoing packets: any host on subnet 10.3.3.0
- Destination port: any
- IP source address of outgoing packets: anyone on subnet 10.2.2.0
- Source port: any
- Packet type: IP (all IP packets)
- Default action: deny any

Syntax

The syntax to implement the access control list in this example is as follows:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#interface services1
[local]RedBack(config-if)#ip address 10.3.3.254 255.255.255.0
[local]RedBack(config-if)#ip access-group acl4 out
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#ip access-list acl4
[local]RedBack(config-acl)#permit ip 10.2.2.0 0.0.0.255 10.3.3.0 0.0.0.255
[local]RedBack(config-acl)#exit
```

Result

All IP packets from any host on the `admin1` subnet are forwarded to any host on the `services1` subnet. All other packets are dropped.

Note Packets that attempt to spoof an address on the `admin1` subnet are dropped if they come in on any interface other than `admin1`.

Advanced Example 5

Purpose

The purpose of the access control list in this example is to:

- Provide access to any machine on the `services1` subnet for any packets originating on the `admin1` subnet.
- Provide access to all machines to a web server on the `services1` subnet with IP address 10.3.3.51.

Definition

The access control list in this example is defined as follows:

- Interface on which the access control list is placed: **services1**
- Direction: outgoing
- IP destination address of outgoing packets:

  if source IP is on the admin1 subnet—any

  for everyone else—only 10.3.3.51
- Destination port:

  if source IP is on the admin1 subnet—any

  for everyone else—port 80
- IP source address of outgoing packets:

  if destination IP address is web server—any

  if destination IP address is anything other than 10.3.3.51 source has

  to be from subnet 10.2.2.0
- Source port: any
- Packet type:

  IP—if source IP address is on the admin1 subnet

  TCP only if source address is anything else

Syntax

The syntax to implement the access control list in this example is as follows:

```
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#interface services1
[local]RedBack(config-if)#ip address 10.3.3.254 255.255.255.0
[local]RedBack(config-if)#ip access-group acl5 out
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#ip access-list acl5
[local]RedBack(config-acl)#permit ip 10.2.2.0 0.0.0.255 10.3.3.0 0.0.0.255
[local]RedBack(config-acl)#permit tcp any host 10.3.3.51 eq 80
[local]RedBack(config-acl)#exit
```

Result

All HTTP packets from any source are forwarded to the web server on the `services1` subnet. All IP packets from the `admin1` subnet are forwarded to any host on the `services1` subnet.

Administrative Access Control List Examples

The examples in this section show applying IP access control lists to contexts to create administrative access control lists.

Administrative Access Control List Example 1

In the following example, we design and apply an administrative access control list to a context named `isp.net` in which we assume that all management is performed by hosts whose IP address prefixes are in the `10.0.0.0/24` network. In other words, we prohibit administrative access by hosts with any other address-prefixes.

We assume that the SNMP management station, and the hosts from which administrators access the box through Telnet and HTTP, along with the RADIUS servers, and DHCP servers, are located on the `10.0.0.0/24` network. In addition, we enable all hosts to send/receive ICMP packets to the SMS device, under the assumption that customers should be allowed to ping their router, should be informed when destinations they seek are unreachable, should be able to perform traceroutes, and so on.

First, we create the access control list and then apply it to the `isp.net` context. The first item in the list enables ICMP; the second item allows all packets from the `10.0.0.0/24` network to be accepted by AOS for any higher-layer protocol. The implicit **deny any** command at the end of the list denies all other access.

```
[local]RedBack(config)#context isp.net
[local]RedBack(config-ctx)#ip access-list IspAdminAcl
[local]RedBack(config-acl)#permit icmp any any
[local]RedBack(config-acl)#permit 10.0.0.0 0.0.0.255
[local]RedBack(config-acl)#exit
[local]RedBack(config-ctx)#ip access-group IspAdminAcl in
```

Note This example causes L2TP tunnels to fail, except in the unlikely event that the tunnel peers have addresses with the prefix `10.0.0.0/24`. This is an example of why we recommend constructing administrative access control lists as deny lists, rather than permit lists. The next example illustrates the simplicity and desirability of deny lists.

Administrative Access Control List Example 2

This example simply disables Telnet (TCP port 23) access to the box through all data paths of a context named `corp.com`:

```
[local]RedBack(config)#context corp.com
[local]RedBack(config-ctx)#ip access-list CorpAdminAcl
[local]RedBack(config-acl)#deny tcp any any eq 23
[local]RedBack(config-acl)#permit any
[local]RedBack(config-acl)#exit
[local]RedBack(config-ctx)#ip access-group CorpAdminAcl in
```

Note Do not forget the **permit any** command at the end of a deny list to counteract the effect of the implicit **deny any** command at the end of every access control list.

Dynamic Redirect Examples

The examples in this section show how redirect/watch entries can be used to trigger dynamic redirect action.

Dynamic Redirect Example 1

The following example redirects traffic with source address `172.20.1.2` port 80 (HTTP) and destination address `1.1.1.10` to address `5.1.1.2` (via interface `eth60`) once the subscriber (at address `1.1.1.10`) has pinged the web server (address `172.20.1.2`):

```
[local]RedBack(config-ctx)#ip access-list protect201
[local]RedBack(config-acl)#redirect eth60 5.1.1.2 tcp host 172.20.1.2 eq 80 host
1.1.1.10 watch icmp host 1.1.1.10 host 172.20.1.2
```

Dynamic Redirect Example 2

The following example shows that a redirect/watch entry can be combined with other types of access control list entries in a single access control list:

```
[local]RedBack(config-ctx)#ip access-list special
[local]RedBack(config-acl)#redirect eth6 200.1.1.2 any watch udp any host 100.1.1.254
eq 65535
[local]RedBack(config-acl)#reflexive ftp any any
[local]RedBack(config-acl)#permit icmp any any
[local]RedBack(config-acl)#deny tcp any host 200.1.1.1 eq 23
```

Configuring Bridge Access Control Lists

This chapter provides an overview of bridge access control lists and describes the tasks involved in configuring them through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "Bridge Access Control List Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

Bridge access control lists are very similar to IP access control lists in form and function except that bridge access control lists filter packets on the basis of information contained in the media access control (MAC) header. This can be some combination of source MAC address, destination MAC address, Ethertype, or link service access point (LSAP) value.

Access control lists are lists of packet filters used to control whether packets are forwarded or dropped. The system examines each packet to determine whether to forward or drop the packet based on the criteria specified in the access control list associated with a given circuit or interface.

- A list can contain multiple entries, and the order is significant. Each entry is processed in the order it appears in the configuration file. As soon as an entry matches, the corresponding action is taken and no further processing takes place.
- Each list has an implicit **deny any** statement at the end; that is, if a packet does not match any filter statement in the list, it is dropped.
- All packets dropped as a result of an access control list are counted as such.
- Both inbound and outbound filters are supported.
- All access lists are defined within a context.

- Access groups can be applied directly to an interface or indirectly to a circuit through a subscriber record. If an access control list is configured for both a circuit and the interface to which it is bound, a packet traverses both filters in order; that is, for input filters—circuit then interface, and for outbound filters—interface then circuit.
- If an interface or circuit is configured with a nonexistent access control list, the default behavior is for the list to be treated as an implicit permit all access control list. Packets are forwarded with no filtering. However, using the **access-list undefined** command, you can specify that a nonexistent access control list be treated as having deny all functionality. Either way, once the access control list is defined, the list's definition is then applied to the subscriber or interface.

Caution You should not allow multiple administrators to simultaneously edit a configuration file. This is especially the case with access control lists. Doing this can cause unpredictable results.

- An empty access control list is treated with implicit deny all functionality.

Once in access control list configuration mode, each command creates a statement in the access control list. When the access control list is applied (to a context, subscriber, interface, or to the Subscriber Management System [SMS] device itself), the action performed by each statement is one of the following:

- A **deny** statement causes any packet matching the criteria to be dropped.
- A **permit** statement causes any packet matching the criteria to be accepted.

All access control lists have an implicit **deny any** command at the end. A packet that does not match the criteria of the first statement is subjected to the criteria of the second statement, and so on, until the end of the access control list is reached, at which point, the packet is dropped.

When used without a prefix, each **deny** or **permit** command creates a new statement in the access control list. When used with the **before**, **after**, or **no** prefix, each command identifies an existing statement in the access control list.

The **before** and **after** prefixes are positioning prefixes. They indicate where in the access control list you want to insert additional statements. For example, if your access control list already consists of five statements and you want to insert more statements between the third and fourth, you would first use the **after** prefix, specifying the third statement (or the **before** prefix, specifying the fourth statement). The next new statement you create is then inserted between the original third and fourth statements. The next new statement is inserted after that one, and so on, until you provide a different positioning command. Without the instruction provided by a positioning command, each new statement you create is appended after the statement you created before it. Without any positioning commands at all, each new statement is appended to the end of the access control list.

Use the **no** form of an access control list configuration command to identify and remove an individual statement from the access control list. To delete an entire access control list, enter context configuration mode, and use the **no** form of the **bridge access-list** command, which names the access list to be deleted. To disassociate an access list from the interface or subscriber to which it was applied, enter the appropriate configuration mode, and use the **no** form of the **bridge-group** command.

Configuration Tasks

To configure a service access list, perform the tasks in the following sections:

- Map Out the Goals of the List
- Create the Bridge Access Control List
- Create the Statements in the List
- Display the Completed List
- Apply the Bridge Access Control List
- Specify the Handling of Undefined Access Control Lists

Map Out the Goals of the List

Before you begin entering the commands that create and configure the bridge access control list, determine what you want to achieve with the list. Consider whether it is better to deny specific accesses and permit all others or to permit specific accesses and deny all others.

Create the Bridge Access Control List

To create a bridge access control list and enter access control list configuration mode, enter the following command in context configuration mode:

bridge access-list *list-name*

Entering this command puts you into access control list configuration mode, where you can enter the individual statements that form the access control list.

Create the Statements in the List

The following commands are available to you for building a bridge access control list:

{**permit** | **deny**} *source* [*source-wildcard* [*destination* [*destination-wildcard*]]] [[**lsap** *lsap* [*lsap-wildcard*]] | [**type** *type* [*type-wildcard*]]]

{**permit** | **deny**} **lsap** *lsap* [*lsap-wildcard*]

{**permit | deny**} **type** *type* [*type-wildcard*]

Enter these commands as many times as needed to filter packets appropriately on the basis of some combination of source MAC address, destination MAC address, Ethertype, or LSAP value.

Note Remember that there is an implicit **deny any** command at the end of every list. That means that anything that does not match a statement in the access list is denied. If, instead, you want anything that does not match a statement to be allowed, insert a **permit any** command as the last explicit statement in your list.

Display the Completed List

It can be useful to see all the statements in the list to ensure that you have achieved the goals of the list. To display the access list, enter the following command in the administrator exec mode:

show bridge access-list [*list-name*]

The optional *list-name* argument allows you to limit the display to a particular access control list. If you omit the *list-name* argument, the display includes all access control lists that have been configured (both bridge and IP access control lists).

For a bridge access control list called `brmacl`, the resulting display might look like this:

```
Bridge access list brmac1
permit 01:00:5e:00:00:00 00:00:00:ff:ff:ff (10 matches)
permit 11:22:33:44:55:66 00:00:00:00:00:00
```

If in displaying the list, you find that you want to add a statement, return to access control list configuration mode, and use the **before** or **after** positioning prefix to indicate where in the list you want to insert an additional statement. For example, to add a statement that says `deny type 0x800 0x0ff` before a statement that says `permit 11:22:33:44:55:66 00:00:00:00:00:00`, enter the following commands:

before permit 11:22:33:44:55:66 00:00:00:00:00:00

deny type 0x800 0x0ff

Displaying the list again shows the added statement in the correct position:

show bridge access-list brmacl

```
Bridge access list brmac1
permit 01:00:5e:00:00:00 00:00:00:ff:ff:ff (10 matches)
deny type 0x00 0x0ff
permit 11:22:33:44:55:66 00:00:00:00:00:00
```

Apply the Bridge Access Control List

Once the bridge access control list is created and its conditions have been set, you can apply the list to an interface using the **bridge-group** command in interface configuration mode, or indirectly to a circuit through a subscriber record (or the subscriber default) using the **bridge-group** command in subscriber configuration mode. See the "Configuring Bridging" chapter for full instructions on using the **bridge-group** command. For purposes of applying an access control list, the important construct in the syntax of this command is **access-group** *group-name.*

Enter the following command in interface or subscriber configuration mode:

bridge-group *group-name* [**aging-time** *time* | **path-cost** *cost* | **spanning-disabled** | **trans-bpdu** | **access-group** *group-name* {**in** | **out**}]

The **access-group** *group-name* construct defines the bridge access control list you want to apply; the **in** and **out** keywords specify whether you want the access control list applied to incoming or outgoing traffic.

Specify the Handling of Undefined Access Control Lists

In each context, you can specify how packets are to be handled (forwarded or dropped) when an undefined access control list is applied to a subscriber or to an interface.

This feature is helpful in cases where an access control list that has not yet been configured is applied to an interface or subscriber, or in cases where an incorrectly named access control list is applied. You can determine whether traffic intended for the interface or subscriber in such an instance is forwarded or dropped. Once a defined access control list is applied to the interface or subscriber, traffic can be transmitted according to the parameters of that access control list.

To specify whether packets are to be forwarded or dropped in cases where an undefined access control list is applied, perform the following command in context configuration mode:

access-list undefined {**permit-all** | **deny-all**}

The **permit-all** keyword causes all traffic to be forwarded and the **deny-all** keyword causes all traffic to be dropped.

Configuration Examples

The following example shows creating an access list used to filter all IP multicast packets and then applying the list to an interface:

```
[local]RedBack(config-ctx)#bridge access-list 101
[local]RedBack(config-acl)#permit 01:00:5e:00:00:00 00:00:00:ff:ff:ff
[local]RedBack(config-acl)#exit
[local]RedBack(config-ctx)#interface service1
[local]RedBack(config-if)#bridge-group telecomm1 access-group 101 in
```

You can further qualify an access list by specifying an Ethertype of `0x800` (for the IP protocol):

```
[local]RedBack(config-ctx)#bridge access-list 102
[local]RedBack(config-acl)#permit 01:00:5e:00:00:00 00:00:00:ff:ff:ff type 0x800 0x0ff
```

The masks are specified in a similar manner to those used in the IP access lists. The 0 bits in the mask indicate bits that need to be examined and the 1 bits indicate those that are not significant.

Configuring Service Access Lists

This section describes how to create and edit service access lists that restrict subscriber access to contexts and domains on a per-circuit basis. The following sections are included:

- Characteristics and Behavior of Service Access Lists
- Configuration Tasks
- Configuration Examples

For detailed information on syntax and usage guidelines for the commands listed in the "Configuration Tasks" section, see the "Service Access List Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

Static bindings for PPP-encapsulated circuits provides a simpler method to restrict subscribers on a particular circuit to a single context. See the "Create a Static Binding for PPP-Encapsulated Circuits" section in Chapter 20, "Configuring Bindings."

Characteristics and Behavior of Service Access Lists

A service access list is a series of statements that define the criteria used to determine whether contexts, domains, and tunnels should be available to subscribers on a per-circuit basis. Use the **service access-list** command in global configuration mode to enter service access list configuration mode. This command requires the name of a new or existing service access list. All subsequent service access list configuration commands are applied to the access list you specify when you enter the mode.

Each service access list configuration command creates a statement in the access list. When the access list is applied (via either the **bind authentication** or **session-auth** command), the action performed by each statement is one of the following:

- A **permit** statement causes any service matching the criteria to be allowed.
- A **deny** statement causes any service matching the criteria to be blocked.

All service access lists have an implicit **deny any** command at the end. A service that does not match the criteria of the first statement is subjected to the criteria of the second statement, and so on, until the end of the access list is reached, at which point, the service is denied. The order of the statements in a service access list is relevant. A service is compared to the criteria defined in each statement, one by one, and the process stops as soon as a match is found. The action (permit or deny) dictated by the first matching

statement is carried out, and the service is never compared to the subsequent statements in the access list. For that reason, you may find it necessary to position a new statement in the middle of your access list, rather than appending it to the end.

When used without a prefix, each **permit** or **deny** command creates a new statement in the access list. When used with the **before**, **after**, or **no** prefix, each command identifies an existing statement in the access list.

The **before** and **after** prefixes are positioning prefixes. They indicate where in the access list you want to insert additional statements. For example, if your access list already consists of five statements and you want to insert more statements between the third and fourth, you would first use the **after** prefix, specifying the third statement (or the **before** prefix, specifying the fourth statement). The next new statement you create is then inserted between the original third and fourth statements. The next new statement is inserted after that one, and so on, until you provide a different positioning command. Without the instruction provided by a positioning command, each new statement you create is appended after the statement you created before it. Without any positioning commands at all, each new statement is appended to the end of the service access list.

Use the **no** form of a service access list configuration command to identify and remove an individual statement from the access list. To delete an entire service access list, enter global configuration mode, and use the **no** form of the **service access-list** command, naming the access list to be deleted. To disassociate a service access list from the circuit, port, or tunnel to which it was applied, enter the appropriate configuration mode (circuit, port, Layer 2 Tunneling Protocol [L2TP], or Layer 2 Forwarding [L2F]), and use the **no** form of either the **bind authentication** or **session-auth** command, naming the service list in the optional **service-group** *group-name* construct.

Configuration Tasks

To configure a service access list, perform the tasks in the following sections:

- Map Out the Goals of the List
- Create the Service Access List
- Create the Statements in the List
- Display the Completed List
- Apply the Service Access List

Map Out the Goals of the List

Before you begin entering the commands that create and configure the service access list, be sure that you are clear about what you want to achieve with the list. Remember that service access lists control access to contexts and domains (including tunnels).

Create the Service Access List

To create the service access list and enter service access list configuration mode, enter the following command in global configuration mode:

service access-list *list-name*

The *list-name* argument is the name by which the service access list is to be known. Entering this command puts you into service access list configuration mode where you can enter the individual statements that form the access list.

Create the Statements in the List

The following commands are available to you for building a service access list:

- {**permit** | **deny**} **any**
- {**permit** | **deny**} **context** *ctx-name*
- {**permit** | **deny**} **domain** *dom-name*

Enter these commands as many times as needed to permit and deny access to the various contexts (and tunnels) and domains in your system configuration. Remember that there is an implicit **deny any** command at the end of every list. That means that anything that does not match a statement in the access list is denied. If, instead, you want anything that does not match a statement to be allowed, insert a **permit any** command as the last explicit statement in your list.

Display the Completed List

It can be useful to see all the statements in the list to ensure that you have achieved the goals of the list. To display the access list, enter the following command from the administrator exec mode:

show service access-list [*list-name* [**circuits**]]

The optional *list-name* argument allows you to limit the display to a particular service access list. If you omit the *list-name* argument, the display includes all service access lists that have been configured. The optional **circuits** keyword allows you to include in the display any circuits to which the named service access list has been applied. In this example, you have not applied the list to anything yet, so the keyword is not useful.

For a service access list called `corporate`, the resulting display might look like this:

```
Service Access Control: corporate
  deny   domain Corp1
  permit context isp1
  permit context other_services
  permit any
```

If you want to add a statement to the list, return to service access list configuration mode and use the **before** or **after** positioning prefix to indicate where in the list you want to insert an additional statement. For example, to add a statement that says **deny domain services_2** before a statement that says **permit context other_services**, enter the following commands:

before permit context other_services

deny domain services_2

Displaying the list again shows the added statement in the correct position:

show service access-list corporate

```
Service Access Control: corporate
  deny   domain Corp1
  permit context isp1
  deny domain services_2
  permit context other_services
  permit any
```

Apply the Service Access List

Service access lists are applied to sessions by way of the authentication process. Two commands make this possible:

bind authentication

session-auth

The **bind authentication** command has an optional **service-group** *group-name* construct that allows you to limit the services available to the Point-to-Point Protocol (PPP)-encapsulated circuit or port to those permitted by the named service access list. The **session-auth** command has the same optional construct. In that case, it allows you to limit the services available to the PPP sessions from the L2TP or L2F peer to those permitted by the named service access list. In both cases, the result is that any attempt to terminate in a prohibited context or any attempt to be tunneled in a prohibited tunnel (a tunnel in a prohibited context) causes the authentication (and therefore, the PPP session) to fail.

If the service access list named in the **service-group** *group-name* construct does not exist, a debug message is written to the log, and all sessions are denied the ability to come up.

Configuration Examples

For the examples in this section, assume the following configuration of contexts, domains, and tunnel peers:

```
context isps
  domain isp1.net
  domain isp2.net
  domain isp3.net
  12tp-peer name isp1.net media pvc
  l2tp-peer name isp2.net media pvc
  l2tp-peer name isp3.net media pvc
```

```
context corporations
  domain corp1.com
  domain corp2.com
  domain corp3.com
  domain corp4.com
  l2tp-peer name corp1.com media pvc
  l2tp-peer name corp2.com media pvc
  l2tp-peer name corp3.com media pvc
  l2tp-peer name corp4.com media pvc

context other-services
  domain games
  domain support
  domain video
```

The following example creates a service access list called `NoCorps`, edits the list to delete a line, creates an Asynchronous Transfer Mode (ATM) circuit, and applies the list to the circuit. The goal of `NoCorps` is to prevent a session from being terminated in context `corporations`, or from being tunneled to any of the peers defined in that same context. The **show service access-list** command is then used to display the access list along with the circuits to which it has been applied:

```
[local]RedBack(config)#service access-list NoCorps
[local]RedBack(config-service)#deny context corporations
[local]RedBack(config-service)#deny context other-services
[local]RedBack(config-service)#permit any
[local]RedBack(config-service)#no deny context other-services
[local]RedBack(config-service)#exit
[local]RedBack(config)#port atm 3/0
[local]RedBack(config-port)#atm pvc 1 32 profile ubr encapsulation ppp
[local]RedBack(config-port)#bind auth pap service-group NoCorps
```

The next example creates a service access list called `CorpOnePlusISPs`. The purpose of this service access list is to allow a telecommuter of `corp1.com` to access a specific corporation's tunnel or to access any of the Internet service providers (ISPs) defined in context `isps`, but to prohibit access to any of the other corporations' tunnels:

