Lucent Technologies
Bell Labs Innovations

Stinger®

IP2000 Configuration Guide

Part Number: 7820-0976-005
For software version 9.7.0
August 2004

Notice

Every effort was made to ensure that the information in this document was complete and accurate at the time of printing, but information is subject to change.

European Community (EC) RTTE compliance

CE Hereby, Lucent Technologies, declares that the equipment documented in this publication is in compliance with the essential requirements and other relevant provisions of the Radio and Telecommunications Technical Equipment (RTTE) Directive 1999/5/EC.

To view the official *Declaration of Conformity* certificate for this equipment, according to EN 45014, access the Lucent INS online documentation library at http://www.lucentdocs.com/ins.

Safety, compliance, and warranty Information

Before handling any Lucent Access Networks hardware product, read the *Edge Access and Broadband Access Safety and Compliance Guide* included in your product package. See that guide also to determine how products comply with the electromagnetic interference (EMI) and network compatibility requirements of your country. See the warranty card included in your product package for the limited warranty that Lucent Technologies provides for its products.

Security statement

In rare instances, unauthorized individuals make connections to the telecommunications network through the use of access features.

Trademarks

Lucent, the Lucent logo, and all Lucent brand and product names are trademarks or registered trademarks of Lucent Technologies Inc. Other brand and product names are trademarks of their respective holders.

Ordering Information

You can order the most up-to-date product information and computer-based training online at http://www.lucentdocs.com/bookstore.

How to comment

To comment on this information product, go to the Online Comment Form (http://www.lucent-info.com/comments/enus/) or email your comments to the Comments Hotline (comments@lucent.com).

Lucent Technologies

Customer Service

Product and service information, and software upgrades, are available 24 hours a day. Technical assistance options accommodate varying levels of urgency.

Finding information and software

To obtain software upgrades, release notes, and addenda for this product, log in to Lucent OnLine Customer Support at `http://www.lucent.com/support`.

Lucent OnLine Customer Support also provides technical information, product information, and descriptions of available services. The center is open 24 hours a day, seven days a week. Log in and select a service.

Obtaining technical assistance

Lucent OnLine Customer Support at `http://www.lucent.com/support` provides access to technical support. You can obtain technical assistance through email or the Internet, or by telephone. If you need assistance, make sure that you have the following information available:

- Active service or maintenance contract number, entitlement ID, or site ID
- Product name, model, and serial number
- Software version
- Software and hardware options If supplied by your carrier, service profile identifiers (SPIDs) associated with your line
- Your local telephone company's switch type and operating mode, such as AT&T, 5ESS Custom, or Northern Telecom National ISDN-1
- Whether you are routing or bridging with your Lucent product
- Type of computer you are using
- Description of the problem

Obtaining assistance through email or the Internet

If your services agreement allows, you can communicate directly with a technical engineer through Email Technical Support or a Live Chat. Select one of these sites when you log in to `http://www.lucent.com/support`.

Calling the technical assistance center (TAC)

If you cannot find an answer through the tools and information of Lucent OnLine Customer Support or if you have a very urgent need, contact TAC. Access Lucent OnLine Customer Support at `http://www.lucent.com/support` and click `Contact Us` for a list of telephone numbers inside and outside the United States.

Alternatively, call 1-866-LUCENT8 (1-866-582-3688) from any location in North America for a menu of Lucent services. Or call +1 510-747-2000 for an operator. You must have an active services agreement or contract.

Contents

Figures

Tables

About This Guide

A Stinger unit with the IP2000 control module (a Stinger IP2000) supports identical Asynchronous Transfer Mode (ATM) capabilities to those in Stinger units with the standard control module. In addition, a Stinger IP2000 can terminate IP traffic and forward it across a built-in Gigabit Ethernet interface.

Note Instructions for installing and configuring the management functions of the IP2000 are found in the *Getting Started Guide* for your Stinger platform.

What is in this guide

This guide focuses on the aspects of Stinger configuration that are specific to IP2000 control module capabilities. To fully configure the system for both ATM and IP capabilities, use this guide with the *Stinger ATM Configuration Guide*.

Note You can configure the amount of bandwidth allocated to LIM interfaces and control modules for carrying upstream traffic. For details about that aspect of using the IP2000 control module, as well as for ATM quality of service (QoS) and other traffic management capabilities, see the *Stinger ATM Configuration Guide*.

This guide describes how to configure IP routing and related functions in the Stinger Stinger. It includes information about local and global network IP issues, as well as how to configure both IP-routed switch-through ATM permanent virtual circuits (PVCs) and RFC 2684 PVCs.

This guide also describes how to set up IEEE 802.1Q virtual local area network (VLAN) support on the Gigabit Ethernet interface, and how to configure the system to support multicast video over DSL with Internet Group Management Protocol (IGMP) version-1 or version-2 messaging.

Warning Before installing your Stinger unit, be sure to read the safety instructions in the *Edge Access and Broadband Access Safety and Compliance Guide*. For information specific to your unit, see the "Safety-Related Physical, Environmental, and Electrical Information" appendix in the *Getting Started Guide* for your Stinger unit.

Documentation conventions

Following are all the special characters and typographical conventions used in this manual:

Convention	Meaning
`Monospace text`	Represents text that appears on your computer's screen, or that could appear on your computer's screen.
`Boldface monospace text`	Represents characters that you enter exactly as shown (unless the characters are also in ***`italics`***—see *Italics*, below). If you could enter the characters but are not specifically instructed to, they do not appear in boldface.
Italics	Represent variable information. Do not enter the words themselves in the command. Enter the information they represent. In ordinary text, italics are used for titles of publications, for some terms that would otherwise be in quotation marks, and to show emphasis.
[ ]	Square brackets indicate an optional argument you might add to a command. To include such an argument, type only the information inside the brackets. Do not type the brackets unless they appear in boldface.
|	Separates command choices that are mutually exclusive.
>	Points to the next level in the path to a parameter or menu item. The item that follows the angle bracket is one of the options that appear when you select the item that precedes the angle bracket.
Key1+Key2	Represents a combination keystroke. To enter a combination keystroke, press the first key and hold it down while you press one or more other keys. Release all the keys at the same time. (For example, Ctrl+H means hold down the Ctrl key and press the H key.)
Press Enter	Means press the Enter or Return key or its equivalent on your computer.
 **Note**	Introduces important additional information.
 **Caution**	Warns that a failure to follow the recommended procedure could result in loss of data or damage to equipment.
 **Warning**	Warns that a failure to take appropriate safety precautions could result in physical injury.
 **Warning**	Warns of danger of electric shock.

Stinger documentation set

The Stinger documentation set consists of the following manuals, which can be found at `http://www.lucent.com/support` and `http://www.lucentdocs.com/ins`.

- **Read me first:**
  - *Edge Access and Broadband Access Safety and Compliance Guide*. Contains important safety instructions and country-specific information that you must read before installing a Stinger unit.
  - *TAOS Command-Line Interface Guide*. Introduces the TAOS command-line environment and shows you how to use the command-line interface effectively. This guide describes keyboard shortcuts and introduces commands, security levels, profile structure, and parameter types.
- **Installation and basic configuration:**
  - *Getting Started Guide* for your Stinger platform. Shows how to install your Stinger chassis and hardware. This guide also shows you how to use the command-line interface to configure and verify IP access and basic access security on the unit, and how to configure Stinger control module redundancy on units that support it.
  - *Stinger Compact Remote Installation and Configuration Guide*. Provides an overview of the Stinger Compact Remote and instructions for the installation and replacement of its components. This guide also describes how to configure and manage the Compact Remote as a hosted unit.
  - Module guides. For each Stinger line interface module (LIM), trunk module, or other type of module, an individual guide describes the module's features and provides instructions for configuring the module and verifying its status.
- **Configuration:**
  - *Stinger ATM Configuration Guide*. Describes how to integrate the Stinger into the ATM and Digital Subscriber Line (DSL) access infrastructure. The guide explains how to configure PVCs, and shows how to use standard ATM features such as quality of service (QoS), connection admission control (CAC), and subtending.
  - *Stinger IP2000 Configuration Guide*. For Stinger IP2000 systems, this guide describes how to integrate the system into the IP infrastructure. Topics include IP-routed switch-through ATM PVCs and RFC 2684 PVCs, IEEE 802.1Q VLAN, and forwarding multicast video transmissions on DSL interfaces.
  - *Stinger Private Network-to-Network Interface (PNNI) Supplement*. For the optional PNNI software, this guide provides quick-start instructions for configuring PNNI and soft PVCs (SPVCs), and describes the related profiles and commands.
  - *Stinger SNMP Management of the ATM Stack Supplement.* Describes SNMP management of ATM ports, interfaces, and connections on a Stinger unit to provide guidelines for configuring and managing ATM circuits through any SNMP management utility.
  - *Stinger T1000 Module Routing and Tunneling Supplement.* For the optional T1000 module, this guide describes how to configure the Layer 3 routing and virtual private network (VPN) capabilities.

- **RADIUS:** *TAOS RADIUS Guide and Reference*. Describes how to set up a unit to use the Remote Authentication Dial-In User Service (RADIUS) server and contains a complete reference to RADIUS attributes.
- **Administration and troubleshooting:** *Stinger Administration Guide*. Describes how to administer the Stinger unit and manage its operations. Each chapter focuses on a particular aspect of Stinger administration and operations. The chapters describe tools for system management, network management, and Simple Network Management Protocol (SNMP) management.
- **Reference**:
  - *Stinger Reference*. An alphabetic reference to Stinger profiles, parameters, and commands.
  - *TAOS Glossary*. Defines terms used in documentation for Stinger units.

Related documents

The following industry documents provide background information about features described in this guide:

- RFC 951, *Bootstrap Protocol*
- RFC 1112, *Host Extensions for IP Multicasting*
- RFC 1334, *PPP Authentication Protocols*
- RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*
- RFC 1587, *The OSPF NSSA Option.*
- RFC 1700, *Assigned Numbers*
- RFC 1723, *RIP Version 2: Carrying Additional Information*
- RFC 1812, *Requirements for IP Version 4 Routers*
- RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*
- RFC 2131, *Dynamic Host Configuration Protocol*
- RFC 2132, *DHCP Options and BOOTP Vendor Extensions*
- RFC 2236, *Internet Group Management Protocol Version 2*
- RFC 2328, *OSPF Version 2*
- RFC 2362, *Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification,* `draft-ietf-pim-sm-v2-new-07.txt`, March 2003, `draft-ietf-pim-sm-bsr-03.txt`, February 2003
- RFC 2364, *PPP over AAL5*
- RFC 2516, *A Method for Transmitting PPP Over Ethernet (PPPoE)*
- RFC 3046, *DHCP Relay Agent Information Option*
- RFC 2697, *A Single Rate Three Color Marker*
- IEEE 802.1Q-1998, *IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks*
- IEEE 802.1P, *LAN Layer 2 QoS/CoS Protocol for Traffic Prioritization*

Welcome to the IP2000

1

Stinger platforms and model numbers

The IP2000 control module supports a fiber-based Gigabit Ethernet (GigE) interface, with a modular Small Form Factor Pluggable (SFP) transceiver. The IP2000 is supported on the Stinger FS, the Stinger FS+, Stinger LS, and Stinger RT platforms. Table 1-1 shows IP2000 model numbers and platform support:

Table 1-1. IP2000 model numbers and platform support

IP2000 model number	Description	Supporting platforms
STGR-CM-IP2000-F	IP2000 with fiber gigabit Ethernet	Stinger FS Stinger FS+ Stinger LS
STGRRT-CM-IP2000-F	IP2000 with fiber gigabit Ethernet, environmentally hardened	Stinger RT Stinger Compact Remote
STGR-SFP-SX	Short-haul gigabit Ethernet SFP module	Stinger FS Stinger FS+ Stinger LS
STGR-SFP-LX	Long-haul gigabit Ethernet SFP module	Stinger FS Stinger FS+ Stinger LS
STGRRT-SFP-LX	Long-haul gigabit Ethernet SFP module, environmentally hardened	Stinger RT Stinger Compact Remote

IP2000 software specifications

Table 1-2 shows connection features supported by the IP2000 control module:

Table 1-2. Connection features

Software capability	Specifications
ATM protocols	ATM Forum UNI (v3.0 and v3.1)
	ATM Forum Traffic Management v4.0
	ATM Forum PNNI 1.0 (optional)
Routing protocols	RIPv1, RIPv2, OSPF
IP multicast	Internet Group Management Protocol (IGMP) v1, v2, Protocol Independent Multicast Sparse Mode (PIM-SM v2)
IP/ATM	Multiprotocol Encapsulation over ATM Adaptation Layer 5 (RFC 2684), bridged IP routing (BIR). Point-to-Point over Ethernet (PPPoE), and Point-to-Point over ATM (PPPoA)
Broadband RAS	Broadband remote access server (BRAS) for PPP sessions over DSL interfaces
IP/Ethernet	IP support for Gigabit Ethernet interface
VLAN	IEEE 802.1Q tagged VLANs, IEEE 802.1P prioritization, stacked VLAN
ATM QoS and IP CoS	ATM traffic is assigned the highest priority and passed through. IP traffic is assigned a strict priority based on the classification rules in a `packet-flows` profile. However, independent of the classification rules, the protocol control messages always take the highest priority.
	IP traffic shaping in downstream direction (toward CPE) on per-VC basis
	Traffic management for CoS and ATM queues
Security	RADIUS, Extended RADIUS
	Password Authentication Protocol (PAP)
	Challenge Authentication Protocol (CHAP)
	Profile-based access
	Telnet Access Control Lists (TACL)

IP2000 hardware specifications

Table 1-3 shows hardware specifications for the IP2000 control module:

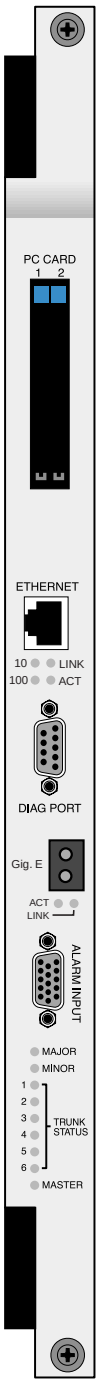


Table 1-3. IP2000 hardware specifications

Category	Specifications	
Physical dimensions	Height:	15 inches (38.1cm)
	Width:	1.06 inches (2.69cm)
	Depth:	9 inches (22.8cm)
Weight	3.4 pounds (1.5kg)	
Operating Requirements	Power:	70 Watts Maximum
	Temperature:	FS/LS version: 32°F–131°F (0°C–55°C)
		RT version: 40°F–149°F (-40°C–65°C)
	Relative humidity:	10% through 95% (noncondensing)
	Operating altitude:	Up to 13,123 feet (4,000m)
LAN interface (fiber)	Modular Small Form Factor Pluggable (SFP) transceiver with duplex LC connector	
	IEEE 802.3z 1000BASE-SX (short haul) over multi-mode fiber, distance support to 550m	
	IEEE 802.3z 1000BASE-LX (long haul) over single mode fiber, distance support to 10km	
Mgmt interfaces	10/100 BASE-T Ethernet, RS-232 serial port	
Status indicators	10/100 BaseT	10 (Green): 10 Mbps speed
		100 (Green): 100 Mbps speed
		LINK (Green): Operational link
		ACT (Green): Traffic activity
	Gigabit Ethernet	LINK (Green): Operational link
		ACT (Green): Traffic activity
	Others	MAJOR (Red): Major alarm detected
		MINOR (Red): Minor alarm detected
		TRUNK STATUS 1-6 (Amber): Trunk port status
		MASTER (Green): Module is master controller
Electromagnetic compliance	FCC Part 15 Class A, EN55022 Class A, AS/NZS 3548 Class A, VCCI Class A, CISPR 22 Class A, EN 300386-2	
Certification	Bellcore GR-63-CORE (NEBS Level 1-3), Bellcore-GR-1089-CORE, EN / IEC 60950	
Expansion slot	One PC card slot for configuration or upgrade storage	
Switching fabric	64x64 nonblocking ATM crosspoint switch	
	1.6Gbps ATM switching capacity	
	2.4Gpbs IP switching/routing capacity	