```
[local]RedBack(config)#service access-list CorpOnePlusISPs
[local]RedBack(config-service)#permit domain corp1.com
[local]RedBack(config-service)#deny context corporations
[local]RedBack(config-service)#deny context other-services
[local]RedBack(config-service)#permit any
```

Part 11

AAA and RADIUS

Configuring AAA

This chapter provides an overview of authentication, authorization, and accounting (AAA) and describes the tasks involved in configuring AAA features through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands mentioned, see the "AAA Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Global AAA
- Context Assignment with Global AAA
- Configuring AAA Hint
- Two-Stage Accounting
- Marking a Context for Explicit Binding Only
- Enabling Access Control List Downloading
- Configuring Custom Formats for Structured Usernames

Global AAA

Global AAA is used to force all authentication requests and accounting updates, regardless of context, to a single set of Remote Authentication Dial-In User Service (RADIUS) servers. An example application is one in which a Subscriber Management System (SMS) device has several contexts configured, but only one of them (namely, local) has IP reachability to a RADIUS server.

Note Global authentication overrides any context-specific authentication commands. Context-specific authentication commands are ignored if global authentication is enabled. The AOS provides warnings if a context-specific authentication server is configured when global authentication is turned on. While the commands to enable global AAA are global configuration commands, the RADIUS servers themselves, and the load-balancing algorithm (if any) are configured in the local context.

The following commands configure global authentication on an SMS device. The global RADIUS server has an IP address of `10.0.0.1` and must be reachable through an interface defined in the local context:

```
[local]RedBack(config)#aaa authentication subscriber radius
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#radius server 10.0.0.1 key shhhhh
```

Context Assignment with Global AAA

When authenticating at a context level (rather than global AAA), a subscriber has to use a structured username of the default form *username@context* or of a configured custom form. This is required for the SMS device to determine the context in which a user should be authenticated. When using global AAA, the user can still use a structured username, but it is not necessary because all authentication takes place within the global AAA context (in other words, the local context).

When global AAA is enabled, all incoming users are bound to the local context prior to authentication. If authentication succeeds, the user is rebound to another context provided the Access-Accept packet contains the Context-Name attribute and the context name is valid. Authentication fails if the Context-Name attribute is missing or if the context name supplied is invalid.

At the context level, subscribers can be authenticated by the local database, or by a RADIUS server. To configure how subscribers are to be authenticated, enter the following command in context configuration mode:

aaa authentication subscriber {**local** [**radius**] | **radius** [**local**] | **none**}

You can use the **local** and **radius** keywords together to specify that one method of authentication is to be attempted first, followed by the other. If you enter the **local** keyword, followed by the **radius** keyword, authentication is attempted first by the local configuration. If the subscriber record cannot be found locally, authentication is attempted by the RADIUS server. If you enter the **radius** keyword, followed by the **local** keyword, authentication is attempted by the local database in the event that the RADIUS server cannot be reached.

Configuring AAA Hint

The purpose of AAA hint is to provide to the RADIUS server a preferred IP address from the SMS device's local pool. The address is sent to the RADIUS server in the Framed-IP-Address attribute of the RADIUS authentication request packet. The RADIUS server may accept the IP address sent by the SMS device or it may not. The SMS device only uses the address if the RADIUS server confirms that it is acceptable. Table 40-1 shows the actions the SMS device takes based on the possible responses from the RADIUS server to the authentication request.

Table 40-1 SMS Actions Based on RADIUS Server Response

RADIUS Response to AAA Hint	Corresponding SMS Action
RADIUS returns the IP address contained in the AAA hint	SMS device uses the address contained in the AAA hint
RADIUS returns a different address from the one contained in the AAA hint	SMS device uses the address sent by the RADIUS server
RADIUS returns the IP address 255.255.255.254	SMS device uses the address contained in the AAA hint
RADIUS returns a named IP pool attribute	SMS device uses an unused IP address in the named pool
RADIUS returns no IP address at all	SMS device uses the address contained in the AAA hint unless RADIUS also instructs the SMS device to set itself up as a Layer 2 Tunneling Protocol (L2TP) access concentrator (LAC) or a link in a multilink PPP connection. In either of those cases, the SMS device responds as appropriate to carry out those instructions.

To use the AAA hint feature, you must enable AAA hint in the context using the **aaa hint ip-address** command in context configuration mode, and there must be a valid pool of IP addresses in the context from which an address can be selected. As long as both these conditions are met, the address can be selected prior to authentication and included in the RADIUS authentication request.

Configuration Tasks

To configure AAA hint, perform the tasks described in the following sections:

- Configure IP Pools
- Enable AAA Hint

Configure IP Pools

The AAA hint feature, even if enabled, will have no effect unless there is a locally-managed pool of IP addresses from which the AOS can select the preferred IP address for a subscriber. To configure a locally managed pool of IP addresses, enter the following command in interface configuration mode:

ip pool *address netmask*

The pool is derived by applying the netmask specified in the *netmask* argument to the address specified in the *address* argument, thus obtaining the network portion of the address. The interface address, the interface all-zeroes address, and the interface broadcast address are automatically excluded if they overlap the pool. See the "Interface Commands" chapter in the *Access Operating System (AOS) Command Reference* publication for a detailed description of the **ip pool** command.

To establish that a subscriber is to be assigned an IP address from a locally managed address pool, enter the following command in subscriber configuration mode:

ip address {*address* [*mask*] | **pool** [**name** *if-name*]}

See the "Subscriber Commands" chapter in the *Access Operating System (AOS) Command Reference* publication for a detailed description of the **ip address** command.

Enable AAA Hint

To enable the AAA hint feature in a context, enter the following command in context configuration mode:

aaa hint ip-address

Configuration Examples

The following example enables the AAA hint feature in the `customers` context, sets subscriber authentication to use RADIUS, and sets up a locally managed IP address pool for each of two interfaces within the `customers` context:

```
[local]RedBack(config)#context customers
[local]RedBack(config-ctx)#aaa hint ip-address
[local]RedBack(config-ctx)#aaa authentication subscriber radius
[local]RedBack(config-ctx)#radius server 10.11.39.153 key secret
[local]RedBack(config-ctx)#interface one
[local]RedBack(config-if)#ip address 1.1.1.1 255.255.255.0
[local]RedBack(config-if)#ip pool 1.1.1.0 255.255.255.0
[local]RedBack(config-if)#interface two
[local]RedBack(config-if)#ip address 2.2.2.2 255.255.255.0
[local]RedBack(config-if)#ip pool 2.2.2.0 255.255.255.0
[local]RedBack(config-if)#port ethernet 6/0
[local]RedBack(config-port)#encapsulation ppp over-ethernet
[local]RedBack(config-port)#bind authentication chap pap
[local]RedBack(config-port)#no shut
[local]RedBack(config-port)#end
```

Two-Stage Accounting

The RADIUS accounting data can be sent to a global set of RADIUS servers, a context-specific set of RADIUS servers, or both. This last case (both global and context-specific accounting) is referred to as two-stage accounting. An example of two-stage accounting is an application in which a wholesaler wants to send a copy of accounting data to both his own RADIUS server and to an upstream provider's RADIUS server, allowing end-of-period accounting data to be reconciled and validated by both parties.

Note As with global RADIUS-based authentication, the global RADIUS accounting server's IP address and other attributes are configured in the local context. It is, therefore, not possible to do two-stage accounting in the local context.

The following actions are required to configure two-stage accounting:

1. Enable global RADIUS accounting for subscribers.
2. Configure the IP addresses of the global RADIUS accounting servers in the local context.
3. Enable RADIUS accounting for subscribers in the relevant context.
4. Configure the IP addresses of the context's RADIUS accounting servers.

The following commands configure a global RADIUS server (10.0.0.1) and a context-specific RADIUS server for the isp.net context (172.16.0.1):

```
[local]RedBack(config)#aaa accounting subscriber radius
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#radius server 10.0.0.1 key secret1
[local]RedBack(config-ctx)#exit
[local]RedBack(config)#context isp.net
[local]RedBack(config-ctx)#aaa accounting subscriber radius
[local]RedBack(config-ctx)#radius server 172.16.0.1 key secret2
```

As configured, accounting data for traffic bound to the isp.net context is sent to both the global server (10.0.0.1) and the isp.net server (176.16.0.1).

Marking a Context for Explicit Binding Only

This feature affects how Point-to-Point Protocol (PPP) sessions are permitted to bind to the context in which the command that enables the feature is entered. When the **aaa binding explicit-only** setting is enabled, the context and its domains become ineligible for dynamic binding by a PPP session. Sessions can only bind to the context if:

- The session arrives over a circuit, tunnel, or port to which a service access list has been applied that permits that context or domain (using the **bind authentication** ... **service-group** *group-name* command or the **session-auth** ... **service-group** *group-name* command in the case of tunnels).
- The context is explicitly named in a **bind authentication** ... **context** *ctx-name* command (or **session-auth** ... **context** *ctx-name* command in the case of tunnels).

Unless this command is used, dynamic binding is allowed in the context and its domains.

To configure a context for explicit binding only, thereby disabling dynamic binding for that context, enter the following command in context configuration mode:

aaa binding explicit-only

The following behaviors result:

- A permanent virtual circuit (PVC) configured with the **bind authentication** {**pap** | **chap** | **chap pap**} command (no context or service group name specified) experiences a change in behavior when the **aaa binding explicit-only** setting is enabled. Sessions are not allowed to bind to any contexts (or tunnels in those contexts) where the **aaa binding explicit-only** setting is enabled. The same is true for a tunnel configured with the **session-auth** {**pap** | **chap** | **chap pap**} command (no context or service-group name specified).
- A PVC configured with the **bind authentication** {**pap** | **chap** | **chap pap**} **context** *ctx-name* experiences no change in behavior, because sessions are already explicitly restricted to the context. The **aaa binding explicit-only** setting imposes no additional restriction. The same is true for a tunnel configured with the **session-auth** {**pap** | **chap** | **chap pap**} **context** *ctx-name* command.

- A PVC configured with the **bind authentication** {**pap** | **chap** | **chap pap**} **service-group** *group-name* command experiences no change in behavior, because whether or not sessions are allowed to terminate in a context is based on the criteria specified by the service access list. The **aaa binding explicit-only** command imposes no additional restriction. The same is true for a tunnel configured with the **session-auth** {**pap** | **chap** | **chap pap**} **service-group** *group-name* command.
- A PVC configured with the **bind session** *peer-name context* command experiences no change in behavior. All sessions proceed through the tunnel, regardless of the **aaa binding explicit-only** setting for the context.

Enabling Access Control List Downloading

You can configure IP access control lists locally, or you can configure them remotely and download them via RADIUS. You must enable the ability to download access lists on a per-context basis. When enabled, if a requested access list does not appear in the local configuration, the AOS looks for the list in the RADIUS database and downloads it from there. The list stays resident, as long as there are subscribers referencing it. When there are no more subscribers referencing a list, the list is deleted from the system.

To enable access list downloading in a context, enter the following command in context configuration mode:

aaa authorization access-list radius

The following is an example of an access control list named `general` that is defined remotely using RADIUS attributes:

```
general
Password = “Redback”
Service-Type = Access-Control-List
Redback:ACL-Definition = “redirect interface one 10.1.1.1. any”,
Redback:ACL-Definition = “deny icmp 12.1.1.1 12.3.4.23 any 100 200”,
Redback:ACL-Definition = “permit ip any any”,
Redback:ACL-Definition = “deny ip 121.24.234.12 121.3.34.129 23.4.34.0 25.23.56.12”
```

The ACL-Definition vendor-specific attribute (VSA) uses the same syntax as the AOS IP access control list commands (see the “IP Access Control List Commands” chapter in the *Access Operating System (AOS) Command Reference* publication).

See Chapter 37, “Configuring IP Access Control Lists,” for information on clearing and displaying downloadable access control lists.

Configuring Custom Formats for Structured Usernames

This section describes how to override the AOS default username format of *user@domain*. You can configure multiple custom formats, in which case, the first format that matches is used.

The following sections are included:

- Implications of Customizing Username Formats

- Configuration Tasks
- Configuration Examples

For a complete description of the commands related to custom structured usernames, see the *Access Operating System (AOS) Command Reference* publication.

Implications of Customizing Username Formats

The AOS default username format of *user@domain* is always expected, unless custom formats have been configured. When creating custom formats, be aware of the related functions that could change as a result. For example:

- The structured username format used in any **bind subscriber** commands must match a valid format as defined by the rules you establish when you customize.
- In the **bind authentication** command, the username string provided during Point-to-Point Protocol (PPP) authentication is interpreted according to those same rules.
- If you configure any custom format, you must also explicitly specify that the format *user@domain* be checked for a match if you want it to be checked. It will not automatically be checked.
- The behavior of the **radius strip-domain** command is affected in that it will strip the domain portion of the structured username in accordance with the format matched by that structured username.
- In addition to actually creating the custom username formats, you must also consider what the default domain name will be and how (or if) it should be applied in the case where an unstructured username is provided during authentication. Use the **aaa default-domain** command for this purpose; it is discussed in the "Configuration Tasks" section.

Configuration Tasks

To configure custom formats for structured usernames, perform the tasks in the following sections:

- Define One or More Custom Formats
- Designate the Default Domain and Define its Behavior

Define One or More Custom Formats

To define a custom username format, enter the following command in global configuration mode:

aaa username-format {**domain** | **username**} *separator-char*

The *separator-char* argument can be one of the following: %, -, @, \\, #, or /. The separator character is the designated separation between the username portion of the structured username and the domain portion. When you want the separator character to be a backslash (\), you must enter it in this command as two backslashes (\\). A single backslash has a reserved meaning in the AOS.

The choice of **domain** | **username** allows you to specify whether the domain portion or the username portion should precede the other. Using this command, you can configure a large variety of custom username formats. Here are some examples:

domain@username
username/*domain*
username%*domain*
domain#*username*

You can configure multiple custom formats in which case, the one that matches first is used. Once you configure a format with a particular separator character however, you will not be allowed to configure another format using the same separator.

If you configure any custom format and you also want the AOS default format of *user@domain* to be searched for a match, you must explicitly configure *user@domain* as a custom format. If you want only the AOS default format to be used, you do not need to configure *user@domain* as a custom format; it will be used automatically.

Designate the Default Domain and Define its Behavior

To designate a default domain name and define how it is to be applied to an unstructured username, enter the following command in global configuration mode:

aaa default-domain *dom-name* [**username-format** {**domain** | **username**} *separator-char*]

When used without the optional construct, this command simply indicates the name of the default domain and specifies that the AOS default of *@domain* is to be appended to any unstructured username that is provided.

The optional construct provides the opportunity to define a different way to apply the default domain name to an unstructured username. As in the **aaa username-format** command, the choice of **domain** | **username** allows you to specify whether the domain portion or the username portion should precede the other. The *separator-char*, the separation between the two portions of the structured username, must be one of the following: %, -, @, \ (entered as \\), #, or /.

The default domain name is applied only if the username is deemed "unstructured." The username would be considered unstructured if:

- Custom formats have been configured, and the name does not match any configured custom format.
- Custom formats have not been configured, and the name does not match the AOS default of *user@domain*.

If you do not use the **aaa default-domain** command at all, no domain name is applied to an unstructured username.

Configuration Examples

The following example configures two custom username formats and establishes the default domain name and behavior:

```
[local]RedBack(config)#aaa username-format username @
[local]RedBack(config)#aaa username-format domain \\
[local]RedBack(config)#aaa default-domain karmatics username-format domain \\
```

Even though *user@domain* is the AOS default format, it needs to be explicitly configured here to be used because an additional format is also configured.

With these two formats in place, a username would first be checked for a match to the format *user@domain*. If it does not pass authentication with that format, it would be checked for a match to the format *domain\user*.

If the name does not match either of the two custom formats, `karmatics\` would be prepended to the entire name and submitted for authentication. If there is no match at that time, the authentication fails.

Suppose the username submitted is **jack-intranet** where jack is the username, intranet is the domain name and - is the separator. The name would not match either of the two configured custom formats, so it would be treated as an unstructured username. In this example configuration, unstructured usernames have **karmatics** prepended to them so the name becomes **karmatics\jack-intranet**. If there is no match with this name, authentication fails.

Configuring RADIUS

This chapter provides an overview of the Remote Authentication Dial-In User Service (RADIUS) and describes the tasks involved in configuring RADIUS features through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands mentioned, see the "RADIUS Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- RADIUS Redundancy and Load Balancing
- Separate RADIUS Authentication and Accounting Servers
- Acct-Session-Id Attribute
- Configuration of IP Access Control Lists via RADIUS
- Auto-Subscriber Function
- Locally Managed IP Address Pools
- Configuring an Interface IP Address as the Source for RADIUS Packets
- Support for Tagged Attributes
- Combining RADIUS Features

Overview

The RADIUS protocol enables the building of a system that secures remote access to networks and network services. RADIUS is based on a client/server architecture. You can configure the AOS to act as a RADIUS client. The use of RADIUS replaces the need for local configuration of subscriber records.

Before using RADIUS, you must first configure the AOS with the IP address or hostname of one or multiple RADIUS servers. You can only use the optional *hostname* argument if you have enabled the Domain Name System (DNS) via the **ip domain-lookup**, **ip domain-name**, and **ip name-servers** commands in context configuration mode. See the "DNS Commands" chapter in the *Access Operating System (AOS) Command Reference* publication for descriptions of these commands.

The following command indicates that the RADIUS server's IP address is `10.10.1.2`. The RADIUS client uses the `opensesame` string to authenticate its requests to the server.

```
[local]RedBack(config)#context bigisp
[local]RedBack(config-ctx)#radius server 10.10.1.2 key opensesame
```

You can configure RADIUS servers on a per-context basis or on a global basis. You can configure the AOS to use global authentication, authorization, and accounting (AAA) servers. The global RADIUS server feature allows an administrator to configure a single RADIUS server to provide all subscriber authentication and accounting for an entire Subscriber Management System (SMS) device, regardless of context. Configure global servers in the local context. You can also configure servers on a per-context basis with separate RADIUS servers for subscribers in each context as shown in the previous example. If global authentication is enabled, you can also set up RADIUS to override the service portion of a subscriber's structured username (for example, the *@context* portion), and cause the subscriber's session to be placed in a specified context.

Note Each SMS device can have up to 256 simultaneous outstanding authentication and accounting requests, or a total of 512 simultaneous requests, per server.

Within a context, all subscriber AAA is accomplished either through local configuration (subscriber records) or through a remote server (the RADIUS server). You can perform AAA functions for administrator accounts through RADIUS. In general, it is recommended to authenticate administrators locally in the event a network disruption makes communications between the SMS device and the RADIUS server impossible. In this event, if authentication for administrator accounts is via RADIUS, logging on to the system would not be possible.

The following commands enable RADIUS authentication and authorization for all subscribers. In addition, the system is configured to send accounting information (such as packets and bytes, sent and received per subscriber) to the RADIUS server for both subscribers and administrators.

```
[local]RedBack(config-ctx)#aaa authentication subscriber radius
[local]RedBack(config-ctx)#aaa accounting subscriber radius
[local]RedBack(config-ctx)#aaa accounting administrator radius
```

The AOS sends the entire set of attributes applied to the subscriber's session to RADIUS accounting. In particular, the IP address that was dynamically assigned to a subscriber's Point-to-Point Protocol (PPP) session is sent to RADIUS accounting, providing an audit trail to trace perpetrators of denial-of-service and other types of network attacks. Similarly, the policing and rate-limiting parameters applied to the session, which might vary with the context (service) selected by a subscriber on a per-session basis, are available in RADIUS accounting.

Configure the Interface's IP Address as the Source for RADIUS Packets

To specify that an interface's primary IP address is to be considered the source address for all RADIUS packets that are sent from the context in which the interface is configured, enter the following command in interface configuration mode:

ip source-address radius

When this feature is enabled, the interface's primary IP address is used as the source, regardless of the interface through which the packet is actually being sent out. See the "Interface Commands" chapter in the *Access Operating System (AOS) Command Reference* publication for syntax and usage guidelines for the **ip source-address** command.

RADIUS Redundancy and Load Balancing

The AOS allows load balancing across multiple RADIUS servers for both accounting and authentication. A number of different load-balancing algorithms are supported. This feature is useful in situations where the amount of session churn (the number of sessions being established and terminated per second) is large and a single RADIUS server is unable to handle the offered load.

When multiple RADIUS servers are configured for a context, the AOS uses them on a strict-priority basis or on a round-robin basis. In either case (assuming that separate authentication and authorization servers are not configured), the AOS attempts to send a subscriber's accounting data to the same server that was used to authenticate the subscriber session. Only if that server is not responding does the AOS send accounting data to a different server than the one used to authenticate the subscriber.

Note Each configured RADIUS client on the SMS device can generate up to 256 simultaneous authentication and accounting requests (for a total of 512 simultaneous requests per client). Some RADIUS servers are configured by default to support only a maximum of 128 simultaneous authentication and accounting requests (for a total of 256 simultaneous requests). Once the authentication queue is filled, all subsequent authentication requests are automatically rejected (the server sends an Access-Reject message to the SMS device). Flooding the RADIUS server with more requests than it can handle can cause serious performance problems. This problem is usually resolved by rebuilding the RADIUS server image with an increased upper request limit.

The following commands configure the `corp.com` context to use round-robin load balancing across RADIUS servers `10.0.0.1` and `10.0.0.2`:

```
[local]RedBack(config)#context corp.com
[local]RedBack(config-ctx)#aaa authentication subscriber radius
[local]RedBack(config-ctx)#aaa accounting subscriber radius
[local]RedBack(config-ctx)#radius server 10.0.0.1 key MyKeyOne
[local]RedBack(config-ctx)#radius server 10.0.0.2 key MyKeyTwo
[local]RedBack(config-ctx)#radius algorithm round-robin
```

Separate RADIUS Authentication and Accounting Servers

You can configure separate RADIUS servers for accounting and for authentication. This provides additional flexibility and load-balancing opportunities for the network administrator.

The following commands configure context `isp.net` to use separate RADIUS servers for authentication (`176.16.0.1`) and accounting (`176.16.0.2`):

```
[local]RedBack(config)#context isp.net
[local]RedBack(config-ctx)#aaa authentication subscriber radius
[local]RedBack(config-ctx)#aaa accounting subscriber radius
[local]RedBack(config-ctx)#radius server 176.16.0.1 key MyKeyOne
[local]RedBack(config-ctx)#radius accounting server 176.16.0.2 key MyKeyTwo
```

Both the **radius server** and **radius accounting server** commands have the option of specifying a hostname rather than an IP address, but only if DNS has been enabled via the **ip domain-lookup**, **ip domain-name**, and **ip name-servers** commands in context configuration mode. See the "DNS Commands" chapter in the *Access Operating System (AOS) Command Reference* publication for descriptions of these commands.

Acct-Session-Id Attribute

You have the option of configuring the SMS device to send the Acct-Session-Id attribute in Access-Request packets, in addition to the default behavior of sending the attribute in Accounting-Request packets. When this feature is in effect, the SMS device creates the Acct-Session-Id attribute when it starts authentication, and then uses it in the Access-Request and Accounting-Request packets.

To configure the SMS device to send the Acct-Session-Id attribute in Access-Request packets, enter the following command in context configuration mode:

radius attribute acct-session access-request

Configuration of IP Access Control Lists via RADIUS

You can configure IP access control lists locally, or you can configure them remotely and download them via RADIUS. The ability to download access lists has to be enabled on a per-context basis. When enabled, if a requested access list does not appear in the local configuration, the SMS device looks for the list in the RADIUS database and downloads it from there. The list stays resident, as long as there are subscribers referencing it. When there are no more subscribers referencing a list, the list is deleted from the system.

To enable the ability to download access control lists in a context, enter the following command in context configuration mode:

aaa authorization access-list radius

See the "AAA Commands" chapter in the *Access Operating System (AOS) Command Reference* publication for a detailed description of this command.

The following example uses RADIUS attributes to remotely define an access control list named `general`:

```
general
Password = “Redback”
Service-Type = Access-Control-List
Redback:ACL-Definition = “redirect interface one 10.1.1.1. any”,
Redback:ACL-Definition = “deny icmp 12.1.1.1 12.3.4.23 any 100 200”,
Redback:ACL-Definition = “permit ip any any”,
Redback:ACL-Definition = “deny ip 121.24.234.12 121.3.34.129 23.4.34.0 25.23.56.12”
```

The ACL-Definition VSA uses the same syntax as the AOS IP access control list commands (see the “IP Access Control List Commands” chapter in the *Access Operating System (AOS) Command Reference* publication).

To dereference a downloaded access control list from bound subscribers, enter the following command in administrator exec mode:

clear access-list *ctx-name* [*list-name*]

The *ctx-name* argument is the name of the context in which you want to clear the access list and the *list-name* argument is the name of the access list. You can also clear all downloaded access lists in the context by omitting the optional *list-name* argument.

Auto-Subscriber Function

When configuring ATM PVCs using RFC 1483 encapsulation to use RADIUS, a subscriber name must be associated with each circuit via configuration. Although the administrator can enter these names manually, the AOS supports a quick method for preallocating a collection of ATM PVCs with contiguous virtual circuit identifiers (VCIs) and automatically generating subscriber names and optional passwords. The following commands create five PVCs, each bound through an automatically generated subscriber name. The string following the **auto-subscriber** keyword can be any arbitrary string and is used as the leading characters in the subscriber names.

```
[local]RedBack(config)#port atm 2/0
[local]RedBack(config-port)#atm pvc 0 100 through 105 profile adsl encapsulation
route1483
[local]RedBack(config-pvc)#bind auto-subscriber green local
```

The following lines are entered into the system configuration as a result of entering the previous commands:

```
port atm 3/0
  atm pvc 0 100 profile adsl encapsulation route1483
   bind subscriber green2.0.0.100@local
  atm pvc 0 101 profile adsl encapsulation route1483
   bind subscriber green2.0.0.101@local
  atm pvc 0 102 profile adsl encapsulation route1483
   bind subscriber green2.0.0.102@local
  atm pvc 0 103 profile adsl encapsulation route1483
   bind subscriber green2.0.0.103@local
  atm pvc 0 104 profile adsl encapsulation route1483
   bind subscriber green2.0.0.104@local
  atm pvc 0 105 profile adsl encapsulation route1483
```

```
bind subscriber green2.0.0.105@local
```

The subscriber names are of the form <*string*><*slot*>.<*port*>.<*vpi*>.<*vci*> and are automatically generated.

Locally Managed IP Address Pools

Administrators can manage a pool of addresses for assignment to subscribers, with or without RADIUS configuration. Enabling locally managed IP address pools requires the following:

- Configuring the pool itself within one or more interfaces
- Configuring subscriber records (either locally or within a RADIUS server) to indicate that address assignment is to be via an address pool
- Optionally, specifying that the subscriber's address is to be assigned from the address pool configured in a particular interface

The following example sets up the `isp.net` context to use a locally managed IP address pool in interface `downstream`. Addresses are assigned from the pool starting at address 10.0.0.2 with a netmask of `255.255.255.0`. The interface address, the interface all-zeroes address, and the interface broadcast address are automatically excluded if they overlap the pool. Here, it is assumed that RADIUS returns a Framed-IP-Address attribute value of 255.255.255.254 (or 0.0.0.0) for all subscribers for which the SMS device is to assign an address from the pool:

```
[local]RedBack(config)#context isp.net
[local]RedBack(config-ctx)#aaa authentication subscriber radius
[local]RedBack(config-ctx)#interface downstream
[local]RedBack(config-if)#ip address 10.0.0.1 255.255.255.0
[local]RedBack(config-if)#ip pool 10.0.0.2 255.255.255.0
```

Alternatively, you can configure the SMS device to authenticate subscribers locally. The following example configures a subscriber named `mary` and establishes that the subscriber's IP address is to be assigned from the SMS-managed pool:

```
[local]RedBack(config)#context corp.com
[local]RedBack(config-ctx)#aaa authentication subscriber local
[local]RedBack(config-ctx)#interface telecommuters
[local]RedBack(config-if)#ip address 176.16.0.1 255.255.255.0
[local]RedBack(config-if)#ip pool 176.16.0.2 255.255.255.0
[local]RedBack(config-if)#exit
[local]RedBack(config-ctx)#subscriber name mary
[local]RedBack(config-sub)#ip address pool
```

Note You can also specify the **ip address pool** command within the **default** subscriber record. If global authentication is enabled, the RADIUS server returns a VSA containing the context to which the subscriber should be assigned and does not return a Framed-IP-Address attribute. When the default subscriber record in the specific context indicates the **ip address pool** command, the AOS can then assign IP addresses from the pool in the context.

Note The **ip address pool** command has an optional **name** *if-name* construct that allows you to specify the address pool configured in a specific interface for purposes of assigning the subscriber's address. If no addresses in that pool are available, the session fails. If the **name** *if-name* construct is not used in the **ip address pool** command, the AOS is free to search for an available address in any available pool in any available interface in the appropriate context.

Configuring an Interface IP Address as the Source for RADIUS Packets

To configure the interface's primary IP address as the source address for all RADIUS packets that are sent from the context, enter the following command in interface configuration mode using the **radius** keyword:

ip source-address {**snmp** [**radius**] | **radius** [**snmp**]}

Note This command is also described in Chapter 7, "Configuring Interfaces."

Support for Tagged Attributes

The AOS supports tagged attributes as defined in the Internet-Draft, *RADIUS Attributes for Tunnel Protocol Support*.

Combining RADIUS Features

The following example combines several of the features described earlier in this section. Specifically, the example demonstrates the set up of two-stage, redundant, RADIUS accounting, in conjunction with global, redundant, authentication via RADIUS.

```
[local]RedBack(config)#aaa accounting subscriber radius
[local]RedBack(config)#aaa authentication subscriber radius
[local]RedBack(config)#context local
[local]RedBack(config-ctx)#radius server 10.0.0.1 key 1
[local]RedBack(config-ctx)#radius server 10.0.0.2 key 22
[local]RedBack(config-ctx)#radius algorithm round-robin
[local]RedBack(config-ctx)#radius accounting server 10.0.0.3 key 333
[local]RedBack(config-ctx)#radius accounting server 10.0.0.4 key 4444
[local]RedBack(config-ctx)#radius accounting algorithm round-robin
[local]RedBack(config-ctx)#exit
[local]RedBack(config)#context corp.com
[local]RedBack(config-ctx)#aaa accounting subscriber radius
[local]RedBack(config-ctx)#radius accounting server 172.16.0.3 key cc
[local]RedBack(config-ctx)#radius accounting server 172.16.0.4 key d
[local]RedBack(config-ctx)#radius accounting algorithm round-robin
```

Part 12

System Management

Monitoring and Testing System Parameters

This chapter provides information on monitoring and testing system-wide parameters and describes the managements tasks involved. For examples and detailed information on syntax and usage guidelines for the commands listed under the "Configuration Tasks" section, see the "System Monitoring and Testing Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

Typically, the Access Operating System (AOS) **show** and **debug** commands are used to provide information to verify correct system operation and to troubleshoot problems. This chapter describes general system-wide monitoring and testing tasks, such as displaying system memory and processes, displaying all system hardware, testing IP connectivity, and enabling debugging messages for all IP packets.

For information on **show** and **debug** commands that are specific to a feature, interfaces, subscribers, ports, or circuits, see the appropriate chapter in this guide. For example, to find out how to display or debug OSPF parameters, see Chapter 33, "Configuring OSPF."

Configuration Tasks

To monitor and test system parameters, perform the tasks described in the following sections:

- Display System Information
- Enable Debugging Messages
- Display Debugging Processes

- Test Connectivity
- Test the Switch Fabric
- Clear Sessions, Circuits, and Fabric Counters

Display System Information

You can display system-wide information, such as active administrators and operators, results of diagnostics tests, hardware types and slot locations, system memory, and so on.

Show Administrators

To display a list of administrators and operators active in all contexts, enter the following command in operator exec mode:

show administrators [**active** [*name*]]

Use the **active** keyword to display summary information about administrators in the current context. Specify a *name* to include only information about the specified administrator.

Show Subscribers

To display subscriber information, enter the following command in operator exec mode:

show subscribers [**access-statistics** [*sub-name*] | **active** [*sub-name*] | **address** *sub-name* | **all** | **minimums** [*ctx-name* | **all**] | **summary**]

Show Diagnostics

To display the results of diagnostic tests, enter the following command in operator exec mode:

show diag [**all** | **backplane** | **ce** | **cm** [*slot*] | **fabric** | **fe** | **midplane** | **power** | *slot*/*port* | **sm** [*slot*] | **timing**] [**err**] [**long**]

The **show diag** command displays the results of power-on diagnostics tests. By default, diagnostics results for all system hardware are displayed. To filter output, use an optional keyword or argument. Use the **err** keyword to display the diagnostics error log and the **long** keyword to provide results and a list of each type of diagnostic test performed.

Show Tech

To save information about the state of the system to a file, enter the following command in administrator exec mode:

show tech *url* [**compress-level** *level*] [**details**] [**-noconfirm**] [**show-password**]

Show Environmental Monitoring

To display environmental monitor status, including fan and power supply failures, enter the following command in operator exec mode:

show envmon

Show Fabric

To display counters for the switch fabric on Connection Manager (CM) and System Manager (SM) modules, enter the following command in operator exec mode:

show fabric counters [*slot* | **all**]

To display the switch fabric table, enter the following command:

show fabric table [**detail**]

Show Hardware

To display information about system hardware, enter the following command in operator exec mode:

show hardware [**all** | **backplane** | **cm** [*slot*] | **fabric** | **fe** | **midplane** | **power** | *slot*/*port* | **sm** [*slot*] | **timing**]

To display detailed information about all the hardware in the system, use the optional **all** keyword. To display detailed information about a specific element, use any other optional keyword or argument. Information displayed includes hardware type, serial number, part number, physical connector, and so on.

Show TCP and UDP Sockets

To display a table of all Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) sockets in use for the current context, enter the following command in operator exec mode:

show ip socket

Show IP Traffic

To display IP traffic statistics for the current context, enter the following command in operator exec mode:

show ip traffic

Show Memory

To display the amount of memory that is in use or is available for Control Engine (CE), CM, Forwarding Engine (FE), or SM modules, depending on the Redback platform, enter the following command in operator exec mode:

show memory [**ce** | **cm** *slot* | **fe** | **sm**]

Show System Processes

To display information on current system processes, enter the following command in operator exec mode:

show process [**cpu** [**non-zero**]]

Use this command without any keywords to display all system processes. Use the optional **cpu** keyword to display average CPU utilization statistics for processes. Use the **cpu non-zero** construct to display only processes with nonzero CPU utilization values.

Show FE Statistics

To display statistics for the FE module, enter the following command in operator exec mode:

show fe stats

Show CM Information

To display statistics for a single CM module, enter the following command in operator exec mode:

show cm stats *slot*

To display the status of all CM modules in the system, enter the following command:

show cm table

Show Slot

To display information about the I/O module hardware type for all slots or for a specified slot, enter the following command in operator exec mode:

show slot {**table** | *slot*}

Show SRAM

To display the amount of static RAM (SRAM) and the format of data stored for each PCMCIA card installed in the Subscriber Management System (SMS) device, enter the following command in operator exec mode:

show sram

Show Stack

To display information about the last system restart, enter the following command in operator exec mode:

show stack

Note Redback Networks technical support personnel may ask you to use the **show stack** command to display the saved stack information from a restart caused by a system error.

Enable Debugging Messages

When you enable debugging messages, the AOS provides output on the specified parameters and logs the messages to the system log file. In addition, you can use the **logging console** or **terminal monitor** commands to display the messages in real time.

Caution Debugging can severely affect system performance. Exercise caution before enabling any debugging on a production system.

Note All commands described in this section are found in administrator exec configuration mode.

Enable All Debugging Messages

To enable the logging of debugging messages for all possible options in the AOS, enter the following command:

debug all

Enable All IP Debugging Messages

To enable the logging of IP debugging messages, enter the following command:

debug ip all

Enable IP Host Debugging Messages

To enable the logging of IP host debugging messages, enter the following command:

debug ip host

Enable IP Packet Debugging Messages

To enable the logging of IP packet debugging messages, enter the following command:

debug ip packet

Enable ICMP Debugging Messages

To enable the logging of Internet Control Message Protocol (ICMP) debugging messages, enter the following command:

debug ip icmp

Enable TCP Debugging Messages

To enable the logging of Transmission Control Protocol (TCP) debugging messages, enter the following command:

debug ip tcp

Enable IP CE-FE or SM-CM Debugging Messages

To enable the logging of debugging messages for CE and FE modules, or for SM and CM modules, depending on the product platform, enter the following command:

debug ip ce-fe

debug ip sm-cm

Display Debugging Processes

To display the debugging options that are currently enabled, enter the following command in operator exec configuration mode:

show debugging

Test Connectivity

Test connectivity by verifying IP reachability of hosts and tracing IP route routes.

Ping Connections

To verify whether an IP host is reachable or not, enter the following command in operator exec configuration mode:

ping {*ip-address* | *hostname*} [*number-of-packets*] [**interface** *name* | **src** *ip-address*] [**pattern** *hex-pattern*] [**size** *bytes*] [**timeout** *seconds*]

Use Traceroute

To trace an IP route to its destination, enter the following command in operator exec configuration mode:

traceroute {*ip-address* | *hostname*} [**count** *number*] [**df**] [**maxttl** *ttl*] [**minttl** *ttl*] [**port** *port*] [**size** *bytes*] [**src** *ip-address*] [**timeout** *seconds*]

This command allows you to discover the routes that packets take when traveling to the specified destination. Each line in the output from the command shows the next hop in the path between the SMS device and the destination address.

Test the Switch Fabric

To test the switch fabric using all SM and CM modules in the system, enter the following command in operator exec configuration mode:

fabric test

This command temporarily (for less than one second, in most cases) disrupts communication over the fabric.

Clear Sessions, Circuits, and Fabric Counters

You can clear administrator and operator Telnet sessions and subscriber sessions, clear circuits from configuration, and clear the counters on CM and SM modules associated with Switch Fabric 42G modules.

Administrator and Operator Sessions

To end a Telnet session for an administrator or operator, enter the following command in operator exec configuration mode:

clear administrator *name*

An administrator in the local context can end any administrator session. Administrators in any other context can only end sessions in their own context. The *name* argument must include the context in which the administrator or operator is configured; for example, *joe@local*.

Subscriber Sessions

To clear the circuit of an active subscriber, enter the following command in operator exec configuration mode:

clear subscriber *name*

The command is useful when a subscriber's record has changed and you want the new parameters to take effect immediately, and when the user account has been removed and you want to log the user off.

Circuits

To clear a circuit from configuration, enter the following command in operator exec configuration mode:

clear circuit {*slot/port* {*vpi vci* [**through** *end-vci*] | [*hdlc-channel*] *dlci* [**through** *end-dlci*] | **all**} | **pppoe** [*cm-index*-]*session-id* [**through** [*end-session-id*] | **all**}

This command tears down any active subscriber sessions on the specified circuit or circuits. The **clear circuit** command is similar to the **clear subscriber** command, except that instead of specifying the username, you specify the circuit. This is particularly useful when a subscriber is using multiple circuits and there is only one that you want to clear.

Fabric Counters

To clear the counters on CM, SM, and Switch Fabric modules, enter the following command in administrator exec configuration mode:

clear fabric counters {*slot* | **all**}

Configuring Bulk Statistics

This chapter provides an overview of bulk statistics (bulkstats) features and describes the tasks involved in configuring bulkstats through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed under the "Configuration Tasks" section, see the "Bulk Statistics Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

Bulk statistics (bulkstats) gathers large amounts of data from the Subscriber Management System (SMS) device, periodically sending updates to a management station. Bulkstats free both the SMS device and the management station from the Simple Network Management Protocol (SNMP) polling processes, and minimize the amount of memory used by the SMS device for statistics collection.

An administrator can configure bulkstats *schemas* to define the statistics to be collected and determine the format in which the data is displayed. A schema consists of a name, a display format, and a list of statistics.

Bulkstats are collected and transferred to a management station as follows:

1. The SMS device samples and stores system, network, traffic, and error statistics at specified sampling intervals. Information can be collected at the system, port profile, port, the default Layer 2 Tunneling Protocol (L2TP) peer, High-Level Data Link Control (HDLC) channel, the default subscriber, and circuit level. Bulkstats data is stored in the form of continuous counter values.
2. Data is periodically sent at a specified transfer interval via FTP or Trivial File Transfer Protocol (TFTP) to a network management station. The file sent is an ASCII format file consisting of data lines of ASCII text terminated by a UNIX new line.
3. Once the file is successfully transmitted, the information is deleted from SMS device memory.

Format Strings and Special-Character Sequences

Bulkstats schemas employ a format string that uses special-character sequences; see Table 43-1.

Table 43-1 Format String Special-Character Sequences

Syntax	Description
\n	A new line is created
%s	A character string
%d	An integer in decimal (base 10)
%u	An unsigned integer in decimal (base 10)
%x	An integer in hexadecimal format (base 16)
%%	A single % character replacement in the output

Replacing Format Strings with AOS Variables

Format strings can be replaced with AOS variables, such as system uptime, date, time of day, port and slot number information, and more. Supported AOS variables vary according to command and configuration mode. See the subsections under the “Create Schemas” section for details.

Configuration Tasks

To configure bulkstats, perform the tasks described in the following sections:

- Enter Bulkstats Configuration Mode
- Configure a Primary, and Optionally, a Secondary Receiver
- Specify the Local Storage Directory
- Configure the Data Filename and Header Format
- Create Schemas
- Print Schema Definitions to the Data File
- Modify the Data Sampling and Transfer Intervals
- Force an Immediate Data Transfer
- Set a Limit on the Amount of Statistics That Can Be Collected
- Enable the Collection of Bulk Statistics
- Display Bulkstats Information

Enter Bulkstats Configuration Mode

To enter bulkstats configuration mode, enter the following command in global configuration mode:

bulkstats mode

Configure a Primary, and Optionally, a Secondary Receiver

To configure a FTP or a TFTP server to receive updated bulkstats data collection files, enter the following command in bulkstats configuration mode:

receiver *ip-address* {**primary** | **secondary**} [**mechanism** {**tftp** | **ftp** {**login** *name* **password** *passwd* | **nopassword**} [**passive**]]

Specify the Local Storage Directory

To define where the bulkstats samples and collection files are stored on the SMS device, enter the following command in bulkstats configuration mode:

localdir *dirname*

You must create a local directory using the **mkdir** command in administrative exec configuration mode before enabling bulkstats collection. You can specify a directory on /pcmcia0, /pcmcia1, or /flash (PCMCIA cards are preferrable due to faster write speed).

Configure the Data Filename and Header Format

To specify the name and location of data collection files on a bulkstats receiver, enter the following command in bulkstats configuration mode:

remotefile format *format-string* [*AOS-variable* [*AOS-variable...*]]

To define the header format of data collection files, enter the following command in bulkstats configuration mode:

header format *format-string* [*AOS-variable* [*AOS-variable...*]]

Use the *format-string* argument as the string to format the filename. String definitions follow the C programming language printf() function syntax. Table 43-1 describes the special-character sequences that can be used. Optional AOS-variables include: date, time of day, hostname, and system uptime.

Create Schemas

Schemas define the type and format of data that is collected. Schemas using system-level AOS variables are configured with the **schema** command in bulkstats configuration mode.

Schemas that are specific to ports and applied globally to all configured ports are configured with the **schema profile** command in global configuration mode.

Schemas that are applied to Asynchronous Transfer Mode (ATM) or Frame Relay port profiles, individual ports, default L2TP peers, HDLC channels, or default subscribers are configured with the **bulkstats schema** command in the following modes respectively:

- ATM profile configuration mode
- Frame Relay profile configuration mode
- port configuration mode
- L2TP peer configuration mode
- HDLC channel configuration mode
- subscriber configuration mode

Define System-Level Schema Formats

To define a system-level schema format, use the following command in bulkstats configuration mode:

schema *name* **format** *format-string* [*AOS-variable* [*AOS-variable...*]]

You can replace the *format-string* argument by the AOS variable. Table 43-2 lists all the possible variables for the **schema** command in bulkstats configuration mode.

Table 43-2 AOS Variables Used with schema Command

AOS Variable	Type	Description
ce_free_user_mem	Integer	Available Control Engine (CE) memory
ce_total_user_mem	Integer	Total CE memory
cpu1min	Integer	Average CPU usage for the last minute
cpu5min	Integer	Average CPU usage for the last five minutes
cpu5sec	Integer	Average CPU usage for the last five seconds
date	String	Today's date in YYYYMMDD format
epochtime	Integer	Time of day in epoch format (number of seconds since January 1, 1970)
fe_free_user_mem	Integer	Available Forwarding Engine (FE) memory
fe_total_user_mem	Integer	Total FE memory
hostname	String	System hostname
rcv_dropped	Integer	Total incoming packets dropped
sysuptime	Integer	System uptime in seconds
timeofday	String	Time of day in HHMMSS format using a 24-hour clock
total_subscribers	Integer	Total number of active subscribers across all contexts
xmt_dropped	Integer	Total transmit packets dropped
xmt_outstanding	Integer	Total packets remaining to be transmitted

Define Schema Profiles for Multiple Ports

You can define a schema profile and apply its parameters to multiple ports on the system. To do so, enter the following command in global configuration mode:

schema profile port *name* **format** *format-sting* [*AOS-variable* [*AOS-variable...*]]

Once you have created the schema profile, use the **bulkstats schema** command (using the **profile** keyword) in port configuration mode to apply the profile to multiple ports.

Table 43-3 lists the AOS variables used with the **schema profile** command.

Table 43-3 AOS Variables Used with schema profile Command

AOS Variable	Type	Description
description	String	Description of port
epochtime	Integer	Time of day in epoch format (number of seconds since January 1, 1970)
inoctets	Integer	Number of octets received on this circuit
inpackets	Integer	Number of packets received on this circuit
mcast_inoctets	Integer	Number of multicast octets received on this circuit
mcast_inpackets	Integer	Number of multicast packets received on this circuit
mcast_outoctets	Integer	Number of multicast octets sent on this circuit
mcast_outpackets	Integer	Number of multicast packets sent on this circuit
outoctets	Integer	Number of octets sent from this circuit
outpackets	Integer	Number of packets sent on this circuit
port	Integer	Port number on the I/O module
portspeed	Integer	Port speed in kbps
porttype	String	Port type
rcv_dropped	Integer	Receive packets dropped
slot	Integer	Slot number in the SMS
sysuptime	Integer	System uptime in seconds
xmt_dropped	Integer	Transmit packets dropped
xmt_outstanding	Integer	Transmit packets outstanding

Define Schema Formats in Miscellaneous Command Modes

You can define schema formats for a range of different command modes. To do so, enter the following command:

bulkstats schema *name* {**format** *format-string* | **profile** *profile-name*} [*AOS-variable* [*AOS-variable...*]]

The modes to which you apply this command are ATM profile, Frame Relay profile, HDLC channel, L2TP peer, port, and subscriber configuration.

The **profile** keyword is only available in port configuration mode.

In subscriber configuration mode, you can only apply a bulkstats schema to a default subscriber. No more than one bulkstats schema can be configured for any subscriber in the context. Changes to the bulkstats schema are applied to subscribers when new sessions are started.

In L2TP peer configuration mode, a bulkstats schema can only be applied to the default L2TP peer.

Caution It is possible to configure multiple schemas, each gathering a different type and format of data. It is advisable to minimize the number of schemas used to reduce impact on system performance. This is especially true for ATM profile, Frame Relay profile, HDLC channel, and port schemas. In those modes, you can instead create one schema that records several subsets of data. Separate each subset within the format string by entering the character sequence \n, which creates a new starting line in the output file.

You can replace the *format-string* argument by AOS variables. Table 43-4 lists the supported AOS variables for the **bulkstats schema** command.

Table 43-4 AOS Variables Used with bulkstats schema Command

AOS Variable	Type	Configuration Mode	Description
active_sessions	Integer	L2TP peer	Active L2TP sessions for the context
activesubs	Integer	context	Active subscribers for the context
active_tunnels	Integer	L2TP peer	Active L2TP tunnels for the context
bind_type	String	subscriber	Subscriber bind type
cct_handle	Integer	subscriber	Subscriber circuit
cctstate	String	ATM profile Frame Relay profile	Circuit status
channel	Integer	Frame Relay profile	Channel on the channelized DS-3 card
context_name	String	context subscriber	Name of the context
description	String	ATM profile Frame Relay profile HDLC channel L2TP peer port	Description of port
dlci	Integer	Frame Relay profile	Data-link circuit identifier
epochtime	Integer	all	Time of day in epoch format (seconds since 1/1/1970)
inoctets	Integer	all	Number of octets received on this circuit
inpackets	Integer	all	Number of packets received on this circuit
ip_addr	String	subscriber	Subscriber IP address
ip_mask	String	subscriber	Subscriber netmask
mcast_inoctets	Integer	ATM profile Frame Relay profile port subscriber	Number of multicast octets received on this circuit

Table 43-4 AOS Variables Used with bulkstats schema Command

AOS Variable	Type	Configuration Mode	Description
mcast_inpackets	Integer	ATM profile Frame Relay profile port subscriber	Number of multicast packets received on this circuit
mcast_outoctets	Integer	ATM profile Frame Relay profile port subscriber	Number of multicast octets sent on this circuit
mcast_outpackets	Integer	ATM profile Frame Relay profile port subscriber	Number of multicast packets sent on this circuit
outoctets	Integer	all	Number of octets sent from this circuit
outpackets	Integer	all	Number of packets sent on this circuit
peer_name	String	L2TP peer configuration	Name of the L2TP peer
port	Integer	ATM profile Frame Relay profile HDLC channel port	Port number on the I/O module
portspeed	String	port	Port speed in kbps
porttype	String	port	Port type
rcv_dropped	Integer	port	Receive packets dropped
slot	Integer	ATM profile Frame Relay profile HDLC channel port	Slot number in the SMS device
subscriber_name	String	subscriber	Name of the subscriber
sysuptime	Integer	all	System uptime in seconds
vpi	Integer	ATM profile	Virtual path identifier
vci	Integer	ATM profile	Virtual circuit identifier
xmt_dropped	Integer	ATM profile Frame Relay profile port	Transmit packets dropped
xmt_outstanding	Integer	ATM profile Frame Relay profile port	Transmit packets outstanding

Print Schema Definitions to the Data File

To print schema definitions to the data collection file (entries appear at the top of the file), enter the following command in bulkstats configuration mode:

schema-dump

Schema formats are displayed as follows:

schema-def *name format-string* [*AOS-variable* [*AOS-variable...*]]

Modify the Data Sampling and Transfer Intervals

To modify the interval the SMS device waits between collecting data samples, enter the following command in bulkstats configuration mode:

sample-interval *minutes*

To modify interval at which bulkstats data files are uploaded to receivers, enter the following command in bulkstats configuration mode:

transfer-interval *minutes*

Force an Immediate Data Transfer

To immediately transfer the bulkstats file to one of the configured receivers, rather than waiting for the next transfer interval, enter the following command in operator exec configuration mode:

bulkstats force transfer

Set a Limit on the Amount of Statistics That Can Be Collected

To set a limit on the space used to store bulkstats data, enter the following command in bulkstats configuration mode:

limit *kilobytes*

Enable the Collection of Bulk Statistics

To enable the collection of bulk statistics, enter the following command in global configuration mode:

bulkstats collection

Before you enable bulkstats collection, you must configure the following:

- One or more schema (use the **schema** or **bulkstats schema** command)
- The primary receiver (use the **receiver** command)
- The directory where samples and collection files are stored (use the **localdir** command)
- The name and location of the collection files on the server (use the **remotefile** command)

Display Bulkstats Information

To display parameters associated with the transmission of bulkstats data, enter the following command in operator exec configuration mode:

show bulkstats [**collection**]

Configuration Examples

The following example illustrates how to configure the bulk statistics feature.

Specify the primary receiver of the uploaded files:

```
[local]RedBack(config)#bulkstats mode
[local]RedBack(config-bulkstats)#receiver 10.10.20.3 primary
```

Specify the local directory, the amount of local file space allocated to bulk statistics storage, and the filename on the remote host:

```
[local]RedBack(config-bulkstats)#localdir /pcmcia0/bulkstat
[local]RedBack(config-bulkstats)#limit 2048
[local]RedBack(config-bulkstats)#remotefile format "bulkstats/%s_%s" hostname date
```

Define the header lines of the bulk statistics file and define the global schema `gbl-bulk`:

```
[local]RedBack(config-bulkstats)#header format "Collection file from host %s" hostname
[local]RedBack(config-bulkstats)#header format "Data collected on %s" date
[local]RedBack(config-bulkstats)#schema gbl-bulk format "global: %u, %s, %s" sysuptime
```

Configure a schema that collects per-circuit statistics for all circuits to which the ATM profile `ubr-bulk` is applied:

```
[local]RedBack(config)#atm profile ubr-bulk
[local]RedBack(config-atmpro)#shaping ubr
[local]RedBack(config-atmpro)#bulkstats schema atm-ubr format "uptime: %u, slot: %u,
port: %u, vpi: %u, vci: %u, inoctets: %u outoctets: %u" sysuptime slot port vpi vci
inoctets outoctets
```

Configure an ATM circuit and apply the ATM profile `ubr-bulk` to the circuit:

```
[local]RedBack(config)#port atm 3/1
[local]RedBack(config-port)#atm pvc 0 1 profile ubr-bulk encapsulation ppp
[local]RedBack(config-pvc)#bind authentication pap
```

Enable the sampling and collection of bulk statistics data:

```
[local]RedBack(config)#bulkstats collection
```

The following example creates a schema called `bart` in ATM profile configuration mode:

```
[local]RedBack(config-atmpro)#bulkstats schema bart format “atm: %u, slot:%u, port:%u,
vpi:%u, inoct:%u, outoct:%u \n” sysuptime slot port vpi vci inoctets outoctets
```

The result of this schema is a line in the data collection file that looks like:

```
bart: atm: 348765, slot:3, port:1, vpi:16, vci:233, inoct:234975, outoct:165444
```

Configuring Logging

This chapter provides an overview of and describes how to configure system event logging through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed under the "Configuration Tasks" section, see the "Logging Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

The AOS contains two log buffers: active and inactive. By default, messages are stored in the active log. If the system restarts as a result of an error, the active log is moved to the inactive log at restart. If the system is restarted normally, the inactive log is initially blank. You can move the entire contents of the active log buffer to the inactive log buffer. This is particularly useful when debugging messages cause the log buffer to fill rapidly. You can also save logs across system restarts and display the contents of logs. Log filtering allows you to isolate events from certain facilities in the logs and trim the flow of information in the system.

By default, log messages are not displayed in real time on the console. However, log messages can be displayed in real time from any Telnet session. In large installations, it is convenient to have all systems log to a remote machine for centralized management and to save space on the SMS device. The AOS uses the UNIX syslog facility for this purpose, and can send log messages to multiple machines concurrently. Logging can be constrained to events occurring on a specific circuit.

All log messages contain a numeric value indicating the severity of the event or condition that caused the message to be logged. Many log messages are normal and do not indicate a system problem.

Table 44-1 provides a list of log message severity levels and their meaning.

Table 44-1 Events Log Messages

Value	Condition	Description
0	Emergencies	Panic condition—the system is unusable.
1	alerts	Immediate administrator intervention is required.
2	critical	Critical conditions have been detected.
3	errors	An error condition has occurred.
4	warnings	A potential problem exists.
5	notifications	Normal, but significant, events or conditions exist.
6	informational	Informational messages only; no problem exists.
7	debugging	Output from an enabled system debugging function.

Configuration Tasks

To configure system event logging, perform the tasks described in the following sections:

- Filter Logging Events
- Move the Active Log Buffer to the Inactive Log Buffer
- Save Log Entries
- Enable Log Messages to Be Displayed in Real Time
- Configure Logs to Be Stored on Remote Systems
- Display Logging Information

Filter Logging Events

To filter the events that are logged on the system, enter the following command in global configuration mode:

logging filter {**console** | **monitor** | **runtime** | **syslog**} {**all** | **global** | *facility*} *level*

Move the Active Log Buffer to the Inactive Log Buffer

To move the active log buffer to the inactive log buffer, enter the following command in administrator exec configuration mode:

log checkpoint

The active log becomes the inactive log, allowing it to be examined without its data being overwritten. (The active buffer is circular in nature, in that newer messages overwrite older messages after the buffer is filled.) The previously inactive log is cleared, and then becomes the active log. Any information in the previously inactive log is lost.

Save Log Entries

To save event log buffers to the flash file system, a PCMCIA card, or to a remote FTP or TFTP server, enter the following command in administrator exec configuration mode:

save log [**active** | **inactive**] [**text**] *url* [**-noconfirm**]

The URL is the location and name of the file to which log entries are saved. The URL format varies according to where the file is stored.

Enable Log Messages to Be Displayed in Real Time

To view log messages in real time on the console, enter the following command in global configuration mode:

logging console [**circuit** {*slot/port* [{*vpi vci* | [*hdlc-channel*] *dlci*}] | **lac** *vcn* | **lns** *vcn* | **pppoe** [*cm-slot-*]*session-id*} [**only**]]

Configure Logs to Be Stored on Remote Systems

To enable event logging to a remote syslog server, enter the following command in context configuration mode:

logging syslog {*ip-address* | *hostname*} [**facility** *name*] [**circuit** {*slot/port* [*vpi vci* | [*hdlc-channel*] *dlci*] | **lac** *vcn* | **lns** *vcn* | **pppoe** [*cm-slot-*]*session-id*} [**only**]]

Display Logging Information

You can display information about system event logs and log filtering parameters by entering the following commands in operator exec configuration mode:

- To display the system event log, enter the following command:

  show log [{**active** | **inactive** | *url*} [**since** *start-time* [**until** *end-time*]] [**level** *level*] [**circuit** {*slot/port* [*vpi vci* | [*hdlc-channel*] *dlci*] | **lac** *vcn* | **lns** *vcn* | **pppoe** [*cm-slot*-]*session-id*} [**only**]]]

  The URL is the location and name of the file where entries are saved. The URL format varies according to where the file is stored.

- To display log filtering parameters, enter the following command:

  show logging [**filter** [**all** | **console** | **monitor** | **runtime** | **syslog**]]

Configuration Examples

The following example configures the system to remotely log all system messages to a network syslog server. Information to forward packets to the `10.1.1.1` address specified for the syslog host is derived from routing tables specific to the `local` context.

```
[local]RedBack#configure

Enter configuration commands, one per line, 'end' to exit
[local]RedBack(config)#context NewContext
[local]RedBack(config-ctx)#logging syslog 10.1.1.1
[local]RedBack(config-ctx)#
```

The following example shows a configuration where log messages are sent to a syslog server (`198.168.148.99`) in the `local` context using the syslog facility `local6` and to another syslog server (`198.168.145.99`) in the `green` context using the syslog facility `local3`:

```
[local]zt3(config)#context local
[local]zt3(config-ctx)#logging sys 198.168.148.99 facility local6
[local]zt3(config-ctx)#exit
[local]zt3(config)#context green
[local]zt3(config-ctx)#logging sys 198.168.145.99 facility local3
[local]zt3(config-ctx)#exit
```

The following example enables logging for events with no circuit associated and for events associated with Point-to-Point Protocol over Ethernet (PPPoE) circuit `5000`:

```
[local]RedBack(config)#context newworld
[local]RedBack(config-ctx)#logging syslog 10/1/1/1 circuit pppoe 5000
```

The following example enables logging for events for an ATM circuit with slot `5`, port `0`, VPI `140`, VCI `777`, and no others:

```
[local]RedBack(config)#context newworld
[local]RedBack(config-ctx)#logging syslog 10.1.1.1 circuit 5/0 140 777 only
```

Part 13

Network Management Services

Configuring SNMP and RMON

This chapter provides overview information about and describes the tasks used to configure Simple Network Management Protocol (SNMP) and Remote Monitoring (RMON) features supported by the Access Operating System (AOS).

For detailed information on syntax and usage guidelines for the commands described in this chapter, see the "SNMP and RMON Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuring SNMPv1 and SNMPv2c
- Configuring SNMPv3
- Configuring RMON

Overview

This section provides a brief overview of the current Simple Network Management Protocol (SNMP) management framework. For a more detailed introduction to the SNMP management framework, see RFC 2570, *Introduction to Version 3 of the Internet-standard Network Management Framework*.

The SNMP management framework has five components:

- An overall architecture, described in RFC 2571, *An Architecture for Describing SNMP Management Frameworks*.
- Mechanisms for describing and naming objects and events for the purpose of management.

  The first version, Structure of Management Information (SMIv1) is described in:

  — STD 16, RFC 1155, *Structure and Identification of Management Information for TCP/IP-based Internets*

  — STD 16, RFC 1212, *Concise MIB Definitions*

  — RFC 1215, *A Convention for Defining Traps for use with the SNMP*

The second version, SMIv2, is described in:

— STD 58, RFC 2578, *Structure of Management Information Version 2 (SMIv2)*

— STD 58, RFC 2579, *Textual Conventions for SMIv2*

— STD 58, RFC 2580, *Conformance Statements for SMIv2*

- Message protocols for transferring management information.

  — The first version, SNMPv1, is described in STD 15, RFC 1157, *A Simple Network Management Protocol (SNMP)*.

  — The second version, SNMPv2, which is not an Internet standards track protocol, is described in RFC 1901, *Introduction to Community-Based SNMPv2* and RFC 1906, *Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)*.

  — The third version, SNMPv3, is described in RFC 1906, RFC 2572, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*, and RFC 2574, *User-Based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*.

- Protocol operations for accessing management information.

  — The first set of protocol operations and associated protocol data unit (PDU) formats is described in STD 15, RFC 1157.

  — The second set of protocol operations and associated PDU formats is described in RFC 1905, *Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)*.

- A set of fundamental applications described in RFC 2573, *SNMP Applications* and the view-based access control mechanism described in RFC 2575, *View-Based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*.

Managed objects are accessed through a virtual information store, the Management Information Base (MIB). MIB objects are defined using the mechanisms set out in the SMI, see Appendix B, “Supported MIBs.”

SNMP Versions

Redback supports SNMP Version 1 (SNMPv1), Version 2c (SNMPv2c), and Version 3 (SNMPv3). All three versions of the protocol can run concurrently on the system.

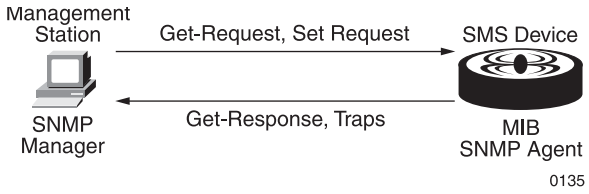
There are several differences between configuring SNMPv1 and SNMPv2c and configuring SNMPv3:

- With SNMPv1 and 2c, communities are created to control access to MIB information. You can configure these communities to meet management requirements. For instance, you can set up the automatic generation of community strings for all managed Redback contexts. This automatically creates a group with the same name as the community string.
- With SNMPv3, groups and users (instead of communities) are manually configured to control access to MIB information. Privacy and encryption options ensure a high level of configurable security.
- SNMPv3 uses Engine IDs to provide additional security.

MIBs, Traps, and Events

Designed to facilitate the exchange of management information between network devices, SNMP consists of three parts: SNMP manager, SNMP agent, and MIB. SNMP agents gather data from variables in the MIB database of the managed device. Then, agents send traps (notifications of certain events) to the SNMP manager (typically, a management station). A management station can also control a managed device by sending a message to one of the device's SNMP agents, requiring the device to change the value of one or more of its MIB variables, see Figure 45-1.

Figure 45-1 SNMP Manager, SNMP Agent, and MIB Relationship



See Appendix B, "Supported MIBs" for the list of Redback-supported MIBS, including Enterprise MIBs.

The following traps and events are supported:

- authenticationFailure, coldStart [RFC 1907]
- linkDown, linkUp [RFC 2233]
- entConfigChange [RFC 2037]
- Fan status change, power supply status change
- fallingAlarm, risingAlarm [RFC 1757]
- dsx1LineStatusChange [RFC 2495]
- dsx3LineStatusChange [RFC 2496]
- apsTrapSwitchover, apsTrapModeMismatch, apsTrapChannelMismatch

Subscriber Enterprise and Subscriber Session MIBs

Redback supports the Subscriber Enterprise MIB, which stores three sets of information pertaining to subscribers in separate tables:

- Configuration Table—Authorized administrator can create, modify and delete subscriber records maintained here.
- Active Table—provides information on active sessions per subscriber name.
- Statistics Table—provides performance data on active subscriber sessions. You can use this data to reset or clear a subscriber session.

Redback also includes support for the Subscriber Session MIB. See Appendix B, "Supported MIBs," for a comprehensive list of all supported MIBS.

The ability to clear a subscriber via SNMP is specified by the acct-session-ID RADIUS attribute. The SNMP object identifier (OID) is of the form prefix.acct-session-id.

RMON Events and Alarms

RMON features monitor specified events and generate an alarm (send an SNMP trap) when those events occur. The following features are an extension of the RMON standard and provide statistics in a standard MIB similar to the following RMON MIBs:

- RFC 1757, *Remote Network Monitoring Management Information Base*
- RFC 2021, *Remote Network Monitoring Management Information Base Version 2 using SMIv2*
- RFC 2074, *Remote Network Monitoring MIB Protocol Identifiers*

The alarm group periodically takes statistical samples from MIB variables in the managed device and compares them to previously configured thresholds. If the monitored variable crosses a threshold, an event is generated. This group consists of the alarmTable and requires the implementation of the event group. The event group controls the generation and notification of events from this device. This group consists of the eventTable and the logTable.

Configuring SNMPv1 and SNMPv2c

This section describes how to configure SNMPv1 and SNMPv2c. For information on configuring SNMPv3, see the “Configuring SNMPv3” section.

Configuration Tasks

To configure SNMPv1 or SNMPv2c, perform the tasks in the following sections:

- Enable the SNMP Server
- Configure SNMP Views
- Configure SNMP Communities
- Configure SNMP Targets and Modify Notification Parameters
- Configure an Interface IP Address as the Source for SNMP Packets
- Display SNMP Information
- Enable SNMP Debugging Messages

Enable the SNMP Server

To enable the SNMP server, enter the following command in global configuration mode:

snmp server [**port** *udp-port*]

This command enables the protocol engines for all supported versions of SNMP.

Configure SNMP Views

An SNMP view defines the set of MIB objects that can be read or modified. By default, a view named restricted is created. This view provides access to the following MIB groups: system, snmp, snmpEngine, snmpMPDStats, and usmStats.

To create additional SNMP views, enter the following command in global configuration mode:

snmp view *view-name oid-tree* {**included** | **excluded**}

Configure SNMP Communities

SNMPv1 and SNMPv2c use community strings to provide read or modify access to a MIB view. To create an SNMP community, enter the following command in global configuration mode:

snmp community *string* [[**context** *ctx-name*] | **all-contexts**] [**view** *view-name*] [**read-only** | **read-write**]

Provide for the automatic generation of community names for all managed contexts by including the **all-contexts** option. For example, if an SMS device has three configured contexts (local, aol, and uunet), the **snmp community Fred all-contexts** command creates the structured community strings (Fred@local, Fred@aol, and Fred@uunet).

In addition to generating community names, this command generates the appropriate entries in the access control tables.

Use the optional **read-only** or **read-write** keyword to allow the community to monitor, or to monitor *and* modify information.

Enforcing the **snmp community** command automatically creates a group with the same name as the community string.

Configure SNMP Targets and Modify Notification Parameters

You can configure the SMS device to send notifications (traps or informs) to management stations (SNMP targets).

Note All commands described in this section are found in global configuration mode.

To configure an SNMP target, enter one of the following two commands:

snmp target *target-name ip-address* [**port** *udp-port*] [**address-context** *name*] **security-name** *name* [**version** {**1** | **2c** | **3** [**security-level** {**auth** | **noauth**}]}] [**group** *name*] [**view** *notify-view*] [**traps** | **informs**]

snmp notify-target *notify-target-name ip-address* [**port** *udp-port*] [**address-context** *name*] **tag** *tag-list* **parameters** *target-parameters* [**timeout** *seconds*] [**retries** *count*] [**filter** *notify-filter-name*]

For SNMPv1 and SNMPv2c, these restrictions to the **snmp target** command apply:

- **security-level auth** | **no auth** option—There is no authorization provided in SNMPv1 and SNMPv2c. You must specify **noauth** for SNMPv1 and SNMPv2c.

- **group** *name* option—Specify the community name as the group name for SNMPv1 and SNMPv2c. The community name is created using the **snmp community** command. For SNMPv3, specify the group name via the **snmp group** command.
- **traps** | **informs** options—SNMPv1 supports traps only.

Note The **snmp target** and the **snmp notify-target** commands are mutually exclusive. The **snmp target** command is equivalent to the set of commands **snmp notify-target**, **snmp notify**, **snmp target-parameters**, and **snmp group** (*if* the **notify** *notify-view* parameter in the **snmp group** command has not been set), where a number of parameters are defaulted to particular values. Parameters defaulted by the **snmp target** command are notifyName, targParmName, tag, tagList, seconds, and count.

If you are configuring the SNMP target using the **snmp notify-target** command, you can use the following commands to configure notification entries, filters, and target parameters. Because these options are specified by the **snmp notify-target** command, implement these commands before configuring the SNMP target.

- To configure an SNMP notification entry, enter the following command:

  snmp notify *notify-name tag-name* {**inform** | **trap**}

- To configure an SNMP notification filter, enter the following command:

  snmp notify-filter *notify-filter-name oid-tree* {**included** | **excluded**}

- To configure notification target parameters, enter the following command:

  snmp target-parameters *parameter-name* **security-name** *name* [**version** {**1** | **2c** | **3**}] [**security-level** {**auth** | **noauth**}]

Configure an Interface IP Address as the Source for SNMP Packets

To configure the interface's primary IP address as the source address for all SNMP trap packets that are sent from the context, enter the following command in interface configuration mode using the **snmp** keyword:

ip source-address {**snmp** [**radius**] | **radius** [**snmp**]}

Note This command is also described in Chapter 7, "Configuring Interfaces."

Display SNMP Information

To display SNMP server status, statistics, and error information, enter the following command in operator exec configuration mode:

show snmp server

To display SNMP statistics, including usage, configured contexts, communities, notifications, SNMP daemon status, targets, and views, enter the following command in administrator exec configuration mode:

show snmp {**accesses** | **communities** | **contexts** | **notifies** | **server** | **transports** | **views**}

Enable SNMP Debugging Messages

To enable the logging of SNMP debug messages, enter the following command in administrator exec configuration mode:

debug snmp {**packet** | **pdu**}

Redback provides output on the specified parameters and logs the messages to the system log file.

Caution Debugging can severely affect system performance. Exercise caution when enabling any debugging on production system.

Configuration Examples

In the following SNMPv2c example, the view `Inet-View` includes all objects in the Internet OID tree. The `Admin` community allows read access to the `Inet-View`. The SMS device is configured to send traps to a system named `NM-Station1` with an IP address of `198.164.190.110`.