Network architecture overview

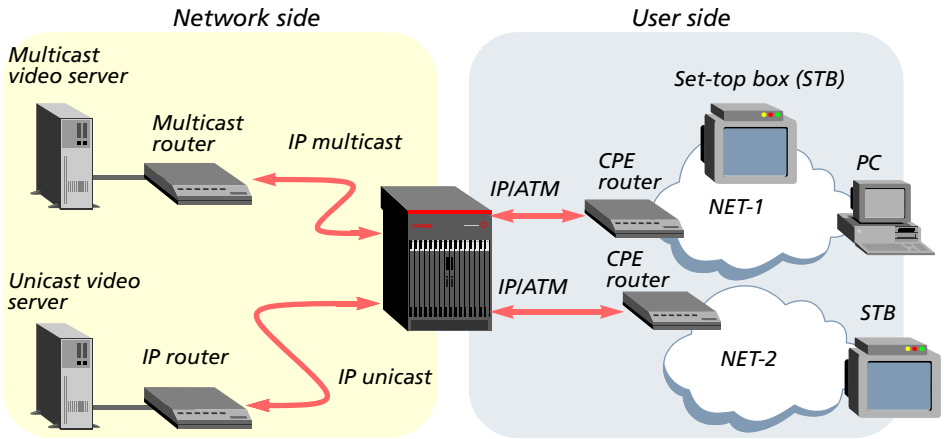
A Stinger IP2000 supports all standard Stinger ATM features, such as data and voice services over DSL. In addition, it supports IP services such as multicast video, unicast video-on-demand, and other video and IPTV applications. The services supported by the IP2000 are provided downstream to DSL subscribers. In the upstream direction, DSL subscribers accessing the Stinger IP2000 can be routed via the IP infrastructure to Internet services, or bridged/routed to a virtual LAN.

A Stinger IP2000 supports IP routing, ATM quality of service and traffic shaping, VLAN, and multicasting capabilities to provide fast, efficient access to ATM and IP services.

Multicast video

A Stinger IP2000 uses the Internet Group Management Protocol (IGMP) to manage group memberships of downstream video to a PC application or set-top box, as shown in Figure 1-1. Administrators can configure levels of service that control subscribers' access to specific multicast groups. Connection to originating router can be across the Gigabit Ethernet interface or through a high-speed IP over ATM connection.

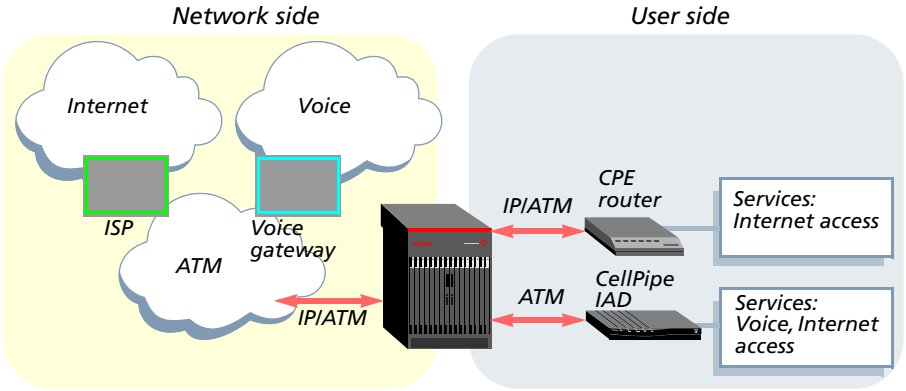
Figure 1-1. Sample setup showing multicast and unicast video services



Internet and voice access

When a subscriber has DSL Integrated Access Device (IAD) equipment (such as a CellPipe®), the Stinger can deliver integrated voice and data services over the local copper loop, providing a efficient, low-cost solution for enterprise, small business, home office, and residential subscribers.

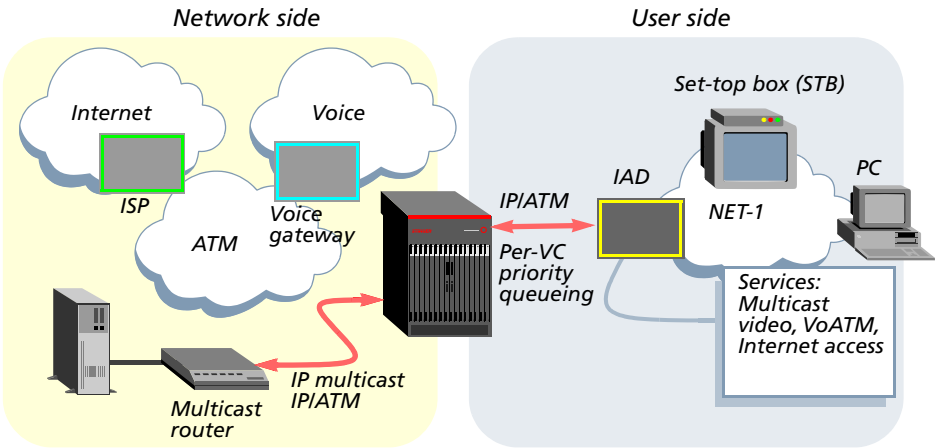
Figure 1-2. Sample setup showing Internet access and voice over ATM



Multiplexing multiple IP flows on a single ATM VCC

A Stinger IP2000 supports an implementation of Class of Service (CoS) that co-exists with the Stinger ATM QoS implementation. This feature allows transferring multiple IP streams (multicast and unicast) over single user-side ATM virtual circuit with different levels of priority.

Figure 1-3. Sample setup showing multiple IP flows to a CPE router



The CoS implementation enables the delivery of differentiated services over an IP infrastructure. All traffic handled by the IP2000, whether encapsulated IP or native ATM, passes through the network processor function.

Non-IP terminated ATM traffic, including operations, administration, and maintenance (OAM) F5 traffic, is treated as highest priority and handled in an *ATM pass-through mode*. This traffic passes through the network processor with no further processing.

RFC 2684 IP traffic that terminates on the IP2000 is reassembled from ATM cells into IP packets. It is then classified and assigned to priority output queues. A default per-VC strict-priority queuing is supported with three priority levels as described in Table 1-4.

Table 1-4. Default classification and prioritization

Priority queue	Priority level	Packet classification assigned to queue
1	High	IP Control Protocol Classification
		■ ARP/RARP protocol messages
		■ ICMP protocol messages
		■ RIP protocol messages
		■ IGMP protocol messages
2	Medium	Multicast Classification
		■ IP multicast data
3	Low	Unicast Classification
		■ IP unicast data

Note Table 1-4 shows a default classification that occurs when no `packet-flows` profile has been applied to the traffic. For information about priority queuing based on flow identification, see Chapter 5, "Ethernet and IP QoS."