```
[local]RedBack(config)#snmp server
[local]RedBack(config)#snmp view Inet-View internet included
[local]RedBack(config)#snmp community Admin view Inet-View read-only
[local]RedBack(config)#snmp target NM-Station1 198.164.190.110 security-name Admin
group Admin version 2c view Inet-View traps
[local]RedBack(config)#end
```

Configuring SNMPv3

This section describes how to configure SNMPv3. For information on configuring SNMPv1 and SNMPv2c, see "Configuring SNMPv1 and SNMPv2c."

Follow these guidelines to maximize security and ensure proper configuration of SNMPv3:

- Define unique engine IDs—Do not define the engine-ID value in a configuration file that will be applied to multiple systems.
- Protect configuration files—If you create configuration files that contain security information, such as authorization passwords and keys, the files should be stored on a secured system.
- Do not use saved configurations on multiple systems—SNMP security data is system-dependent. You compromise security if the same SNMP security data is assigned to multiple systems.

Redback supports the User-Based Security Model (USM) and the following applications specific to RFC 2573, *SNMP Applications*:

- Command Responder—The AOS accepts SNMP read-class and write-class requests, performs the appropriate protocol operation, and generates a response message.
- Notification Originator—The AOS monitors the system for particular events and conditions and generates notification-class messages based on these events or conditions.

The following RFCs provide detailed information on SNMPv3:

- RFC 2570, *Introduction to Version 3 of the Internet-Standard Network Management Framework*
- RFC 2571, *An Architecture for Describing SNMP Management Frameworks*
- RFC 2572, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*
- RFC 2573, *SNMP Applications*
- RFC 2574, *The User-Based Security Model for Version 3 of the Simple Network Management Protocol (SNMPv3)*
- RFC 2575, *The View-Based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*
- RFC 2576, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-Standard Network Management Framework*

Configuration Tasks

To configure SNMPv3, perform the tasks in the following sections:

- Enable the SNMP Server
- Configure the Engine ID
- Configure SNMP Views
- Configure SNMP Groups
- Configure SNMP Users
- Configure SNMP Targets and Modify Notification Parameters
- Display SNMP Information
- Enable SNMP Debugging Messages

Enable the SNMP Server

To enable the SNMP server, enter the following command in global configuration mode:

snmp server [**port** *udp-port*]

This command enables the protocol engines for all supported versions of SNMP.

Configure the Engine ID

SNMPv3 uses engine IDs to uniquely identify an SNMP engine to provide added security. By default, the AOS creates an engine-ID that consists of the Redback Network Enterprise MIB object identifier, the UDP port in use for the SNMP server, and the management station IP address.

To specify a different engine ID that can be either local or remote, enter the following command in global configuration mode:

snmp engine-id [**local** | **remote** *name*] *id-string*

Changing the engine-ID invalidates security information for all users using authentication, and requires you to reenter user and community commands.

Configure SNMP Views

An SNMP view defines the set of MIB objects that can be read or modified. By default, a view named `restricted` exists. This view provides access to the following MIB groups: system, snmp, snmpEngine, snmpMPDStats, and usmStats.

To create additional SNMP views, enter the following command in global configuration mode:

snmp view *view-name oid-tree* {**included** | **excluded**}

Configure SNMP Groups

An SNMP group defines the degree of access for the protocol version and security model.

To configure an SNMP group, enter the following command in global configuration mode:

snmp group *name* [**context** *ctx-name*] [**exact** | **prefix**] [**security-model** {**1** | **2c** | **usm** {**auth** | **noauth**}}] [**read** *read-view*] [**write** *write-view*] [**notify** *notify-view*]

A group named `initial` is automatically created if needed (for instance, if the **snmp user** command is used without specifying a group). This group uses the user security model with the **noauth** security level, and allows read access to the view `restricted`. No write view or notify view is defined.

Configure SNMP Users

To create an SNMP user and specify security-model and password or key information for that user, enter the following command in global configuration mode:

snmp user *name* [**engine** *name*] [**group** *name*] [**security-model** {**1** | **2c** | **usm** {**noauth** | {**md5** | **sha**} [{**password** *auth-pwd* [**des56** *priv-pwd*]] | **key** [**encoded base64**] *auth-key* [**des56** *des-key*]}]

Configure SNMP Targets and Modify Notification Parameters

Note All commands described in this section are found in global configuration mode.

You can configure the SMS device to send notifications (traps or informs) to management stations (SNMP targets).

To configure an SNMP target, enter one of following two commands:

snmp target *target-name ip-address* [**port** *udp-port*] [**address-context** *name*] **security-name** *name* [**version** {**1** | **2c** | **3** [**security-level** {**auth** | **noauth | priv**}]}] [**group** *name*] [**view** *notify-view*] [**traps** | **informs**]

Among the configurable options of the **snmp target** command are three security options: **auth | noauth | priv**. By adding either **auth** or **priv**, you apply SNMPv3 authPriv level support to the targeted management station.

snmp notify-target *notify-target-name ip-address* [**port** *udp-port*] [**address-context** *name*] **tag** *tag-list* **parameters** *target-parameters* [**timeout** *seconds*] [**retries** *count*] [**filter** *notify-filter-name*]

Note The **snmp target** and the **snmp notify-target** commands are mutually exclusive. The **snmp target** command is equivalent to the set of commands **snmp notify-target**, **snmp notify**, **snmp target-parameters**, and **snmp group** (*if* the **notify** *notify-view* parameter in the **snmp group** command has not been set), where a number of parameters are defaulted to particular values. Parameters defaulted by the **snmp target** command are notifyName, targParmName, tag, tagList, seconds, and count.

If you are configuring the SNMP target using the **snmp notify-target** command, you can use the following commands to configure notification entries, filters, and target parameters. Because these options are specified by the **snmp notify-target** command, implement these commands before configuring the SNMP target.

- To configure an SNMP notification entry, enter the following command:

  snmp notify *notify-name tag-name* {**inform** | **trap**}

- To configure an SNMP notification filter, enter the following command:

  snmp notify-filter *notify-filter-name oid-tree* {**included** | **excluded**}

- To configure notification target parameters, enter the following command:

  snmp target-parameters *parameter-name* **security-name** *name* [**version** {**1** | **2c** | **3**}] [**security-level** {**auth** | **noauth** | **priv**}]

This command enables the application of authorization or privacy support to the designated SNMP target.

Configure an Interface IP Address as the Source for SNMP Packets

To configure the interface's primary IP address as the source address for all SNMP trap packets that are sent from the context, enter the following command in interface configuration mode using the **snmp** keyword:

ip source-address {**snmp** [**radius**] | **radius** [**snmp**]}

Note This command is also described in Chapter 7, "Configuring Interfaces."

Display SNMP Information

To display SNMP server status, statistics, and error information, enter the following command in operator exec configuration mode:

show snmp server

To display SNMP statistics, including usage, configured contexts, communities, notifications, SNMP daemon status, targets, and views, enter the following command in administrator exec configuration mode:

show snmp {**accesses** | **communities** | **contexts** | **notifies** | **server** | **transports** | **views**}

Enable SNMP Debugging Messages

To enable the logging of debugging messages for SNMP, enter the following command in administrator exec configuration mode:

debug snmp {**packet** | **pdu**}

Issuing this command produces output regarding the specified parameters and logs the messages to the system log file.

Caution Debugging can severely affect system performance. Exercise caution when enabling any debugging on production system.

Configuration Examples

In the following example a view named `Inet-View` is configured to include all objects in the Internet MIB tree. An authenticated group named `Group4` is configured to allow read, write, and notify access to `Inet-View`. A user named `Admin` is then configured as part of `Group4`, with the authorization password `xyzzy`. The system is then configured to send inform notifications from the `Inet-View` to a system named `Nm-Station1` (IP address 10.3.4.5), excluding `rpMauNotifications`.

```
[local]RedBack(config)#snmp server
[local]RedBack(config)#snmp engine-id local AA:00:00:00:01
[local]RedBack(config)#snmp view Inet-View internet included
[local]RedBack(config)#snmp group Group4 security-model usm auth read Inet-View write
Inet-View notify Inet-View
[local]RedBack(config)#snmp user Admin group Group4 security-model usm md5 password
"xyzzy"
[local]RedBack(config)#snmp notify Notify-Inform Tag-Inform inform
[local]RedBack(config)#snmp notify-filter Filter-incInet 1.3.*.4 included
[local]RedBack(config)#snmp notify-filter Filter-NOrpMau rpMauNotifications excluded
[local]RedBack(config)#snmp target-parameters Param2 security-name ADMIN version 3
security-level auth
[local]RedBack(config)#snmp notify-target Nm-Station1 10.3.4.5 tag Inet-Informs
parameters Param2 filter Filter-norpMau
```

Configuring RMON

This section describes how to configure RMON to monitor specified events and generate an alarm (send an SNMP trap) when those events occur. You must enable and configure SNMP before you configure RMON alarms and events. See the "Configuring SNMPv1 and SNMPv2c" or "Configuring SNMPv3" section.

Configuration Tasks

To configure RMON, perform the tasks described in the following sections:

- Configuring RMON Alarms
- Configuring RMON Events

Configuring RMON Alarms

To create an RMON alarm, enter the following command in global configuration mode:

rmon alarm *index object-id interval* {**delta** | **absolute**} **rising-threshold** *value* [*event-index*] **falling-threshold** *value* [*event-index*] [**owner** *name*]

Configuring RMON Events

To create an RMON event, enter the following command in global configuration mode:

rmon event *index* [**log**] [**trap** *community*] [**owner** *name*] [**description** *text*]

Configuration Examples

In the following example, an RMON alarm is triggered if the number of forwarded IP datagrams in the context `local` exceeds 50,000 over a 60-second interval. A delta alarm examines the difference between successive samples of the `ipForwDatagrams` variable over a 60-second interval to extrapolate the packets/second number. The rising threshold is set to 3,000,000 (50,000 packets * 60 seconds). The falling threshold is set to 600,000 (10,000 packets * 60 seconds). The rising threshold event is set to RMON event `1`, which is configured to send an SNMP trap to the community `IspAdmin` in the local context. The falling event is set to RMON event `2`, which is configured to log a message.