Per-VC queuing operates in conjunction with the associated ATM shaping rate. The aggregate rate of the combination of three priority queues (Class of Service Queuing with Strict Priority) associated with a particular ATM virtual circuit is controlled by the SCR (sustained cell rate) configured for the VC. In this case, SCR is configured equal to PCR (peak cell rate). Rate information is configurable in the `atm-qos` profile for each virtual circuit. For details about configuring ATM QoS, see the *Stinger ATM Configuration Guide*.

Gigabit Ethernet Configuration 2

The IP2000 controller has two Ethernet interfaces, one 10/100 BASE-T interface for management access to the unit via Telnet or SNMP, and one Gigabit Ethernet interface for high-speed access to a local IP subnet. For information about configuring the management interface, see the *Getting Started Guide* for your Stinger platform.

The Gigabit Ethernet MAC (GMAC) physical interface operates only in full-duplex mode only for a full 1Gbps throughput. It supports auto-negotiation for advertising its rate and duplex mode, but not for renegotiating it on the IEEE 802 LAN.

Stinger units with redundant IP2000 controllers can be configured to enable Gigabit Ethernet port redundancy. With proper configuration, RFC 2684 (MPoA) connections and MBONE interface functions can be maintained across the Gigabit Ethernet interface following a controller switchover.

Configuring the physical and logical interface

The system creates configuration profiles for both IP2000 Ethernet interfaces. For each controller, interface 1 is always the 10/100 BASE-T management interface, and interface 2 is the Gigabit Ethernet interface. For example:

```
admin> dir ethernet
    18  07/11/2004 13:55:31  { shelf-1 first-control-module 1 }
    24  07/11/2004 19:34:41  { shelf-1 first-control-module 2 }
    18  07/11/2004 13:55:31  { shelf-1 second-control-module 1 }
    24  07/11/2004 13:57:32  { shelf-1 second-control-module 2 }
admin> dir ip-interface
    21  07/11/2004 13:55:31  { { any-shelf any-slot 0 } 0 }
    31  07/11/2004 22:46:34  { { shelf-1 first-control-module 1 } 0 }
    21  07/11/2004 13:57:01  { { shelf-1 first-control-module 2 } 0 }
    21  07/11/2004 13:55:31  { { shelf-1 second-control-module 1 } 0 }
    21  07/11/2004 13:57:01  { { shelf-1 second-control-module 2 } 0 }
```

Overview of ethernet profile settings

With the default settings, the Gigabit Ethernet interface is fully operational. To change defaults or to enable bridging to allow the system to perform VLAN operations, open the `ethernet` profile. For example:

```
admin> read ethernet { 1 8 2 }

admin> list
[in ETHERNET/{ shelf-1 first-control-module 2 }]
interface-address* = { shelf-1 first-control-module 2 }
link-state-enabled = no
enabled = yes
ether-if-type = fiber
bridging-enabled = no
filter-name = ""
duplex-mode = full-duplex
pppoe-options = { no no }
bridging-options = { 0 no no transparent-bridging 0 0 "" 0 }
media-speed-mbit = 100mb
auto-negotiate = no
vlan-stack-tag-type = 91:00
```

For details about each of the parameters shown above, see the *Stinger Reference*. Following are some Gigabit Ethernet-specific notes about the profile contents:

Parameter	Notes about Gigabit Ethernet settings
`interface-address*`	The profile index and `interface-address` value of the profile for a Gigabit Ethernet interface always specifies an interface number of `2`. For example: `shelf-1 first-control-module 2`
`link-state-enabled`	With the default value, the system discards packets and does not choose an alternate route if the interface is down. If you set this to `yes`, the system deletes routes to the interface when the interface is unavailable, and then restores the routes when the interface becomes available again.
`enabled`	If you set this to `no` and write the profile, the interface is unavailable.
`ether-if-type`	This setting is read-only and specifies either `fiber` or `utp` (CAT5 unshielded twisted pair).
`bridging-enabled`	See "Enabling layer 2 bridging for VLAN operations" on page 2-3.
`filter-name`	Applies a data filter to the interface. See Chapter 10, "Filter Configuration."
`duplex-mode`	This setting is read-only and specifies full-duplex mode.
`pppoe-options`	*Not used by the IP2000.*
`bridging-options`	The options in this subprofile are not used for VLAN bridging.
`media-speed-mbit`	This setting is read-only and specifies 1Gbps.

Parameter	Notes about Gigabit Ethernet settings
`auto-negotiate`	Setting this parameter to `yes` does not cause the IP2000 to negotiate its duplex mode or speed, but it does cause the system to advertise a full-duplex 1Gbps port, which helps to ensure compatibility with remote Gigabit Ethernet interfaces that support autonegotiation.
`vlan-stack-tag-type`	If the layer-2 core network is using a specific EtherType value for stacked VLAN frames, you must set the `vlan-stack-tag-type` parameter to that value. For details, see "Configuring stacked VLANs" on page 3-13.

Enabling layer 2 bridging for VLAN operations

Following is the relevant parameter, shown with its default setting, for enabling bridging on the Gigabit Ethernet port:

```
[in ETHERNET/{ shelf-1 first-control-module 2 }]
bridging-enabled = no
```

Parameter	Setting
`bridging-enabled`	Configures the physical port to accept Ethernet frames for bridging purposes. With the default `no` value, the system does not accept unicast packets received on this port unless the destination MAC address is equal to the MAC address of the port. Set this parameter to `yes` if the system will support VLAN operations.

The following commands enable bridging on Gigabit Ethernet:

```
admin> read ethernet { 1 8 2 }
admin> set bridging-enabled = yes
admin> write -f
```

Assigning an IP address in the `ip-interface` profile

For details about the `ip-interface` profile, and about enabling dynamic routing or configuring static routes to enable the system to communicate beyond its own subnet, see "Configuring ip-interface profiles for Ethernet ports" on page 4-5. The following commands provide the minimal configuration of an IP address for the Gigabit Ethernet interface:

```
admin> read ip-interface { { 1 8 2 } 0 }
admin> set ip-address = 10.99.99.101/24
admin> write -f
```

Verifying the Gigabit Ethernet interface setup

After you assign an IP address, you can verify that the Gigabit Ethernet interface is able to handle IP traffic by checking some command output. For details about the `netstat` and `gmac` commands, see the *Stinger Reference*.

You can also use the debug-level `ifmgr -d` command to verify that the Gigabit Ethernet interface is active. This is described in Appendix A, "IP2000 Diagnostics."

Checking the routing table

The following command output verifies that the routing table has an entry for the Gigabit Ethernet interface (IP address 100.1.1.3/32):

```
admin> netstat -rn
Destination       Gateway        IF         Flg   Pref Met     Use        Age
0.0.0.0/0         1.1.2.1       ie0        SGP     60   1    3817        828
20.1.2.0/24       -             ie1-1       C       0   0       0        828
20.1.2.3/32       -             local       CP      0   0       0        828
100.0.0.0/8       -             ie1         C       0   0    4683        828
100.1.1.3/32      -             local       CP      0   0    1580        828
127.0.0.0/8       -             bh0         CP      0   0       0        828
127.0.0.1/32      -             local       CP      0   0       0        828
127.0.0.2/32      -             rj0         CP      0   0       0        828
1.1.2.0/24        -             ie0         C       0   0    1428        828
1.1.2.65/32       -             local       CP      0   0    2937        828
224.0.0.0/4       -             mcast       CP      0   0       0        828
224.0.0.1/32      -             local       CP      0   0       0        828
224.0.0.2/32      -             local       CP      0   0       0        828
224.0.0.9/32      -             local       CP      0   0       0        828
255.255.255.255/32 -             ie0         CP      0   0       0        828
```

Verifying the network processor setup for the interface

The network processor on the IP2000 creates a connection entry for the Gigabit Ethernet interface when the interface becomes operational. You can force the network processor to create a connection entry for the Gigabit Ethernet interface by using the following command:

```
admin> gmac -n
NP setup for gmac done.
```

Verifying the SAR setup for the interface

The Stinger Segmentation and Reassembly (SAR) creates an ATM connection entry for the Gigabit Ethernet interface. You can force the SAR setup by using the following command:

```
admin> gmac -s
GMAC: SAR conn. open with vpi = 0, vci = 200
```

Verifying IP packet transfer on the interface

The following command clears statistics gathered on the Gigabit Ethernet interface:

```
admin> gmac -d -c
```

The next command pings a host on the same subnet as the Gigabit Ethernet interface:

```
admin> ping 100.1.1.10
PING 100.1.1.10 (100.1.1.10): 56 data bytes
64 bytes from 100.1.1.10: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 100.1.1.10: icmp_seq=1 ttl=255 time=0 ms
```

```
64 bytes from 100.1.1.10: icmp_seq=2 ttl=255 time=0 ms
64 bytes from 100.1.1.10: icmp_seq=3 ttl=255 time=0 ms
64 bytes from 100.1.1.10: icmp_seq=4 ttl=255 time=0 ms
64 bytes from 100.1.1.10: icmp_seq=5 ttl=255 time=0 ms
64 bytes from 100.1.1.10: icmp_seq=6 ttl=255 time=0 ms
64 bytes from 100.1.1.10: icmp_seq=7 ttl=255 time=0 ms
--- 100.1.1.10 ping statistics ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 0/0/0 ms
```

The following command displays GMAC statistics that show packet transfer. The `txGoodPackets` and `rxGoodPackets` fields in the command output show 8 packets transmitted and received in the ICMP sequence shown immediately above. For more details on the command output fields, see "Total transmit statistics" on page A-26 and "Total receive statistics" on page A-27.

```
admin> gmac -d
Gigabit Ethernet port statistics:

txOctetsLow     = 162450        rxOctetsLow     = 13192
txOctetsHigh    = 0             rxOctetsHigh    = 0
txGoodPackets   = 1874          rxGoodPackets   = 53
txPkt64         = 11            rxPkt64         = 3
txPkt65127      = 1813          rxPkt65127      = 0
txPkt128255     = 0             rx128255        = 0
txPkt256511     = 50            rx256511        = 50
txPkt5121023    = 0             rx5121023       = 0
txPkt1024Max    = 0             rx1024Max       = 0
txPktDefer      = 0             rxMacType       = 0
txPktUndSz      = 0             rxCrcErrors     = 0
txUnderFlow     = 0             rxUnderSize     = 0
txPfcf          = 0             rxOverSize      = 0
txPfcc          = 0             rxAlmostFull    = 0
txRfcf          = 0             rxOverRun       = 0
txRfcc          = 0             rxMulticastPkts = 0
txOverFlow      = 0             rxBroadcastPkts = 50
txAlmostFull    = 0             rxJabber        = 0
                                rxPfc           = 0
                                rxRfc           = 0
```

Gigabit Ethernet port redundancy

With the proper configuration, systems with redundant controllers support Gigabit Ethernet redundancy, which enables the system to maintain RFC 2684 (MPoA) connections, VLAN connections, and LAN MBONE interface functions across a controller switchover.

Note A soft IP interface configuration is required for Gigabit Ethernet redundancy of RFC 2684 connections and a LAN MBONE interface.

Configuring a soft IP interface for Gigabit Ethernet redundancy

The soft IP interface is an internal interface that is not associated with a specific physical port, but that can be accessed through the Ethernet interface of whichever controller is primary. For background information, see "Defining a soft interface for increased accessibility" on page 4-8.

The system creates one soft interface profile by default. For example:

```
admin> dir ip-interface
        35  07/10/2004 11:26:10  { { any-shelf any-slot 0 } 0 }
        35  07/10/2004 11:26:10  { { shelf-1 first-control-module 1 } 0 }
        38  07/10/2004 11:26:11  { { shelf-1 first-control-module 2 } 0 }
        35  07/10/2004 11:26:10  { { shelf-1 second-control-module 1 } 0 }
        38  07/10/2004 11:26:11  { { shelf-1 second-control-module 2 } 0 }
```

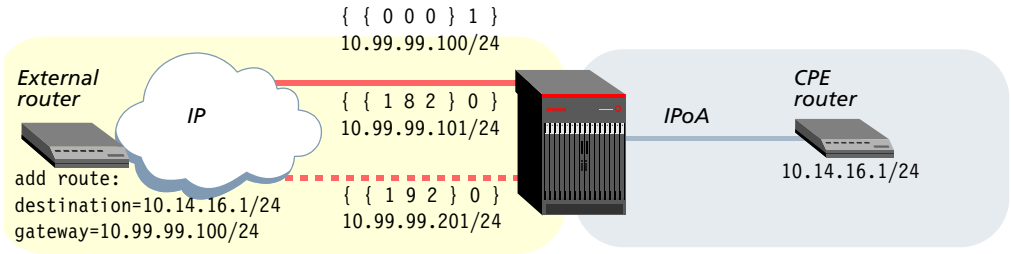
You can use the default soft IP interface { {0 0 0 } 0 } for Gigabit Ethernet redundancy. However, if you have already used the default profile for the soft IP address of the 10/100M base Ethernet management ports, you can create another soft IP interface using a profile index of { { 0 0 0 } *x* }, as long as the IP address in that profile is on the same subnet as the Gigabit Ethernet ports.

Note The system associates its Ethernet interfaces with a particular soft address based on the subnet assignment. The IP interface address of the Gigabit Ethernet ports on the primary and secondary controllers and the soft IP interface address must be on the same subnet.

Configuring Gigabit Ethernet redundancy for RFC 2684 (IPoA) connections

Figure 2-1 shows a Stinger with redundant IP2000 controllers. The Gigabit Ethernet port in slot 8 ({ { 1 8 2 } 0 }), the Gigabit Ethernet port in slot 9 ({ { 1 9 2 } 0 }), and the soft IP interface ({ { 0 0 0 } 1 }), all have IP address assignments on the same subnet. In addition, the external router has a routing table entry that specifies the soft IP interface address as the gateway to the CPE router destination.

Figure 2-1. Gigabit Ethernet redundancy for RFC 2684 connectivity



The following commands configure the Gigabit Ethernet port in slot 8:

```
admin> read ip-interface { { 1 8 2 } 0 }
admin> set ip-address = 10.99.99.101/24
admin> write -f
```

The next commands configure the Gigabit Ethernet port on slot 9:

```
admin> read ip-interface { { 1 9 2 } 0 }
admin> set ip-address = 10.99.99.201/24
admin> write -f
```

The following commands configure a soft IP interface on the same subnet:

```
admin> new ip-interface { { 0 0 0 } 1 }
admin> set ip-address = 10.99.99.100/24
admin> write -f
```

When you write the profile of the soft interface, the system displays a message:

```
LOG notice, Shelf 1, Controller-1, Time: 11:42:57--
Soft ip will be effective if the ip-addr of primary controller is
configured.
```

To ensure that the external router can reach the CPE router in Figure 2-1, the external router must specify the soft IP address as the gateway to the CPE router destination address. For example,

```
destination-address = 10.14.16.1/24
gateway-address = 10.99.99.100/24
```