```
[local]RedBack(config)#rmon alarm 1 ipForwDatagrams.0 60 delta rising-threshold 3000000
event 1 falling-threshold 600000 event 2 owner local
[local]RedBack(config)#rmon event 1 trap IspAdmin owner IspAdmin description “Packets
per second too high in context local”
[local]RedBack(config)#rmon event 2 log owner local description “Packets per second
returning to normal in context local”
[local]RedBack(config)#exit
```

Configuring Web Management

This chapter provides an overview of and describes the tasks involved in configuring Web management features through the Access Operating System (AOS). For detailed information on syntax and usage guidelines for the commands listed under the "Configuration Tasks" section, see the "Web Management Commands" chapter in the *Access Operating System (AOS) Command Reference* publication.

This chapter includes the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

Overview

With HTTP server capability enabled, you can view the AOS information via a web browser to monitor system parameters, hardware configurations, subscribers, ports, channels, circuits, and Asynchronous Transfer Mode (ATM) and Frame Relay profiles, and view the configuration file on the system.

Note The web browser must be Java script enabled, and we recommend using Internet Explorer version 4.x or higher, or Netscape Navigator (or Communicator) version 4.x or higher.

In addition, you can add, delete, or modify circuit information and save changes to the Redback AOS configuration.

Configuration Tasks

To configure Web management features, perform the tasks described in the following sections:

- Enable HTTP Server Capability
- Log On To the Web Management Interface
- Monitor the System

- Monitor, Add, or Modify Circuit Information
- Clear HTTP Sessions

Enable HTTP Server Capability

To enable the AOS HTTP server capability, enter the following command in global configuration mode:

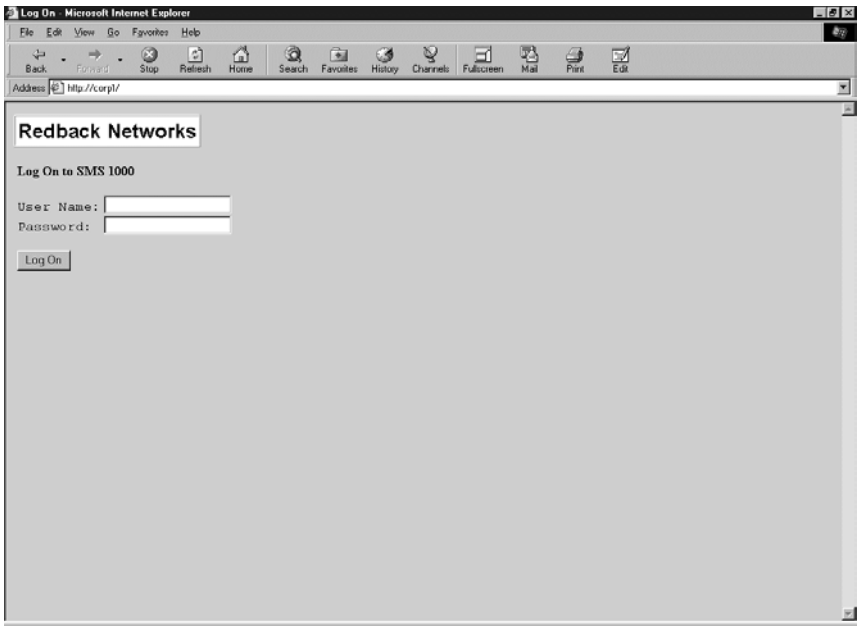
http server

Log On To the Web Management Interface

You must configure an administrator username and password to log on to the web management page. In addition, you must configure an interface for Transmission Control Protocol/Internet Protocol (TCP/IP) access to the Redback Networks system and you must bind the interface to a circuit.

To connect to the web management page, enter the URL in the form http://*ip-address* or http://*hostname*, where the IP address is the address of the interface configured for TCP/IP access and the hostname is the name of the Redback Networks device. At the prompt, log on using the administrator username and password; see Figure 46-1.

Figure 46-1 SMS 1000 Logon Window



Once the logon procedure is complete, the Redback web management main page appears. Figure 46-2 provides a sample main page for a Subscriber Management System (SMS) 1000 device. Click any module shown to display basic statistics associated with that module.

Figure 46-2 SMS 1000 Main Page

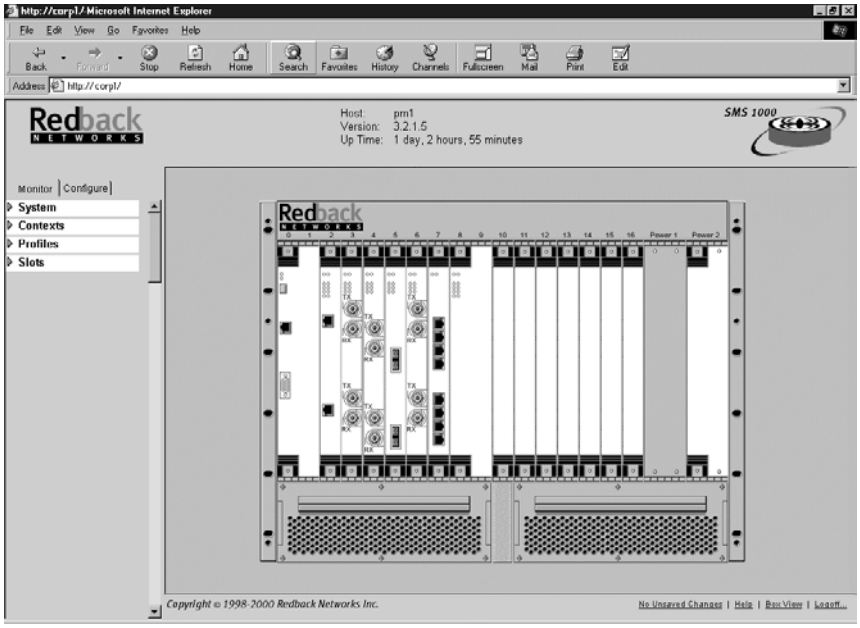
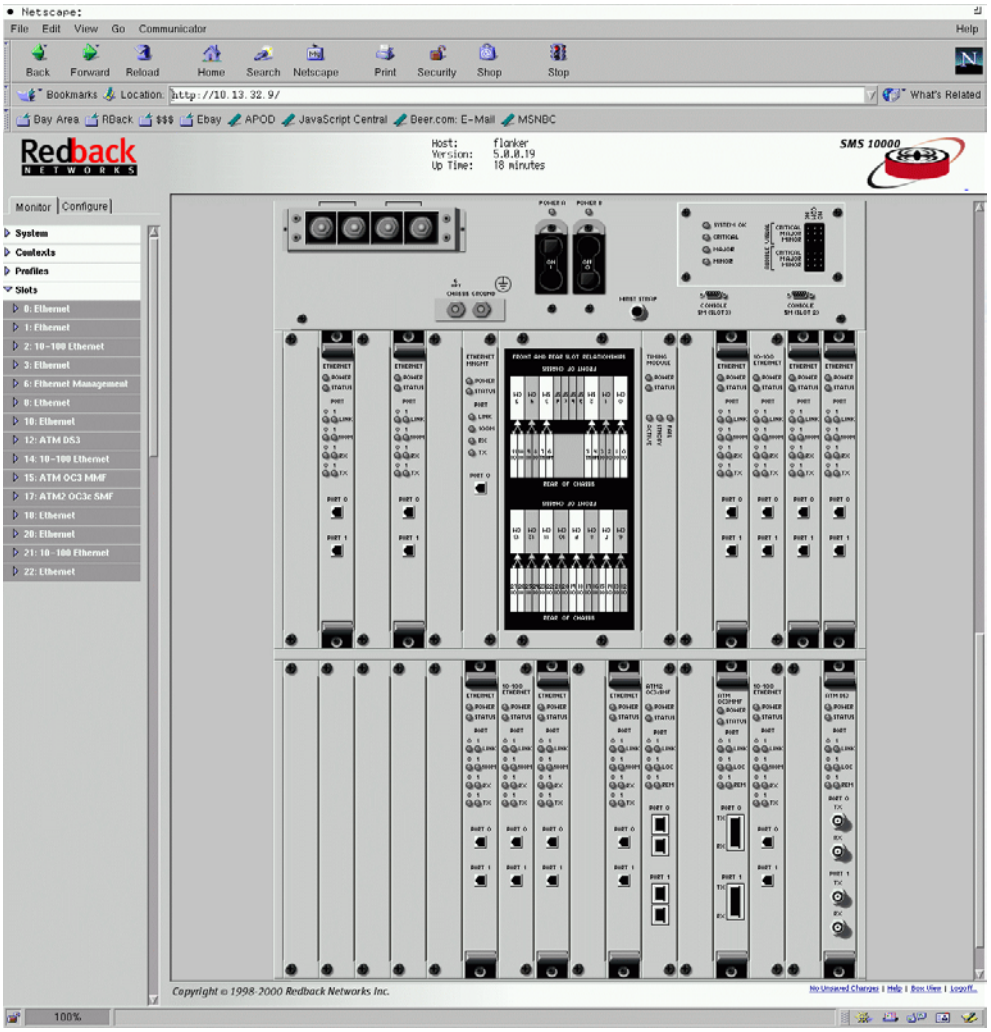


Figure 46-3 illustrates a sample SMS 10000 main page. Click any module shown to display basic statistics associated with that module.

Figure 46-3 SMS 10000 Main Page



Monitor the System

The web management interface allows you to monitor information described in the following sections. Click the **Monitor** tab to view monitoring options.

System Information

You can monitor configured administrators, the system configuration, view all hardware devices in the system, and SNMP server status. Figure 46-4 illustrates an SMS 1000 hardware display.

Figure 46-4 SMS 1000 Hardware Display

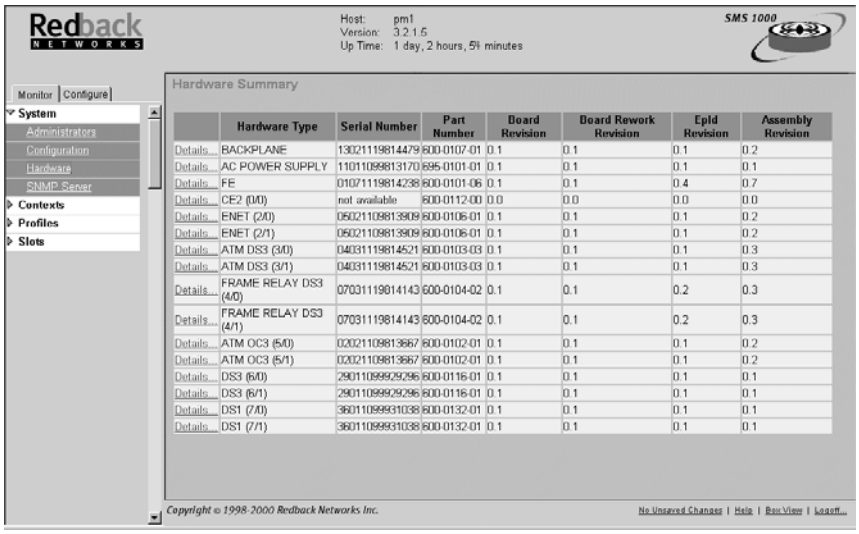


Figure 46-5 illustrates a sample SMS 10000 hardware summary display.

Figure 46-5 SMS 10000 Hardware Display

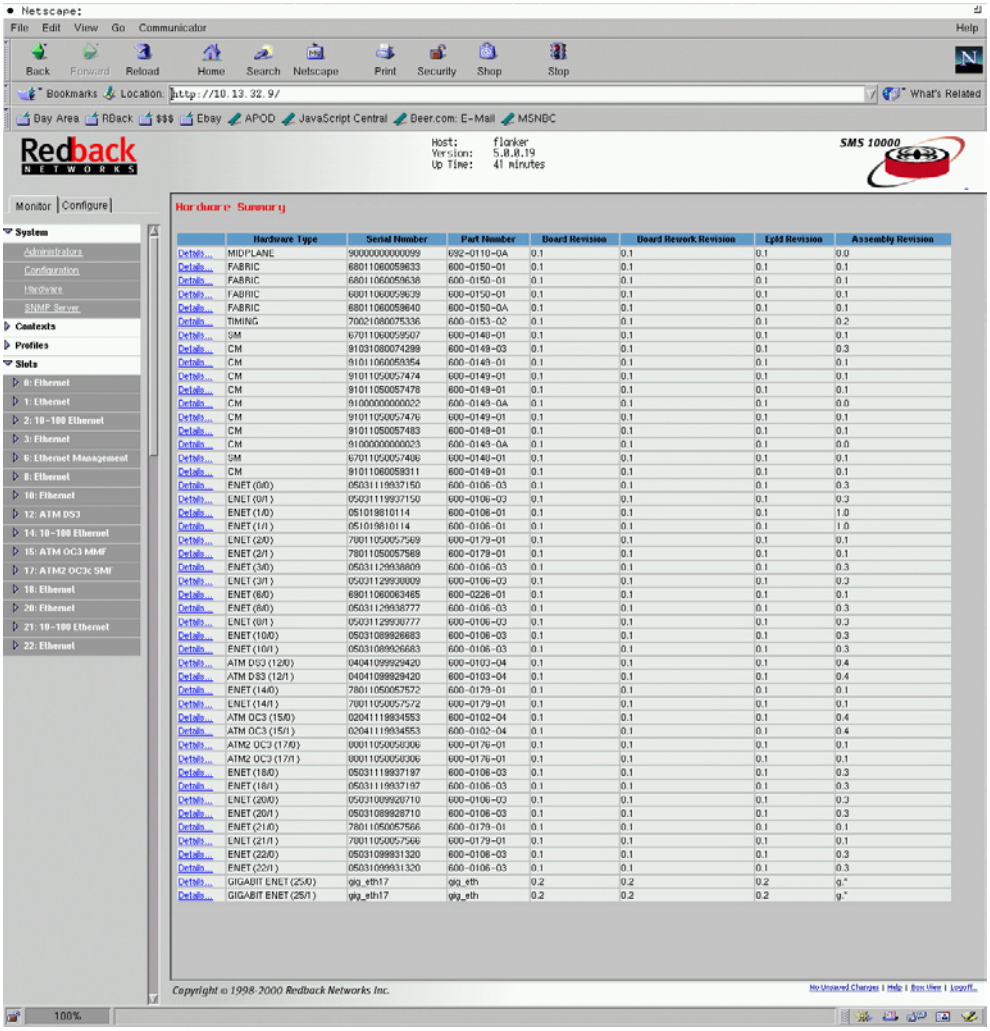


Figure 46-6 provides an example of an SMS 10000 Connection Manager (CM) module summary display.

Figure 46-6 SMS 10000 CM Summary Display

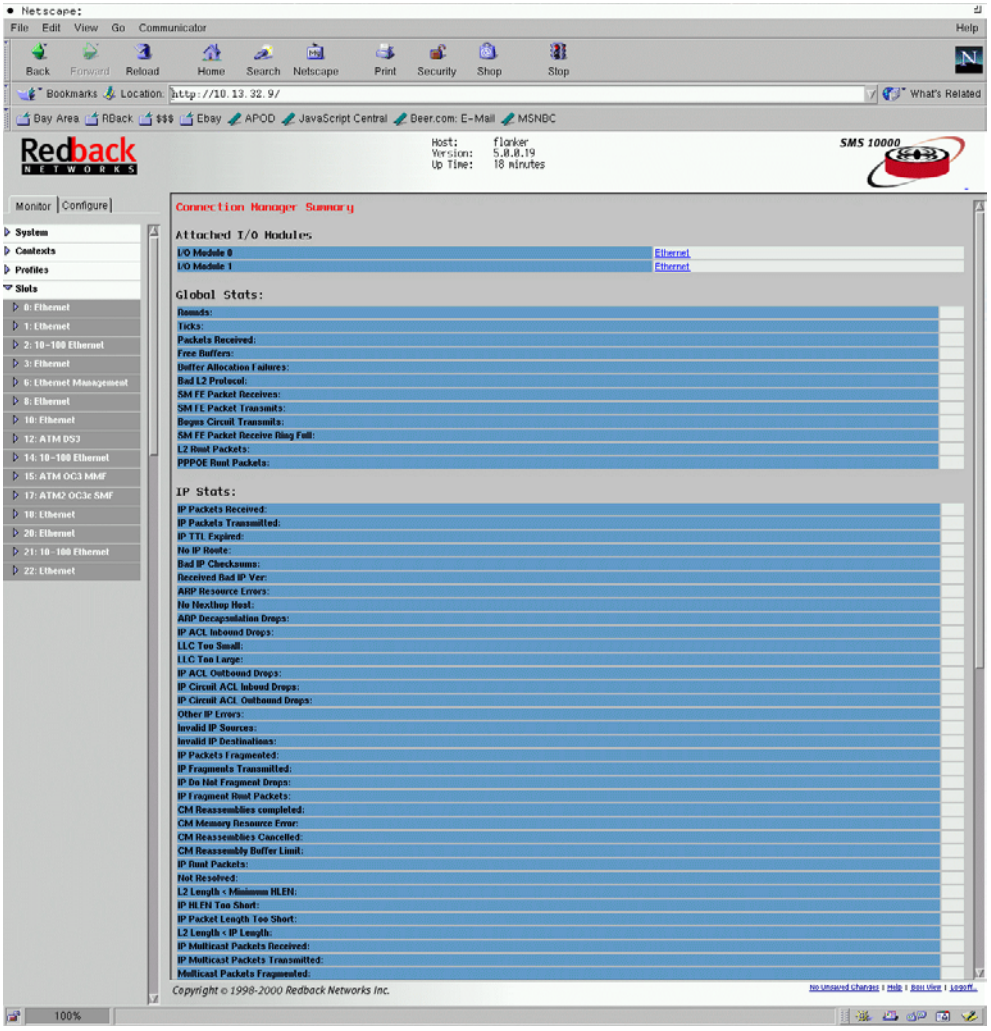
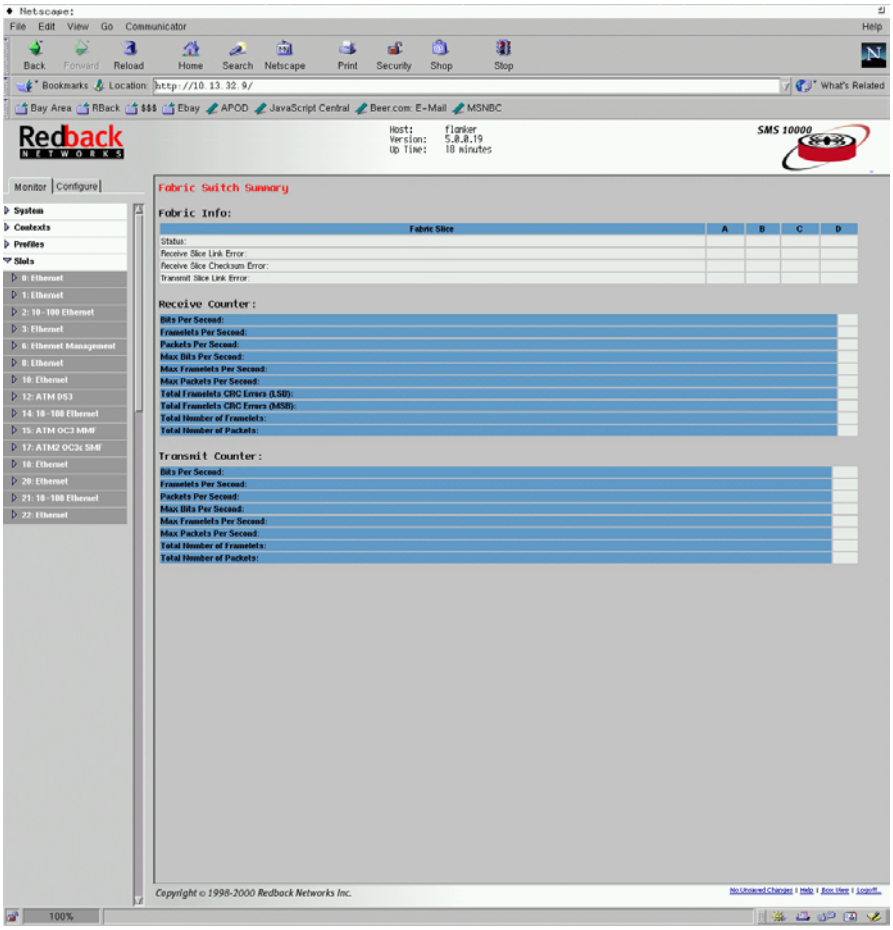


Figure 46-7 provides an example of an SMS 10000 switch fabric 42G display.

Figure 46-7 SMS 10000 Switch Fabric Display

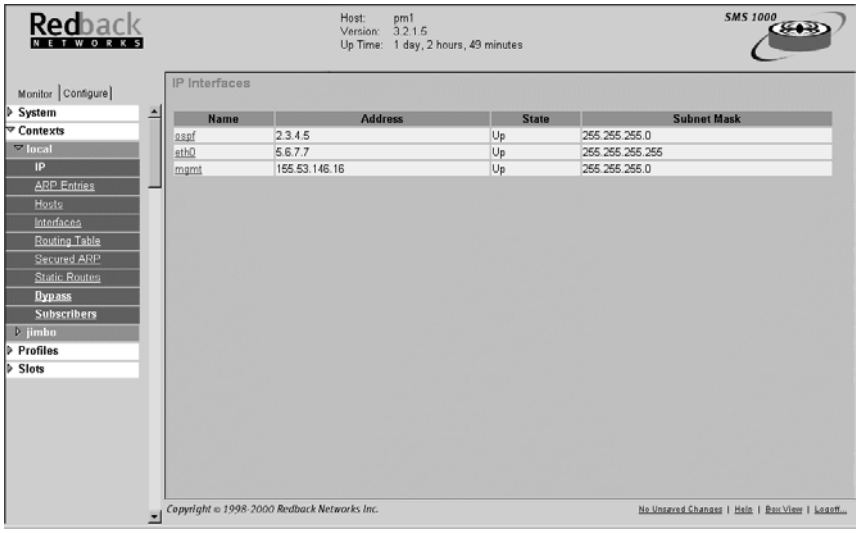


Context Information

You can view information for all configured contexts. Information includes IP Address Resolution Protocol (ARP) entries, and host, interface, and secured ARP configuration. Routing table information and static routes can also be displayed.

Figure 46-8 illustrates the interfaces configured for the local context.

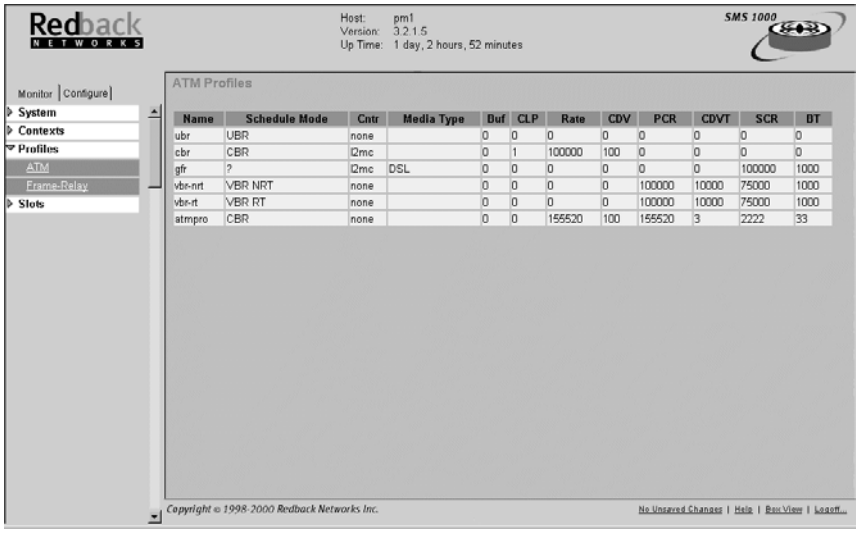
Figure 46-8 Context Interfaces



Profiles

ATM and Frame Relay profile configurations can be displayed. Figure 46-9 illustrates an ATM profile display.

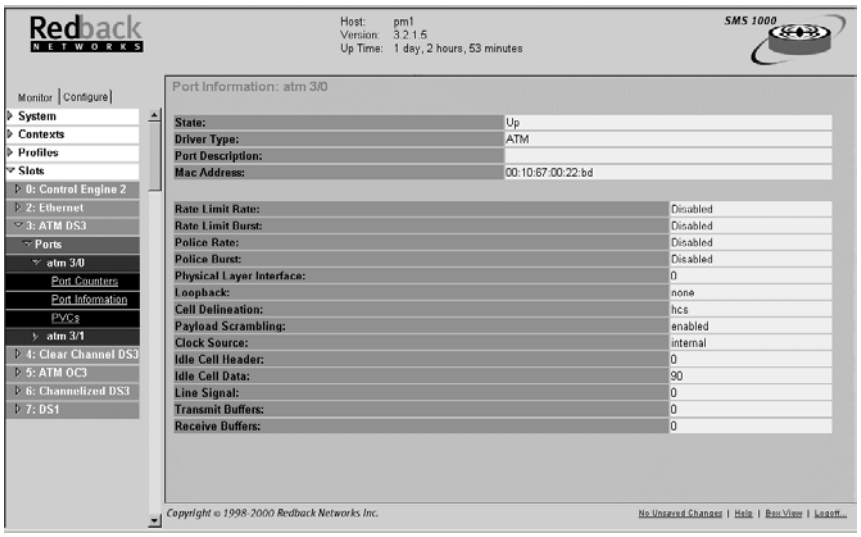
Figure 46-9 ATM Profile View



Slots

Information for each slot in the system, including the port type, slot number, port counters, circuits, channels, and so on, can be displayed. Figure 46-10 provides an example of ATM circuit information.

Figure 46-10 ATM Circuit Information View

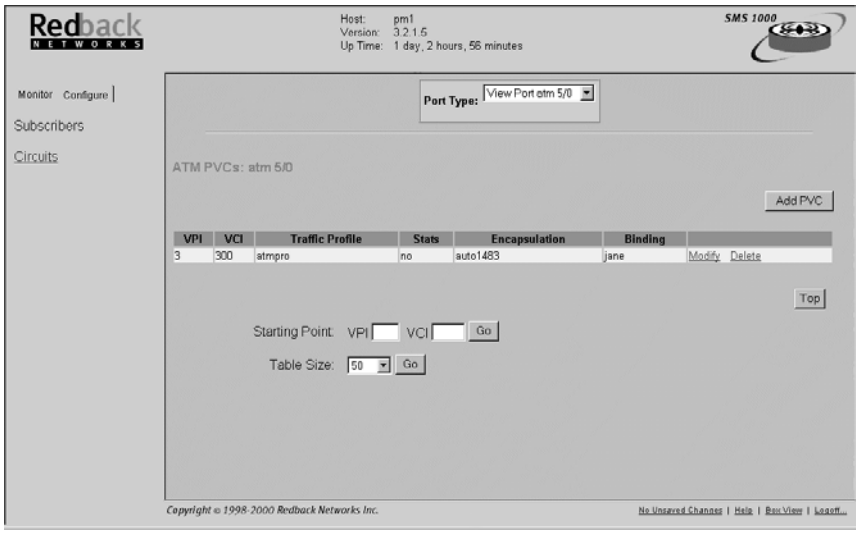


Monitor, Add, or Modify Circuit Information

You can add, modify, or delete circuits. Changes are not saved to flash memory. To save changes, use the AOS **save configuration** command in global configuration mode.

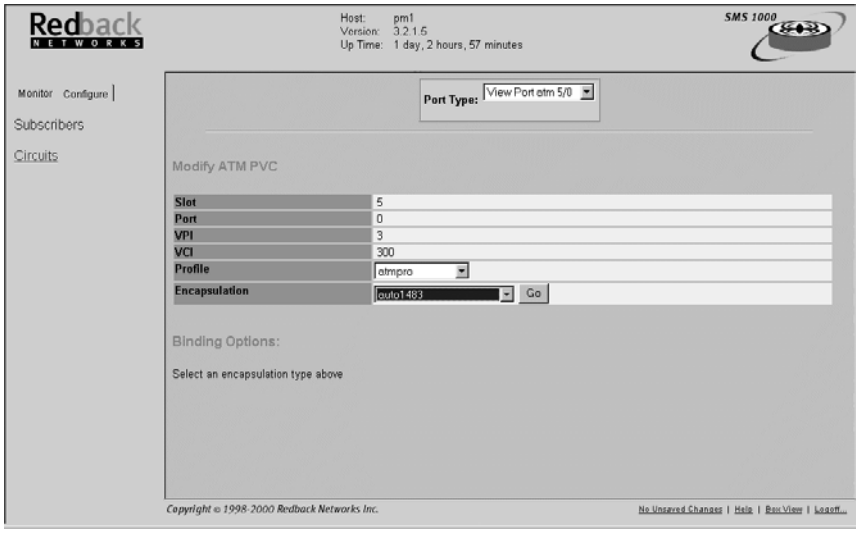
To make configuration changes, select the Configure tab. Select a port from the **Port Type** menu. See Figure 46-11. To modify the circuit associated with the port, click **Modify**. To delete it, click **Delete**. To add a circuit, click **Add PVC**.

Figure 46-11 Port Selection



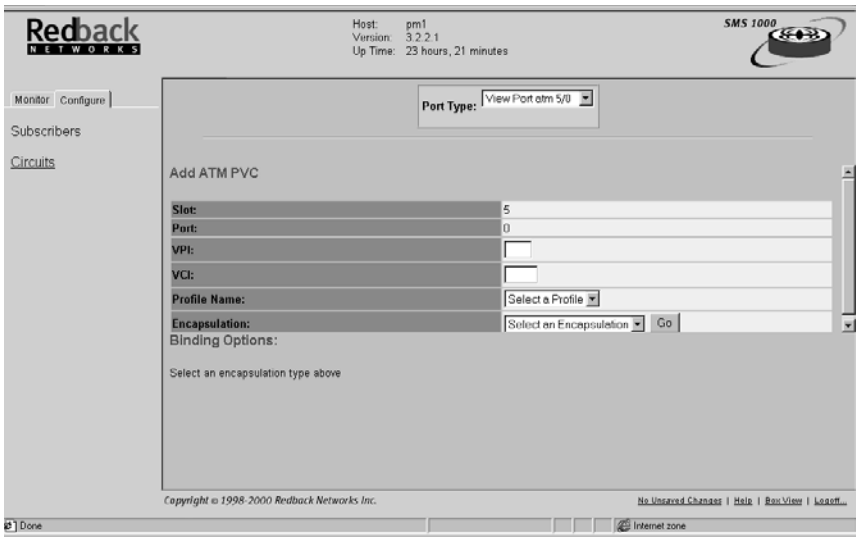
In Figure 46-12, ATM 5/0 has been selected for modification. You can change an associated profile or select a different encapsulation for the circuit.

Figure 46-12 Circuit Modification



When adding a circuit, as shown in Figure 46-13, you can enter the virtual path identifier (VPI) and virtual circuit interface (VCI) for the port, and apply an ATM profile and set the encapsulation type.

Figure 46-13 Adding a Circuit



Clear HTTP Sessions

To clear an HTTP session, enter the following command in operator exec configuration mode:

clear http *session-number*

Configuration Examples

The following example enables the SNMP server first, and then the web server:

```
[local]RedBack(config)#snmp server
[local]RedBack(config)#http server
```

Configuring NetOp Support

This chapter describes the basic tasks involved in configuring the Access Operating System (AOS) to support management of the Subscriber Management System (SMS) device via the NetOp Network Manager product.

This chapter contains the following sections:

- Overview
- Configuration Tasks
- Configuration Examples

For detailed information on syntax and usage guidelines for commands listed in "Configuration Tasks," see the "NetOp Commands" chapter in the *Access Operating System (AOS) Command Reference*.

Overview

This chapter describes the task used to configure the Netop server port on the SMS device that will be used to communicate with the NetOp Network Manager product.

Note To enable the SMS device to operate with the NetOp Network Manager, you must also enable the SNMP server, configure the network management port, and set up an administrator account on the SMS device. See the appropriate chapters in this guide to complete these tasks.

For further details on the NetOp Network Manager product, see the *NetOp Network Manager for SMS Installation Guide* and the *NetOp Network Manager for SMS Operations Guide*.

Configuration Tasks

To configure the NetOp server port on the SMS device, enter the following command in global configuration mode:

netop server *port-num*

The port number must match the port number specified in the NetOp Network Manager product.

Configuration Examples

The following example enables the NetOp server on port `2001`:

```
[local]RedBack(config)#netop server port 2001
```

Part 14

Appendixes

Configuration File Example

The following shows a sample configuration file:

```
! last updated: SUN FEB 04 06:09:52 2001
console-break-enable
boot system tftp:/155.53.198.99/redback.bin local 155.53.198.201
boot configuration /pcmcia0/redback.cfg
context local
interface e60
ip address 10.11.11.254 255.255.255.0
ip arp arpa
interface atm51
ip address 10.11.12.254 255.255.255.0
ip arp arpa
subscriber name wash2
ip address 10.11.12.253
ip route 10.11.13.0 255.255.255.0 10.11.12.253 atm51
frame-relay profile fr1
atm profile ubr
shaping ubr
port ethernet 0/0
bind interface e60 local
port atm 2/0
atm pvc 1 33 profile ubr encapsulation ppp
bind subscriber wash2@local
port channelized-ds3 3/0
shutdown
port channelized-ds3 3/1
shutdown
port ds3 6/0
shutdown
port ds3 6/1
frame-relay pvc 20 profile fr1 encapsulation ppp
framing c-bit
```

```
line console
line tty 1
line tty 2
line tty 3
line tty 4
end
```

Supported MIBs

The Access Operating System (AOS) supports the IETF standard Management Information Bases (MIBs) listed in Table B-1.

Table B-1 Standard MIBs Supported by the AOS

MIB Name	Reference Document
ATM MIB (partial)	RFC 1695, *Definitions of Managed Objects for ATM Management Version 8.0 Using SMIv2*
BGP-4 MIB	RFC 1657, *Definitions of Managed Objects for the Fourth Version of the Border Gateway Protocol (BGP-4) Using SMIv2*
DS1/E1 MIB	RFC 2495, *Definitions of Managed Objects for the DS1, E1, DS2 and E2 Interface Types*
DS3/E3 MIB	RFC 2496, *Definitions of Managed Objects for the DS3/E3 Interface Type*
Entity MIB	RFC 2037, *Entity MIB Using SMIv2*
EtherLike MIB	RFC 2358, *Definitions of Managed Objects for the Ethernet-like Interface Types*
Frame Relay DTEs MIB	RFC 2115, *Management Information Base for Frame Relay DTEs Using SMIv2*
IGMP MIB	RFC 2933, *Internet Group Management Protocol MIB*
Interfaces MIB	RFC 2233, *The Interfaces Group MIB Using SMIv2*
IP MIB	RFC 2011, *SNMPv2 Management Information Base for the Internet Protocol Using SMIv2*
IP Forwarding MIB	RFC 2096, *IP Forwarding Table MIB*
IP Tunnel MIB	RFC 2667, *IP Tunnel MIB*
L2TP MIB (subset)	draft-ietf-l2tpext-l2tp-mib-01.txt, *Layer Two Tunneling Protocol 'L2TP' Management Information Base*
MAU MIB	RFC 2668, *Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)*
OSPF MIB	RFC 1850, *OSPF Version 2 Management Information Base*

Table B-1 Standard MIBs Supported by the AOS

MIB Name	Reference Document
RMON MIB (alarms and events groups)	RFC 1757, *Remote Network Monitoring Management Information Base*
SNMPv2 MIB	RFC 1907, *Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)*
SNMP Framework MIB	RFC 2571, *An Architecture for Describing SNMP Management Frameworks*
SNMP Message Processing and Dispatching MIB	RFC 2572, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*
SNMP Target MIB, SNMP Notification MIB	RFC 2573, *SNMP Applications*
SNMPv3 User Security Model MIB	RFC 2574, *User-Based Security Model (USM) for Version 3 of the Simple Network Management Protocol (SNMPv3)*
SNMP View-Based Access Control Model MIB	RFC 2575, *View-Based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*
SNMP Community MIB	RFC 2576, *Coexistence Between Version 1, Version 2, and Version 3 of the Internet-Standard Network Management Framework*
SONET/SDH MIB	RFC 2558, *Definitions of Managed Objects for the SONET/SDH Interface Type*
TCP MIB	RFC 2012, *SNMPv2 Management Information Base for the Transmission Control Protocol Using SMIv2*
UDP MIB	RFC 2013, *SNMPv2 Management Information Base for the User Datagram Protocol Using SMIv2*

Table B-2 lists the Redback Enterprise MIBs supported by the AOS:

Table B-2 Redback Enterprise MIBs

MIB Name	Description
RBN-AAL5-VCL-STAT-MIB	Used for instrumenting statistics associated with an ATM VCL beyond those instrumented by standards-track MIBs.
RBN-APS-MIB	Supports the configuration and management of SONET linear APS groups. The definitions and descriptions used in this MIB have been derived from GR-253-CORE Revision 2, January 1999, section 5.3.
RBN-ATM-PROFILE-MIB	Used for instrumenting parameters associated with an ATM profile (traffic descriptor) beyond those instrumented by standards-track MIBs.
RBN-CPU-METER-MIB	Measures CPU utilization on a Redback Networks Control Engine (CE). Includes objects for five-second, one-minute, and five-minute averages.
RBN-ENVMON-MIB	Used to generically manage Environmental Monitor functionality on RedBack Networks devices.

Table B-2 Redback Enterprise MIBs

MIB Name	Description
RBN-PRODUCT-MIB	Contains the administrative assignments which are used to uniquely identify physical components.
RBN-PVC-MIB	The RBN-PVC-MIB Defines the objects necessary to support the creation, deletion and management of ATM and Frame Relay permanent virtual circuits (PVCs).
RBN-SMI	Defines the Structure of Management Information for RedBack Networks.
RBN-SUBSCRIBER-MIB	Defines the objects necessary to support the management of subscribers in an SMS. All MIB objects defined in the module are viewed within the context identified in the SNMP protocol (i.e. the community string in Version 1 or Version 2C, or the contextName in Version 3). This MIB module supports subscribers configured locally in an SMS (as opposed to in a RADIUS server). It also defines objects for monitoring subscribers that have active connections (i.e. are logged in), regardless of the source of the subscriber configuration attributes.
RBN-SUBSCRIBER-SESSION-MIB	Defines the objects used to manage a subscriber session identified by the Attr-Session-Id RADIUS attribute. All MIB objects defined in this MIB module are accessible regardless of context.

If you have a support contract, you can download the Redback Networks Enterprise MIBs from the Support area of the Redback Networks website at http://www.redback.com.

RADIUS Attributes

The following tables contain all Remote Authentication Dial-In User Service (RADIUS) attributes supported by the Access Operating System (AOS). An attribute is sent to RADIUS accounting only if both of the following conditions are met:

- The table contains a "Yes" in the Sent in Accounting-Request column for the attribute.
- The attribute is actually applied to the subscriber's session configuration.

Unless otherwise stated, the following values apply:

- strings: maximum of 253 characters
- integers: 32 bits
- IP addresses: 32 bits

Table C-1 contains the standard RADIUS attributes supported by the AOS, listed in order by attribute number.

Table C-1 Standard RADIUS Attributes Supported by the AOS

Num	Attribute Name	Receivable in Access-Response	Sent in Access-Request	Sent in Accting-Request	Notes
1	User-Name	No	Yes	Yes	String. Name of the user to be authenticated; only used in Access-Request packets.
2	User-Password	No	Yes	No	Sent unless using the CHAP-Password attribute.
3	CHAP-Password	No	Yes	No	Sent in Access-Request unless using the User-Password attribute.
4	NAS-IP-Address	No	Yes	Yes	IP address of the SMS device; by default, this is not sent unless explicitly enabled using the **radius attribute nas-ip-address** context command.

Table C-1 Standard RADIUS Attributes Supported by the AOS

Num	Attribute Name	Receivable in Access-Response	Sent in Access-Request	Sent in Accting-Request	Notes
5	NAS-Port	No	Yes	Yes	Format in bits as follows: SSSSSPPPCCCCCCCCCCCCCCCCCCCCCCCCCC where: S = Slot P = Port C = Circuit (for ATM, 8-bits of VPI and 16-bits of VCI) RFC 2058, *Remote Authentication Dial In User Service (RADIUS),* defines this field as 32-bits, but only 16-bits are valid. However, the SMS device cannot uniquely represent all circuits in 16-bits, so all 32 are used.
6	Service-Type	Yes	Yes	No	Integer. Type of service requested or provided. Supported values: 2 = Framed 5 = Outbound 6 = Administrative 7 = NAS Prompt
7	Framed-Protocol	Yes	Yes	Yes	Indicates the framing to be used for framed access. This attribute must not be used in a user-profile designed for RFC 1483 and RFC 1490 bridged or routed circuits, or for Telnet sessions (sent only for PPP service types). Value for PPP = 1.
8	Framed-IP-Address	Yes	Yes (depending on config.)	Yes	In Accounting Request packets, returns the IP address assigned to the subscriber either dynamically or statically. A return value of 255.255.255.254 or 0.0.0.0 causes the SMS device to assign the subscriber an address from a pool maintained locally on the SMS device.
9	Framed-IP-Netmask	Yes	No	No	Assigns a range of addresses to a subscriber circuit—it is not a netmask in the conventional sense of determining which address bits are host vs. prefix, and so on.
11	Filter-ID	Yes	No	No	Specifies that inbound or outbound traffic be filtered. Use the form **in:**<*name*> and **out:**<*name*>.
12	Framed-MTU	Yes	No	No	The MTU to be configured for the user when it is not negotiated by some other means (such as PPP). It is only used in Access-Accept packets.
18	Reply-Message	Yes	No	No	String. Text that can be displayed to the user. Multiple Reply-Messages can be included. If any are displayed, they must be displayed in the same order as they appear in the packet.

Table C-1 Standard RADIUS Attributes Supported by the AOS

Num	Attribute Name	Receivable in Access-Response	Sent in Access-Request	Sent in Accting-Request	Notes
22	Framed-Route	Yes	No	No	h.h.h.h/nn g.g.g.g where: h.h.h.h = IP address of destination host or network nn = optional netmask size in bits (if not present it defaults to 32) g.g.g.g = IP address of gateway
25	Class	Yes	No	Yes	If received, this information must be sent on, without interpretation, in all subsequent packets sent to the RADIUS server for that subscriber session.
26	Vendor-Specific	No	Yes	Yes	String. Allows Redback to support Redback VSAs. See Table C-2 for the VSAs supported by the AOS.
27	Session-Timeout	Yes	No	Yes	Sets the maximum number of seconds of service allowed the user before termination of the session. Corresponds to the AOS **timeout absolute** command in subscriber configuration mode except that the attribute calls for seconds instead of minutes.
28	Idle-Timeout	Yes	No	Yes	Sets the maximum number of consecutive seconds of idle connection allowed to the user before termination of the session. Corresponds to the AOS **timeout idle** command in subscriber configuration mode except that the attribute calls for seconds instead of minutes.
30	Called-Station-Id	No	Yes	Yes	Contains DNIS information (SHELL and L2TP only).
31	Calling-Station-Id	No	Yes	Yes	IP address of remote peer (sent only in shell service types). The SMS device has the ability to take the DNIS information provided via L2TP (when the SMS device is acting as an LNS) and put it into the Calling-Station-Id attribute.
32	NAS-Identifier	No	Yes	Yes	Value of system hostname configuration.
40	Acct-Status-Type	No	No	Yes	Values include Acct-Start, Acct-Interim, Acct-Stop.
41	Acct-Delay-Time	No	No	Yes	Time in seconds for which the client has been trying to send the record.
42	Acct-Input-Octets	No	No	Yes	The number of octets that have been received from the port over the course of this service being provided. Can only be present in Accounting-Request records where the Acct-Status-Type is set to Stop or Update.
43	Acct-Output-Octets	No	No	Yes	The number of octets that have been sent to the port in the course of delivering this service. Can only be present in Accounting-Request records where the Acct-Status-Type is set to Stop or Update.

Table C-1 Standard RADIUS Attributes Supported by the AOS

Num	Attribute Name	Receivable in Access-Response	Sent in Access-Request	Sent in Accting-Request	Notes
44	Acct-Session-Id	No	Yes (depending on config.)	Yes	A unique accounting ID to easily match start and stop records in a log file. The start and stop records for a given session must have the same Acct-Session-Id attribute value. It is strongly recommended that the Acct-Session-Id attribute be a printable ASCII string.
45	Acct-Authentic	No	No	Yes	Values are RADIUS and local.
46	Acct-Session-Time	No	No	Yes	The number of seconds for which the user has received service. Can only be present in Accounting-Request records where the Acct-Status-Type attribute is set to Stop.
47	Acct-Input-Packets	No	No	Yes	The number of packets that have been received from the port over the course of this service being provided to a framed user. Can only be present in Accounting-Request records where the Acct-Status-Type attribute is set to Stop or Update.
48	Acct-Output-Packets	No	No	Yes	The number of packets that have been sent to the port in the course of delivering this service to a Framed User. Can only be present in Accounting-Request records where the Acct-Status-Type attribute is set to Stop or Update.
50	Acct-Multi-Session-Id	No	No	Yes	Contains the PPP multilink session ID value that is used to associate multiple PPP sessions to a multilink session.
51	Acct-Link-Count	No	No	Yes	Contains the current number of links in a multilink bundle.
61	NAS-Port-Type	No	Yes	Yes	The type of the physical port of the NAS that authenticates the user. It can be used instead of or in addition to the NAS-Port attribute. Either the NAS-Port attribute or the NAS-Port-Type attribute, or both should be present in an Access-Request packet, if the NAS differentiates among its ports. Sent for Shell service types only. Possible values: 0 = Async (console) 5 = Virtual (connection to NAS via some transport protocol rather than physical port)
62	Port-Limit	Yes	No	Yes	Controls the maximum number of sessions a particular subscriber can have active at one time.
64	Tunnel-Type	Yes	No	Yes	Tunneling protocols to be used. Supported protocols are L2TP and GRE. Required attribute for both L2TP and GRE.

Table C-1 Standard RADIUS Attributes Supported by the AOS

Num	Attribute Name	Receivable in Access-Response	Sent in Access-Request	Sent in Accting-Request	Notes
65	Tunnel-Medium-Type	Yes	No	Yes	Transport medium to use when creating a tunnel for those protocols that can operate over multiple transports. Required attribute for both L2TP and GRE. Supported values: 1 = IP (IP version 4) 8 = E.164 (SMDS, Frame Relay, ATM) For GRE, the value must always be set to 1 (IP).
66	Tunnel-Client-Endpoint	Yes	No	Yes	String. Address of the initiator end of the tunnel. The string field follows the tag field. Used for L2TP and GRE. Required attribute for GRE.
67	Tunnel-Server-Endpoint	Yes	No	Yes	String. Address of the server end of the tunnel. Required attribute for L2TP (except L2TP PVC media tunnels) and for GRE. Format of the string depends on the Tunnel-Medium-Type attribute. If the Tunnel-Medium-Type attribute value is IP (1), then this string is either the fully qualified domain name of the tunnel client machine, or it is a dotted-decimal IP address. For GRE, the Tunnel-Medium-Type attribute must always be set to 1. If the Tunnel-Medium-Type attribute value is not IP, this string is a tag referring to configuration data local to the RADIUS client that describes the interface and medium-specific address to use.
68	Acct-Tunnel-Connection	No	No	Yes	A unique accounting ID to easily match start and stop records in a log file, used only for L2TP sessions. The start and stop records for a given session must have the same Acct-Tunnel-Connection attribute value. It is strongly recommended that the Acct-Tunnel-Connection attribute be a printable ASCII string.
69	Tunnel-Password	Yes	No	No	String. The string field follows the tag and salt fields. Only used in Access-Accept packets.
77	Connect-Info	No	Yes	Yes	String containing an ATM or Frame-Relay profile name being sent to the RADIUS server.
82	Tunnel-Assignment-Id	Yes	No	Yes	String. The tunnel to which the session is to be applied. The string field follows the tag field. Required attribute for L2TP one-pass.
83	Tunnel-Preference	Yes	No	Yes	If more than one set of tunneling attributes is returned by the RADIUS server to the tunnel initiator, this attribute should be included in all sets to indicate the relative preference assigned to each.

Table C-1 Standard RADIUS Attributes Supported by the AOS

Num	Attribute Name	Receivable in Access-Response	Sent in Access-Request	Sent in Accting-Request	Notes
90	Tunnel-Client-Auth-Id	Yes	No	Yes	String. Defines the local hostname provided to remote tunnel-peer (used during tunnel-setup). Behavior is identical to Redback VSA 16, Tunnel-Local-Name attribute.
91	Tunnel-Server-Auth-Id	Yes	No	Yes	String. Defines an alias for the remote peer name. Behavior is identical to Redback VSA 17, Tunnel-Remote-Name attribute.
242	Ascend-Data-Filter	Yes	No	No	String containing a subscriber level access control list, downloadable from the RADIUS server (specific to Ascend).

Redback VSAs are embedded according to the procedure recommended in RFC 2138, *Remote Authentication Dial-In User Service*, with the Vendor-ID attribute set to 2352. Table C-2 lists the Redback VSAs supported by the AOS, in order by attribute number.

Table C-2 Redback Networks VSAs Supported by AOS

Num	Attribute Name	Receivable in Access-Response	Sent in Access-Request	Sent in Accting-Request	Notes
1	Client-DNS-Pri	Yes	No	No	IP address of the primary DNS server for this user's connection.
2	Client-DNS-Sec	Yes	No	No	IP address of the secondary DNS server for this user's connection.
3	DHCP-Max-Leases	Yes	No	No	4-byte integer. Maximum number of DHCP addresses this user can allocate.
4	Context-Name	Yes	No	No	Binds user's session to specified context, overriding the structured username—only interpreted when global AAA is enabled.
5	Bridge-Group	Yes	No	No	*bridge-group-name*; attaches subscriber to the named bridge-group.
6	BG-Aging-Time	Yes	No	No	*bg-name*:*val*; configures bridge aging time for subscriber attaching to the named bridge-group.
7	BG-Path-Cost	Yes	No	No	*bg-name*:*val*; configures bridge path cost for subscriber attaching to the named bridge-group.
8	BG-Span-Dis	Yes	No	No	*bg-name*:*val*; disables spanning tree for subscriber attaching to the named bridge-group. The *val* argument can have the following values: 1 = TRUE 2 = FALSE

Table C-2 Redback Networks VSAs Supported by AOS

Num	Attribute Name	Receivable in Access-Response	Sent in Access-Request	Sent in Accting-Request	Notes
9	BG-Trans-BPDU	Yes	No	No	*bg-name*:*val*; sends transparent Spanning Tree BPDUs for subscriber attaching to the named bridge-group. The *val* argument can have the following values: 1 = TRUE 2 = FALSE
10	Rate-Limit-Rate	Yes	No	No	4-byte integer. Configures rate-limit rate for subscriber in kbps.
11	Rate-Limit-Burst	Yes	No	No	4-byte integer. Configures rate-limit burst rate for subscriber in bytes.
12	Police-Rate	Yes	No	No	4-byte integer. Configures policing rate for subscriber in kbps.
13	Police-Burst	Yes	No	No	4-byte integer. Configures policing burst rate for subscriber in bytes.
14	Source-Validation	Yes	No	No	4-byte integer. Enables source validation for subscriber. 1 = TRUE 0 = FALSE
15	Tunnel-Domain	Yes	No	Yes	Integer. Binds the subscriber to a tunnel based on the domain name portion of the username. 1 = TRUE 0 = FALSE
16	Tunnel-Local-Name	Yes	No	Yes	String. Defines the local hostname provided to the remote tunnel peer (used during tunnel setup).
17	Tunnel-Remote-Name	Yes	No	Yes	String. Defines an alias for the remote peer name.
18	Tunnel-Function	Yes	No	Yes	Integer. Determines whether this tunnel configuration can only be used as a LAC-only endpoint, an LNS-only endpoint, or both. 1 = LAC only 2 = LNS only 3 = LAC/LNS
21	Tunnel-Max-Sessions	Yes	No	Yes	Integer. Limits the number of sessions per tunnel using this tunnel configuration.
22	Tunnel-Max-Tunnels	Yes	No	Yes	Integer. Limits the number of tunnels that can be initiated using this tunnel configuration (does not apply to PVC-based tunnels).
23	Tunnel-Session-Auth	Yes	No	Yes	Integer. Specifies the authentication method to use during tunnel authentication. 1 = CHAP 2 = PAP 3 = CHAP-PAP
24	Tunnel-Window	Yes	No	Yes	Integer. Configures window size for incoming L2TP messages.

Table C-2 Redback Networks VSAs Supported by AOS

Num	Attribute Name	Receivable in Access-Response	Sent in Access-Request	Sent in Accting-Request	Notes
25	Tunnel-Retransmit	Yes	No	Yes	Integer. Number of times the SMS device retransmits a control message.
26	Tunnel-Cmd-Timeout	Yes	No	Yes	Integer. Number in seconds. Configures timeout between control message retransmissions.
27	PPPOE-URL	Yes	No	Yes	String in PPPoE URL format. Defines the PPPoE URL that is sent to the remote PPPoE client via the PADM packet.
28	PPPOE-MOTM	Yes	No	Yes	String. Defines the PPPoE MOTM message that is sent to the remote PPPoE client via the PADM packet.
29	Tunnel-Group	Yes	No	Yes	Integer. Indicates whether this record is or is not a tunnel group with a list of member peers. 1 = TRUE 0 = FALSE
30	Tunnel-Context	Yes	No	Yes	String. Context name. Used in a DNIS peer record, this attribute specifies the context where the named peer should be found.
31	Tunnel-Algorithm	Yes	No	Yes	Integer. Specifies the session distribution algorithm used for the tunnel group. Applies to both tagged and nontagged tunnel groups. 1 = First 2 = Load-Balance
32	Tunnel-Deadtime	Yes	No	Yes	Integer. Number of minutes during which no sessions are attempted to a peer once the peer is declared dead.
33	Mcast-Send	Yes	No	Yes	Integer. Defines whether or not the subscriber can send multicast packets. 1 = NO SEND 2 = SEND 3 = UNSOLICITED SEND
34	Mcast-Receive	Yes	No	Yes	Integer. Defines whether or not the subscriber can receive multicast packets. 1 = NO RECEIVE 2 = RECEIVE
35	Mcast-MaxGroups	Yes	No	Yes	Integer. Specifies the maximum number of multicast groups of which the subscriber can be a member.
36	Ip-Address-Pool-Name	Yes	No	Yes	String. Name of the interface used to assign an IP-Pool address to the subscriber.
37	Tunnel-DNIS	Yes	No	Yes	Integer. L2TP peer parameter specifying if incoming sessions from this peer are to be switched based on the incoming DNIS AVP if present or on the incoming DNIS AVP only (terminated if no DNIS AVP is present). 1 = DNIS 2 = DNIS ONLY

Table C-2 Redback Networks VSAs Supported by AOS

Num	Attribute Name	Receivable in Access-Response	Sent in Access-Request	Sent in Accting-Request	Notes
38	Medium-Type	No	Yes	Yes	Integer. Contains the medium type of the circuit as configured by the administrator in the ATM profile, Frame Relay profile, or the Ethernet port configuration. 11 = DSL 13 = Wireless 14 = Satellite
39	PVC-Encapsulation-Type	Yes	No	No	Integer. Encapsulation type to be applied to the circuit. Also specifies the encapsulation for Ethernet over L2TP sessions on the LNS. Only 22 and 23 of the following encapsulations are valid for Ethernet over L2TP sessions. 2 = Routed 1483 3 = auto 1483 4 = ATM multi 5 = Bridged 1483 6 = ATM PPP 7 = ATM PPP serial 8 = ATM PPP NLPID 9 = ATM PPP auto 10 = ATM PPPoE 11 = ATM L2TP 12 = ATM PPP LLC 13 = auto 1490 14 = Frame Relay multi 15 = Bridged 1490 16 = Frame Relay PPP 17 = Frame Relay PPP auto 18 = Frame Relay PPPoE 19 = Routed 1490 20 = Frame Relay L2TP 21 = L2TP VC muxed 22 = Ethernet 23 = Ethernet over PPPoE 24 = Ethernet multi
40	PVC-Profile-Name	Yes	No	No	String. Shaping profile name.
42	Bind-Type	Yes	No	No	Integer. Binding type to be applied to this circuit. Also specifies the same for Ethernet over L2TP sessions on the LNS. Only 1 and 3 of the following binding types are valid for the Ethernet over L2TP mode. 1 = authentication 2 = bypass 3 = interface 4 = subscriber 5 = tunnel 6 = session 7 = q8021 8 = multi
43	Bind-Auth-Protocol	Yes	No	No	Integer. Authentication protocol to use for this circuit. Also specifies the same for PPPoE sessions tunneled with Ethernet encapsulation over L2TP on the LNS. Only 1, 2, and 4 of the following protocols are valid for the Ethernet over L2TP mode. 1 = PAP 2 = CHAP 3 = CHAP wait 4 = CHAP PAP 5 = CHAP PAP wait

Table C-2 Redback Networks VSAs Supported by AOS

Num	Attribute Name	Receivable in Access-Response	Sent in Access-Request	Sent in Accting-Request	Notes
44	Bind-Auth-Max-Sessions	Yes	No	No	Integer. Maximum number of PPPoE sessions allowed to be created for this circuit. Also specifies the same for PPPoE sessions tunneled with Ethernet encapsulation over L2TP on the LNS.
45	Bind-Bypass-Bypass	Yes	No	No	String. Name of the bypass being bound.
46	Bind-Auth-Context	Yes	No	No	String. Bind authentication context name. Also specifies the same for PPPoE sessions tunneled with Ethernet encapsulation over L2TP on the LNS.
47	Bind-Auth-Service-Grp	Yes	No	No	String. Bind authentication service group name. Also specifies the same for PPPoE sessions tunneled with Ethernet encapsulation over L2TP on the LNS.
48	Bind-Bypass-Context	Yes	No	No	String. Bind bypass context name.
49	Bind-Int-Context	Yes	No	No	String. Bind interface context name. Also specifies the same for IP bridging sessions tunneled with Ethernet encapsulation over L2TP on the LNS.
50	Bind-Tun-Context	Yes	No	No	String. Bind tunnel context name.
51	Bind-Ses-Context	Yes	No	No	String. Bind session context name.
52	Bind-Dot1q-Slot	Yes	No	No	Integer. Bind 802.1Q slot number.
53	Bind-Dot1q-Port	Yes	No	No	Integer. Bind 802.1Q port number.
54	Bind-Dot1q-Vlan-Tag-Id	Yes	No	No	Integer. Bind 802.1Q VLAN tag ID.
55	Bind-Int-Interface-Name	Yes	No	No	String. Bind interface name. Also specifies the same for IP bridging sessions tunneled with Ethernet encapsulation over L2TP on the LNS.
56	Bind-L2TP-Tunnel-Name	Yes	No	No	String. Bind L2TP tunnel name.
57	Bind-L2TP-Flow-Control	Yes	No	No	Integer. Bind L2TP flow control.
58	Bind-Sub-User-At-Context	Yes	No	No	String. Bind subscriber context name.
59	Bind-Sub-Password	Yes	No	No	String. Bind subscriber password.
60	Ip-Host-Addr	Yes	No	No	String in the form A.B.C.D hh:hh:hh:hh:hh:hh. IP host address and MAC address. A space must separate the IP address from the MAC address.

Table C-2 Redback Networks VSAs Supported by AOS

Num	Attribute Name	Receivable in Access-Response	Sent in Access-Request	Sent in Accting-Request	Notes
61	IP-Tos	Yes	No	No	Integer. Specifies the value of the IP ToS field. Used for soft QoS. 0 = normal 1 = min-cost only 2 = max-reliability only 3 = max-reliability plus min-cost 4 = max-throughput only 5 = max-throughput plus min-cost 6 = max-throughput plus max-reliability 7 = max-throughput plus max-reliability plus min-cost 8 = min-delay only 9 = min-delay plus min-cost 10 = min-delay plus max-reliability 11 = min-delay plus max-reliability plus min-cost 12 = min-delay plus max-throughput 13 = min-delay plus max-throughput plus min-cost 14 = min-delay plus max-throughput plus max-reliability 15 = min-delay plus max-throughput plus max-reliability plus min-cost
62	NAS-Real-Port	No	Yes	Yes	Integer. Indicates the port number of the physical circuit on which the session was received. Format in bits is as follows: SSSSSPPPCCCCCCCCCCCCCCCCCCCCCCC where: S = Slot P = Port C = Circuit (for ATM, 8-bits of VPI, and 16-bits of VCI) RFC 2058, *Remote Authentication Dial In User Service (RADIUS),* defines this field as 32 bits, with only 16 bits being valid. However, the SMS device cannot uniquely represent all circuits in 16 bits, so all 32 bits are used.
63	Tunnel-Session-Auth-Ctx	Yes	No	Yes	String. L2TP peer parameter specifying the context in which all incoming PPP over L2TP sessions should be authenticated, regardless of the domain specified in the username.
64	Tunnel-Session-Auth-Service-Grp	Yes	No	Yes	String. L2TP peer parameter specifying the service group (service access list) to be used for all incoming PPP over L2TP sessions.
65	Tunnel-Rate-Limit-Rate	Yes	No	Yes	4-byte integer. L2TP or GRE peer parameter specifying the rate-limit rate for a tunnel in kbps.
66	Tunnel-Rate-Limit-Burst	Yes	No	Yes	4-byte integer. L2TP or GRE peer parameter specifying the rate-limit burst for a tunnel in kbps.
67	Tunnel-Police-Rate	Yes	No	Yes	4-byte integer. L2TP or GRE peer parameter specifying the policing rate for a tunnel in kbps.

Table C-2 Redback Networks VSAs Supported by AOS

Num	Attribute Name	Receivable in Access-Response	Sent in Access-Request	Sent in Accting-Request	Notes
68	Tunnel-Police-Burst	Yes	No	Yes	4-byte integer. L2TP or GRE peer parameter specifying the policing burst for a tunnel in kbps.
69	Tunnel-L2F-Second-Password	Yes	No	Yes	String. L2F peer parameter specifying the password string used to authenticate the L2F remote peer. Note that the Tunnel-Password attribute is used for authentication in the other direction.
70	ACL-Definition	Yes	No	Yes	String. Used to define ACL definitions in the RADIUS database. The ACL-Name attribute is the username and the Service-Type attribute must be set to Access-Control-List. The data content of this attribute contains ACL definitions similar to the AOS CLI.
71	PPPoE-IP-Route-Add	Yes	No	Yes	String. Allows PPPoE subscribers routing table to be populated in terms of what routes to be installed in multiple PPPoE sessions exist. A more granular set of routes can be achieved when multiple sessions are active to the client.
72	TTY-Level-Max	Yes	No	Yes	Integer. Range of values is 0 to 15. Must be greater than or equal to the value of TTY-Level-Start. Corresponds to the **privilege max** command and supports privilege levels.
73	TTY-Level-Start	Yes	No	Yes	Integer. Range of values is 0 to 15. Must be less than or equal to the value of TTY-Level-Max. Corresponds to the **privilege start** command and supports privilege levels.
74	Tunnel-Checksum	Yes	No	Yes	Integer. Enables GRE checksums. When enabled, a checksum is computed for each outgoing GRE packet. This allows the remote system to verify the integrity of each packet. Incoming packets that fail the checksum are discarded. A value of 1 = enabled. Any other value for this attribute = disabled.
75	Tunnel-Profile	Yes	No	No	String. Attaches a profile to the tunnel. Used when configuring a tunnel from a RADIUS server. A Tunnel-Profile attribute in a subscriber record is ignored.
78	Tunnel-Client-VPN	Yes	No	Yes	String. Name of the target context (VPN) on the client side of the tunnel. Required for GRE. If omitted, the system automatically sets the value equal to the value set for the Tunnel-Server-VPN attribute.
79	Tunnel-Server-VPN	Yes	No	Yes	String. Name of the target context (VPN) on the server side of the tunnel.

Table C-2 Redback Networks VSAs Supported by AOS

Num	Attribute Name	Receivable in Access-Response	Sent in Access-Request	Sent in Accting-Request	Notes
80	Tunnel-Client-Rhost	Yes	No	Yes	String. Normally configured in the **ip host** command on the client system. If omitted, the system uses the value of the Tunnel-Client-Int-Addr attribute on the server side.
81	Tunnel-Server-Rhost	Yes	No	Yes	String. Normally configured in the **ip host** command on the server system. If omitted, the system uses the value of the Tunnel-Server-Int-Addr attribute on the client side.
82	Tunnel-Client-Int-Addr	Yes	No	Yes	IP address of the interface to bind in the VPN context. This address is also used in the **ip host** statement on the server system. Required attribute for GRE.
83	Tunnel-Server-Int-Addr	Yes	No	Yes	IP address of the server interface. This address is also used in the **ip host** statement on the client system. Required attribute for GRE.
128	Acct-Input-Octets-64	No	No	Yes	64-bit value for the Acct-Input-Octets standard attribute.
129	Acct-Output-Octets-64	No	No	Yes	64-bit value for the Acct-Output-Octets standard attribute.
130	Acct-Input-Packets-64	No	No	Yes	64-bit value for the Acct-Input-Packets standard attribute.
131	Acct-Output-Packets-64	No	No	Yes	64-bit value for the Acct-Output-Packets attribute.
132	Assigned-IP-Address	No	No	Yes	IP address. Reports IP addresses assigned to a subscriber via IP pools or DHCP.
133	Acct-Mcast-In-Octets	No	No	Yes	Integer. Reports the number of inbound multicast octets received on this circuit.
134	Acct-Mcast-Out-Octets	No	No	Yes	Integer. Reports the number of outbound multicast octets received on this circuit.
135	Acct-Mcast-In-Packets	No	No	Yes	Integer. Reports the number of inbound multicast packets received on this circuit.
136	Acct-Mcast-Out-Packets	No	No	Yes	Integer. Reports the number of outbound multicast packets received on this circuit.
137	LAC-Port	No	Yes	Yes	Integer. Contains the CCT handle for the incoming session on an L2TP LAC. This attribute should be present for a subscriber on an L2TP tunnel switch or LNS only. The CCT can be virtual for a PPPoE session.
138	LAC-Real-Port	No	Yes	Yes	Integer. Contains the CCT handle for the real circuit of an incoming PPPoE session on an L2TP LAC. This attribute should be present for a subscriber on an L2TP tunnel switch or LNS only.

Table C-2 Redback Networks VSAs Supported by AOS

Num	Attribute Name	Receivable in Access-Response	Sent in Access-Request	Sent in Accting-Request	Notes
139	LAC-Port-Type	No	Yes	Yes	Integer. Contains the port type for the incoming session on an L2TP LAC. This attribute should be present for a subscriber on an L2TP tunnel switch or LNS only. The port can be virtual for a PPPoE session. Values for port types: NAS_PORT_TYPE_10BT = 40 NAS_PORT_TYPE_100BT = 41 NAS_PORT_TYPE_DS3_FR = 42 NAS_PORT_TYPE_DS3_ATM = 43 NAS_PORT_TYPE_OC3 = 44 NAS_PORT_TYPE_HSSI = 45 NAS_PORT_TYPE_EIA530 = 46 NAS_PORT_TYPE_T1 = 47 NAS_PORT_TYPE_CHAN_T3 = 48 NAS_PORT_TYPE_DS1_FR = 49 NAS_PORT_TYPE_E3_ATM = 50 NAS_PORT_TYPE_IMA_ATM = 51 NAS_PORT_TYPE_DS3_ATM_2 = 52 NAS_PORT_TYPE_OC3_ATM_2 = 53 NAS_PORT_TYPE_1000BSX = 54 NAS_PORT_TYPE_E1_FR = 55 NAS_PORT_TYPE_E1_ATM = 56 NAS_PORT_TYPE_E3_FR = 57 NAS_PORT_TYPE_OC3_POS = 58 NAS_PORT_TYPE_OC12_POS = 59 NAS_PORT_TYPE_PPPOE = 60
140	LAC-Real-Port-Type	No	Yes	Yes	Integer. Contains the port type for the real circuit of an incoming PPPoE session on an L2TP LAC. This attribute should be present for a subscriber on an L2TP tunnel switch or LNS only. Values for port types: NAS_PORT_TYPE_10BT = 40 NAS_PORT_TYPE_100BT = 41 NAS_PORT_TYPE_DS3_FR = 42 NAS_PORT_TYPE_DS3_ATM = 43 NAS_PORT_TYPE_OC3 = 44 NAS_PORT_TYPE_HSSI = 45 NAS_PORT_TYPE_EIA530 = 46 NAS_PORT_TYPE_T1 = 47 NAS_PORT_TYPE_CHAN_T3 = 48 NAS_PORT_TYPE_DS1_FR = 49 NAS_PORT_TYPE_E3_ATM = 50 NAS_PORT_TYPE_IMA_ATM = 51 NAS_PORT_TYPE_DS3_ATM_2 = 52 NAS_PORT_TYPE_OC3_ATM_2 = 53 NAS_PORT_TYPE_1000BSX = 54 NAS_PORT_TYPE_E1_FR = 55 NAS_PORT_TYPE_E1_ATM = 56 NAS_PORT_TYPE_E3_FR = 57 NAS_PORT_TYPE_OC3_POS = 58 NAS_PORT_TYPE_OC12_POS = 59 NAS_PORT_TYPE_PPPOE = 60

L2TP Attribute Value Pairs

The following tables contain all standard and vendor-specific attribute value pairs (AVPs) supported by the Access Operating System (AOS).

Table D-1 lists the standard Layer 2 Tunneling Protocol (L2TP) AVPs supported by the AOS, in order by AVP number.

Table D-1 Standard L2TP AVPs Supported by the AOS

Num	AVP Name	Mandatory	May be Hidden	Message Types Used In	Notes
0	Message Type	Yes (see Notes)	Yes	All	2-octet unsigned integer. Must be the first AVP in a message. When Mandatory (M) bit=1, tunnel must be cleared if message type is unknown to the implementation. If M-bit=0, unknown message type can be ignored.
1	Result Code	Yes	No	CDN StopCCN	2-octet unsigned integer plus an optional error code and optional error message.
2	Protocol Version	Yes	No	SCCRP SCCRQ	1-octet unsigned integer for the version and 1-octet unsigned integer for the revision.
3	Framing Capabilities	Yes	Yes	SCCRP SCCRQ	32-bit mask with 2 bits defined. The A-bit indicates whether asynchronous framing is supported. The S-bit indicates whether synchronous framing is supported.
4	Bearer Capabilities	Yes	Yes	SCCRP SCCRQ	32-bit mask with 2 bits defined. The A-bit indicates whether analog access is supported. The D-bit indicates whether digital access is supported.
5	Tie Breaker	No	No	SCCRQ	8-octet value used to select a single tunnel when both LAC and LNS simultaneously request a tunnel. Lower value equals higher priority.
6	Firmware Revision	No	Yes	SCCRP SCCRQ	2-octet unsigned integer encoded in a vendor-specific format.
7	Host Name	Yes	No	SCCRP SCCRQ	String. Arbitrary number of octets, with a minimum length of 1 octet.
8	Vendor Name	No	Yes	SCCRP SCCRQ	Vendor-specific string.

Table D-1 Standard L2TP AVPs Supported by the AOS

Num	AVP Name	Mandatory	May be Hidden	Message Types Used In	Notes
9	Assigned Tunnel ID	Yes	Yes	SCCRP SCCRQ StopCCN	2-octet, nonzero unsigned integer.
10	Receive Window Size	Yes	No	SCCRP SCCRQ	2-octet unsigned integer.
11	Challenge	Yes	Yes	SCCRP SCCRQ	1 or more octets of random data.
12	Q.931 Cause Code	Yes	No	CDN	Returned Q.931 cause code and returned Q.931 message code in their native ITU encodings. Optional ASCII text advisory message can also be included.
13	Challenge Response	Yes	Yes	SCCCN SCCRP	16-octet value.
14	Assigned Session ID	Yes	Yes	CDN ICRP ICRQ OCRP OCRQ	2-octet, non-zero unsigned integer.
15	Call Serial Number	Yes	Yes	ICRQ OCRQ	32-bit value.
16	Minimum BPS	Yes	Yes	OCRQ	32-bit value indicating minimum speed in bits per second.
17	Maximum BPS	Yes	Yes	OCRQ	32-bit value indicating maximum speed in bits per second.
18	Bearer Type	Yes	Yes	ICRQ OCRQ	32-bit mask with 2 bits defined. The A-bit indicates if the call refers to an analog channel. The D-bit indicates if the call refers to a digital channel. Both bits can be set. For ICRQ messages, it is also valid to set neither.
19	Framing Type	Yes	Yes	ICCN OCCN OCRQ	32-bit mask with 2 bits defined. The A-bit indicates asynchronous framing. The S-bit indicates synchronous framing.
21	Called Number	Yes	Yes	ICRQ OCRQ	ASCII string.
22	Calling Number	Yes	Yes	ICRQ	ASCII string.
23	Sub-Address	Yes	Yes	ICRQ OCRQ	ASCII string.
24	(Tx) Connect Speed	Yes	Yes	ICCN OCCN	4-octet value indicating the speed in bits per second.
25	Physical Channel ID	No	Yes	ICRQ OCRP	4-octet value for logging purposes only. Sent to RADIUS from the LNS side. Encodes the vendor specific physical channel number used for a call.
26	Initial Received LCP CONFREQ	No	Yes	ICCN	Arbitrary number of octets. A copy of the body of the initial CONFREQ received, starting at the first option within the body of the LCP message.

Table D-1 Standard L2TP AVPs Supported by the AOS

Num	AVP Name	Mandatory	May be Hidden	Message Types Used In	Notes
27	Last Sent LCP CONFREQ	No	Yes	ICCN	Arbitrary number of octets. A copy of the body of the final CONFREQ sent to the client to complete LCP negotiation, starting at the first option within the body of the LCP message.
28	Last Received LCP CONFREQ	No	Yes	ICCN	Arbitrary number of octets. A copy of the body of the final CONFREQ received from the client to complete LCP negotiation, starting at the first option within the body of the LCP message.
29	Proxy Authen Type	No	Yes	ICCN	2-octet unsigned integer.
30	Proxy Authen Name	No	Yes	ICCN	String. Arbitrary number of octets.
31	Proxy Authen Challenge	No	Yes	ICCN	String. 1 or more octets.
32	Proxy Authen ID	No	Yes	ICCN	2-octet unsigned integer.
33	Proxy Authen Response	No	Yes	ICCN	String. Arbitrary number of octets.
34	Call Errors	Yes	Yes	WEN	Includes the following fields: Reserved, CRC Errors, Framing Errors, Hardware Overruns, Buffer Overruns, Time-out Errors, and Alignment Errors.
35	ACCM	Yes	Yes	SLI	Send and Receive ACCM are each 4-octet values preceded by a 2-octet reserved quantity.
36	Random Vector	Yes	No	All	String of arbitrary length. Must precede the first AVP with the Hidden (H) bit set. More than one can be used per message. Hidden AVP uses the Random Vector AVP most closely preceding it.
37	Private Group	No	Yes	ICCN	Arbitrary number of octets.
38	Rx Connect Speed	No	Yes	ICCN OCCN	4-octet value indicating the speed in bits per second.
39	Sequencing Required	Yes	No	ICCN OCCN	This AVP has no value field. Indicates that sequence numbers must be present on the data channel. The Redback implementation of L2TP prefers not to require sequencing. Therefore, if the SMS device is functioning as a LAC, it will never request this attribute. If the LNS uses it, the LAC will honor it, however. If the SMS device is functioning as an LNS, it will honor a LAC's request for this attribute, but will never volunteer it.

Redback vendor-specific AVPs are embedded according to the procedure recommended in RFC 2661, *Layer 2 Tunneling Protocol "L2TP."* Table D-2 lists the Redback vendor-specific L2TP AVPs supported by the AOS, in order by AVP number.

Table D-2 Redback Vendor-Specific L2TP AVPs Supported by the AOS

Num	AVP Name	Mandatory	May be Hidden	Message Types Used In	Notes
40	Ethernet MAC Address	No	No	ICRQ	6 octets. Sent by the LAC for an Ethernet-encapsulated session. Indicates the Ethernet MAC address at the LAC.
41	Real Physical Channel ID	No	No	ICRQ	4 octets. Sent to RADIUS from the LNS side. The circuit handle of the real circuit on the LAC for an originating session. Used only for PPPoE.
42	Media Type	No	No	ICRQ	4 octets. Sent to RADIUS from the LNS side. Passes on any configured media type (for example, DSL, cable, wireless, or satellite) on the originating circuit on the LAC.
43	NAS Port Type	No	No	ICRQ	4 octets. Sent to RADIUS from the LNS side. Indicates the port type for the originating circuit on the LAC.
44	Real NAS Port Type	No	No	ICRQ	4 octets. Sent to RADIUS from the LNS side. The port type of the real port for the originating circuit on the LAC. Used only for PPPoE.
45	First LAC Name	No	No	ICRQ	String of one or more octets. Generated only when tunnel switching. Contains the host name of the originating LAC.
46	Framing Capabilities	No	No	SCCRQ SCCRP	32 bits. The only valid value is 8 to indicate the peer is capable of supporting Ethernet-encapsulated sessions on tunnel. Only present in SCCRQ and SCCRP tunnel setup messages.
47	Bearer Type	No	No	ICRQ	32 bits. The only valid value is 8 to indicate the session is Ethernet encapsulated. Only present in ICRQ.

Part 15

Indexes

Index

Symbols

Numerics

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y