Configuring Gigabit Ethernet redundancy for VLAN bridging

To enable the system to maintain VLAN bridging connections across a controller switchover, you must configure the VLAN on the soft interface, using the expression *any-slot* or 0 as the slot number. For example, the following commands define a new Gigabit Ethernet-redundant VLAN with bridge group 95 and VLAN ID 95:

```
admin> new bridge-group 95
admin> set enable = yes
admin> set bridging-group = 95
admin> set mac-entry-age-time = 300
admin> set igmp-snooping-enabled = yes
admin> set port-block-enabled = yes
admin> set lan-router-interface-address physical-address shelf = shelf-1
admin> set lan-router-interface-address physical-address slot = any-slot
admin> set lan-router-interface-address physical-address item-number = 2
admin> set lan-router-interface-address logical-item = 95
admin> write -f
admin> new vlan-ethernet { { 1 a 2 } 95 }
admin> set enabled = yes
admin> set bridging-options bridging-group = 95
admin> set bridging-options bridge = yes
admin> set bridging-options bridge-type = transparent-bridging
admin> write -f
```

The following commands define a `connection` profile that uses the redundant interface, so it will be maintained across a switchover:

```
admin> admin> new connection raj-agg-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options ip-routing-enabled = no
admin> set bridging-options bridging-group = 95
admin> set bridging-options bridge = yes
admin> set bridging-options bridge-type = transparent-bridging
admin> set atm-options vci = 95
admin> set atm-options nailed-group = 151
admin> write -f
```

To modify an existing VLAN bridge circuit for Gigabit Ethernet redundancy, you must create a new configuration and then delete the old one. For example, the following command shows an existing VLAN bridge circuit with VLAN ID 50:

```
admin> dir vlan
37  07/21/2004 17:38:24  { { shelf-1 first-control-module 2 } 50 }
```

The next commands modify the VLAN bridge circuit to enable Gigabit Ethernet redundancy for the connection:

```
admin> read vlan { { 1 8 2 } 50 }
admin> set interface-address = { { 1 0 2 } 50 }
(New index value; will save as new profile VLAN-ETHERNET/{ { shelf-1 any-
slot 2} 50 }.)
admin> write -f
```

The following commands list and then delete the older `vlan-ethernet` profile:

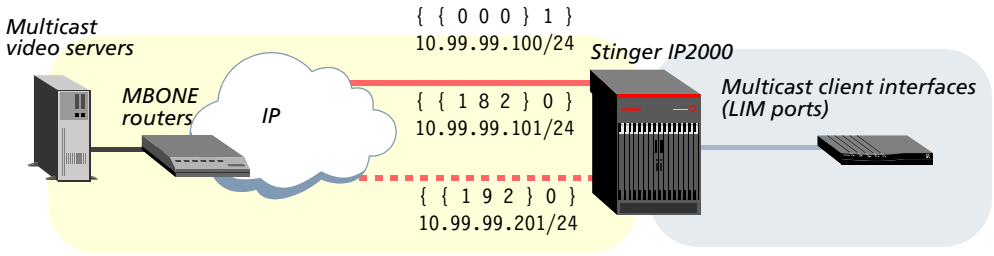
```
admin> dir vlan
37  07/15/2004 09:00:30  { { shelf-1 any-slot 2 } 50 }
37  07/21/2004 17:38:24  { { shelf-1 first-control-module 2 } 50 }
admin> delete vlan { {1 8 2 } } 50}
Delete profile VLAN-ETHERNET/{ { shelf-1 first-control-module 2 } 50 }?
[y/n] y
VLAN-ETHERNET/{ { shelf-1 first-control-module 2 } 50 } deleted
```

Configuring a redundant LAN MBONE

To support redundancy for a LAN MBONE interface, you must configure a soft IP interface for the Gigabit Ethernet ports, enable multicast on both ports, and use the `any-slot` expression in the `mbone-lan-interface` parameter setting.

Figure 2-2 shows a Stinger with redundant IP2000 controllers. The Gigabit Ethernet port in slot 8 ({ { 1 8 2 } 0 }), the Gigabit Ethernet port in slot 9 ({ { 1 9 2 } 0 }), and the soft IP interface ({ { 0 0 0 } 1 }), all have IP address assignments on the same subnet and both physical ports enable multicast.

Figure 2-2. Gigabit Ethernet redundancy for a LAN MBONE



The following commands configure the Gigabit Ethernet port in slot 8:

```
admin> read ip-interface { { 1 8 2 } 0 }
admin> set ip-address = 10.99.99.101/24
admin> set multicast-allowed = yes
admin> write -f
```

The next commands configure the Gigabit Ethernet port on slot 9:

```
admin> read ip-interface { { 1 9 2 } 0 }
admin> set ip-address = 10.99.99.201/24
admin> set multicast-allowed = yes
admin> write -f
```

The following commands configure a soft IP interface on the same subnet:

```
admin> new ip-interface { { 0 0 0 } 1 }
admin> set ip-address = 10.99.99.100/24
admin> write -f
```

The following commands enable the multicast forwarding function and specify a redundant LAN MBONE configuration:

```
admin> read ip-global
admin> set multicast-forwarding = yes
admin> set multiple-mbone mbone-lan-interface 1 = { { 1 0 2 } 0 }
admin> write -f
admin> list multiple-mbone mbone-lan-interface
[in IP-GLOBAL:multiple-mbone:mbone-lan-interface]
mbone-lan-interface[1] = { { shelf-1 any-slot 2 } 0 }
mbone-lan-interface[2] = { { any-shelf any-slot 0 } 0 }
mbone-lan-interface[3] = { { any-shelf any-slot 0 } 0 }
mbone-lan-interface[4] = { { any-shelf any-slot 0 } 0 }
```

Note With this configuration, the LAN MBONE is supported on the Gigabit Ethernet port of the controller in slot 8 or slot 9, whichever is primary. Following a switchover, each IGMP client must rejoin its group to receive multicast traffic.

Administrative tools for Gigabit Ethernet

The system supports the `gmac` command for administrative information about Gigabit Ethernet ports. If you are managing the system remotely, some of this information is also available through the `ip2kstats` MIB. For details, see "Gigabit Ethernet diagnostics" on page A-2 and "SNMP MIB for GMAC and VLAN statistics" on page A-24. For other commands that can be used to monitor activity on any Ethernet port, such as `etherdisplay`, see the *Stinger Reference*.

VLAN Configuration

3

This chapter describes configuration of virtual LAN (VLAN) operations and traffic characteristics. For detailed background information about VLAN, see *IEEE standard 802.1Q (1998) for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks*. In this chapter, the following VLAN terminology is used:

Table 3-1. Definition of VLAN terms

VLAN term	Definition
VLAN bridging (1:1)	A bridging configuration between a single PVC and a VLAN, with a 1:1 mapping. The setup uses `vlan-circuit` bridging.
VLAN bridging (N:1)	A bridging configuration between multiple PVCs and a VLAN. The setup uses `transparent` bridging and a bridge group.[a]
Stacked VLAN	A bridging configuration that encapsulates a VLAN within another VLAN to greatly increase the VLAN space. A stacked VLAN uses double tagging, where one 802.1Q tag represents a service provider (NSP) and a second 802.1Q tag represents an ID that is unique within the NSP VLAN.
Routed VLAN	A routing configuration in which the upstream IP interface is VLAN enabled. A routed VLAN interface is always mapped to a virtual IP interface on the IP2000 Gigabit Ethernet port. Packets received on a routed VLAN interface are routed based on the IP address, and packets are sent through the routed VLAN interface based on an IP routing decision.

a. A bridge group can specify traffic characteristics such as port blocking between DSL users belonging to the same VLAN, layer 2 multicast with IGMP snooping, and MAC address aging from bridge tables.

VLAN is supported with an optional software license. Enter the following command to determine whether the VLAN license is enabled:

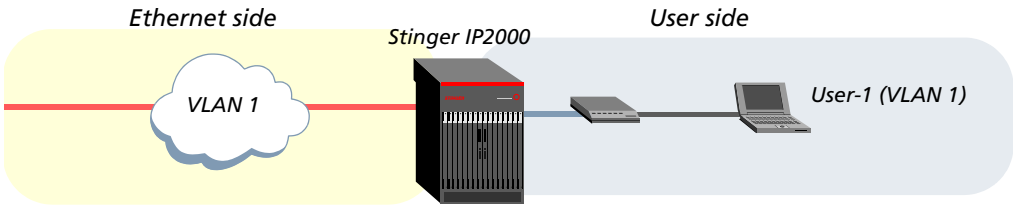
```
admin> get base vlan
[in BASE]
vlan-enabled = yes
```

The system sets this parameter to `yes` when the VLAN license is enabled. If the license is not enabled, the system displays an error message if you configure VLAN. For information about obtaining and enabling Lucent Technologies software licenses, contact your Lucent sales representative.

Configuring 1:1 VLAN bridging

This section describes how to configure `vlan-circuit` bridging between one user ATM PVC and one VLAN on Gigabit Ethernet, as shown in Figure 3-1. Because a single user-side PVC is bridged to the VLAN, source MAC address learning is not applicable.

Figure 3-1. Bridging VLAN: One PVC to one VLAN (1:1)



The user CPE encapsulates data using RFC 2684 encapsulation for bridged protocols. The system bridges the frames received on the user PVC only to the paired VLAN, and vice versa.

To configure `vlan-circuit` bridging, complete the following steps:

1 Create a `vlan-ethernet` profile on Gigabit Ethernet and assign a bridge group number.

2 Create a `connection` profile and specify the same bridge group.

Overview of `vlan-ethernet` and `connection` settings

The index of a `vlan-ethernet` profile specifies the physical address of the Gigabit Ethernet port and a unique VLAN ID. Following are the profile contents, shown for VLAN ID 50:

```
[in VLAN-ETHERNET/{ { shelf-1 first-control-module 2 } 50 }]
interface-address* = { { shelf-1 first-control-module 2 } 50 }
vlan-id = 0
enabled = no
filter-name = ""
pppoe-options = { no no }
bridging-options = { 0 no no vlan-circuit 0 }
```

```
[in VLAN-ETHERNET/{ { shelf-1 first-control-module 2} 50}:bridging-options]
bridging-group = 0
bridge = no
bridge-type = vlan-circuit
mac-address-learning-limit = 16
```

The following `connection` settings, shown with default settings, are required for subscribers to bridge to the configured VLAN:

```
[in CONNECTION/"":bridging-options]
bridging-group = 0
bridge = no
bridge-type = vlan-circuit
mac-address-learning-limit = 16
```

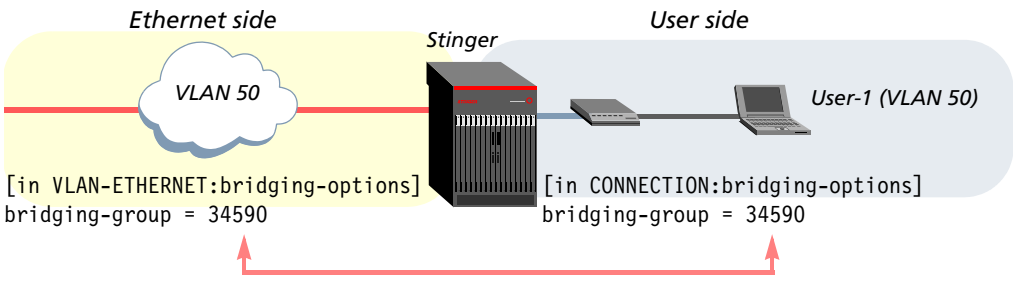
Parameter	Setting
`interface-address`	Address of the Gigabit Ethernet port followed by the VLAN ID, using the following format: { { *shelf-n slot-n port-n* } *vlan-id* } The *slot-n* is `first-control-module` or `second-control-module`, depending on the slot in which the active control module is installed, and *port-n* is 2 for the Gigabit Ethernet port. The *vlan-id* value is the IEEE 802.1Q VLAN tag value added to the IP packets transmitted on the Ethernet interface. The valid range is from 0 to 4095, but for full compatibility with IEEE 802.1Q, Lucent recommends that you do not use the *vlan-id* values of 0, 1 or 4095. However, the system does not prevent you from assigning these values.
`vlan-id`	VLAN ID. This setting is read-only. You must set it in the index of the `vlan-ethernet` profile.
`enabled`	Enable/disable the `vlan-ethernet` profile.
`filter-name`	*Not currently supported.*
`pppoe-options`	*Not currently supported.*
`bridging-group`	Number from 0 to 65535, used to group bridged interfaces. For 1:1 VLAN bridging, this setting must match in the `vlan-ethernet` and `connection` profiles. For N:1 VLAN bridging, this setting must match in the `bridge-group`, `vlan-ethernet` and `connection` profiles.
`bridge`	If enabled, associates the port with the specified `bridging-group`.

Parameter	Setting	
`bridge-type`	Type of bridging to use on the VLAN or WAN interface:	
	`vlan-circuit`	Required setting for 1:1 VLAN bridging (the default).
	`transparent-bridging`	Required setting for N:1 VLAN bridging.
	`stacked-vlan`	Required setting for stacked VLAN bridging.
	`no-bridging`	Required setting for routed VLANs.
`mac-address-learning-limit`	Maximum number of MAC addresses the system will learn across the interface. With a zero (0) setting, no limit is set in software. This is the default setting on a VLAN interface. With a nonzero value, the system adds only the specified number of addresses to the bridge table. The maximum number of MAC addresses the system will learn on a DSL or VLAN interface is 1024. In a `connection` profile, a change to the `mac-address-learning-limit` setting takes effect when you write the profile. The connection is bounced to instate the new limit. However, a change to this parameter in the `vlan-ethernet` side requires that you manually disable the profile and then reenable it for the new value to take effect. The `bridge-type` parameter must be set to `transparent-bridging` for this setting to take effect. For more details, see "How address limiting works" on page 3-10.	

Sample 1:1 VLAN bridging configuration

Figure 3-2 shows a Stinger system bridging a PVC to a VLAN:

Figure 3-2. Sample 1:1 VLAN circuit



To configure the VLAN circuit, first verify that bridging is enabled on the physical interface. For example:

```
admin> get ethernet { 1 8 2 } bridging-enabled
[in ETHERNET/{ shelf-1 first-control-module 2}:bridging-enabled]
bridging-enabled = yes
```

If bridging is not enabled, enable it as described in "Enabling layer 2 bridging for VLAN operations" on page 2-3. Then, follow these steps:

1 Create a `vlan-ethernet` profile.

```
admin> new vlan-ethernet { { 1 8 2 } 500 }
admin> set enabled = yes
admin> set bridging-options bridging-group = 34590
admin> write -f
```

2 Create a `connection` profile for the PVC side of the VLAN circuit.

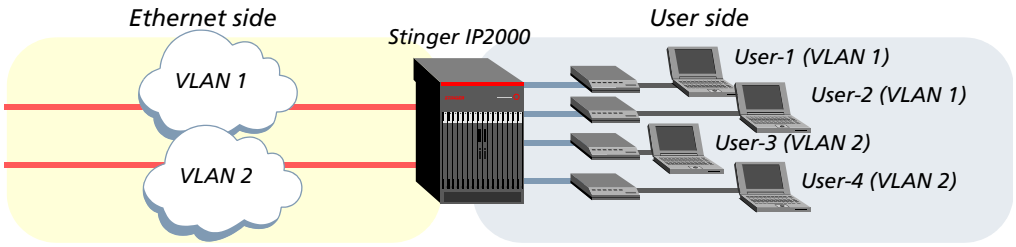
```
admin> new connection dslcpe
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options ip-routing-enabled = no
admin> set bridging-options bridging-group = 34590
admin> set bridging-options bridge = yes
admin> set atm-options vpi = 8
admin> set atm-options vci = 100
admin> set atm-options nailed-group = 51
admin> write -f
```

For background information about configuring PVCs, see the *Stinger ATM Configuration Guide*.

Configuring N:1 VLAN bridging

This section describes how to bridge multiple user PVCs onto a VLAN, as shown in Figure 3-3.

Figure 3-3. Bridging multiple PVCs to a VLAN



The user CPE encapsulates data using RFC 2684 encapsulation for bridged protocols, and the system bridges the frames to the associated VLAN (and vice versa), performing transparent bridging to build a table of known MAC addresses and the port associated with each address. If the system receives packets for an unknown

MAC address, or if it receives broadcast packets, the traffic is forwarded on all ports that are part of the bridge group.

A `bridge-group` profile defines the traffic characteristics of the VLAN and assigns a `bridge-group` number that is shared by the VLAN configuration and all user PVCs that are part of it.

To configure VLAN bridging, complete the following steps:

1 Create a `bridge-group` profile that defines traffic characteristics for the VLAN and assigns its `bridging-group` number.

2 Create a `vlan-ethernet` profile on the Gigabit Ethernet port and bind it to the `bridge-group` profile.

3 Create `connection` profiles for user PVCs, and bind them to the same bridge group.

Creating and configuring bridge groups

To define a limited broadcast domain for a bridged VLAN with multiple subscriber interfaces, the system requires bridge groups. Each VLAN must have a unique bridge group number, which is shared by all interfaces within the same VLAN.

The system creates a bridging table that contains only interfaces in the same bridge group, and when it receives packets from one interface in the group, it consults only that bridging table for destination ports. It will not forward the traffic to interfaces that are not in the same bridge group.

To optimize its forwarding operations over time, the system uses an IEEE 802.1 transparent bridging algorithm to build a table of known MAC addresses and the port associated with each address. If it receives packets for an unknown MAC address, or if it receives broadcast packets, the traffic is forwarded on all ports that are part of the bridge group except the port on which the packets were received.

Overview of **`bridge-group`** settings

The `bridge-group` profile defines traffic characteristics for VLAN bridging. The index of a `bridge-group` profile specifies a `bridging-group` number. Following are the `bridge-group` parameters, shown with default settings:

```
[in BRIDGE-GROUP/0]
enable = no
bridging-group = 0
mac-entry-age-time = 300
igmp-snooping = no
port-block-enabled = yes
lan-router-interface-address = { { any-shelf any-slot 0 } 0 }
wan-router-interface-profile = ""
```

Parameter	Setting
`enable`	Enables or disables the `bridge-group` profile. Set this parameter to `yes`.

Parameter	Setting
`bridging-group`	Number from 0 to 65535, used to group bridged interfaces. The same number specified here must also be specified as the `bridging-group` number in the `vlan-ethernet` and `connection` profiles.
`mac-entry-age-time`	Timeout interval (in seconds) at which the system ages out inactive MAC addresses from the bridge group's bridging table. Valid values are from 0 (which disables the address aging function) to 65535. The default is 300 seconds. When this timer expires, the system traverses all source MAC addresses in the group's bridge table and deletes those addresses from which no traffic has been received since the last traversal. A maximum of 32K addresses can be aged out systemwide. If you modify this value for a bridge group that is in use, the new value is used for the next timeout.
`igmp-snooping`	Enables or disables IGMP snooping. When IGMP is disabled (the default), multicast data streams are forwarded to all ports in the VLAN, even those who have not registered for the multicast. When IGMP snooping is enabled and a Join is received from a subscriber interface, the system snoops the packet and makes an entry in its bridge table, along with the IP multicast address. This Join is forwarded only on the "router" interface and not on the other interfaces in the bridge group. To use IGMP snooping, set this parameter to `yes`, and configure the `lan-router-interface-address` or `wan-router-interface-address` parameter.
`port-block-enabled`	Enables or disables port blocking to prevent traffic flows between subscriber interfaces in the same VLAN. Port blocking is enabled by default, but to use it you must configure the `lan-router-interface-address` or `wan-router-interface-address` parameter.
`lan-router-interface-address`	Index of the `vlan-ethernet` profile to be used as the "router" interface in terms of traffic handling for this bridge group. The specified interface is the default path for uplinks from users, and is used in port blocking to prevent other, unintended uses of the subscriber PVC interfaces. If IGMP snooping is enabled, the index must specify the Ethernet or VLAN interface on which downstream multicast streams are received. This setting and the `wan-router-interface-address` setting are mutually exclusive.

Parameter	Setting
`wan-router-interface-profile`	Name of the `connection` profile to be used as the "router" port in terms of traffic handling for this bridge group. The specified connection must be on a trunk interface. It is the default path for uplinks from users, and is used in port blocking to prevent other, unintended uses of the subscriber PVC interfaces. If IGMP snooping is enabled, the `connection` profile name must specify the trunk interface on which downstream multicast streams are received. This setting and the `lan-router-interface-address` setting are mutually exclusive.

Sample **`bridge-group`** configuration with MAC address aging

The following commands modify `bridge-group` 451 to extend the duration of the timer for discovering and dropping inactive source MAC addresses from its bridge tables from the default five minutes (300 seconds) to 10 minutes.

```
admin> read bridge-group 451
admin> set mac-entry-age-time = 600
admin> write -f
```

Sample **`bridge-group`** configuration with port blocking

Port blocking prevents users who are bridged to the same LAN or VLAN from exchanging traffic flows user-to-user or from building virtual networks. To use it, you must identify one "router" interface for the bridge group, which can be a VLAN interface or a `connection` profile on a trunk interface. All other ports in the same bridge group are then considered subscriber interfaces for traffic handling purposes. Traffic received on the "router" and subscriber interfaces in a bridge group is restricted as shown in Table 3-2.

Table 3-2. Traffic restrictions when port blocking is enabled

Received on "router" port	Received on subscriber ports
No restrictions are placed on traffic received on the "router" interface.	Packets received on a DSL port and destined for another DSL port in the same bridge group are discarded. This prevents users from building virtual networks using the VLAN, or for making user-to-user connections through the VLAN.
	Broadcast packets are forwarded only to the "router" interface.
	Packets destined for a MAC address the system does not recognize are forwarded only to the "router" interface.

The next commands modify `bridge-group` 275 (VLAN 500) to use DSL port blocking, by specifying a "router" interface:

```
admin> read bridge-group 275
admin> set lan-router-interface-address shelf = 1
admin> set lan-router-interface-address slot = 8
admin> set lan-router-interface-address item = 2
admin> set lan-router-interface-address logical-item = 500
admin> write -f
```

Sample **bridge-group** configuration with IGMP snooping

Layer 2 multicasting maps an IP multicast traffic to a MAC multicast address, which is treated at layer 2 as broadcast traffic. Because broadcast traffic would typically be forwarded to all interfaces in the VLAN, IGMP snooping is implemented on a bridge group basis to enable efficient support for layer 2 multicasting to VLANs.

IGMP snooping "peeks" into the layer 3 content of multicast packets, and allows the IP2000 module to forward multicast traffic for a particular group only to those user PVCs that have registered in the group.

When IGMP snooping is enabled within a bridge group, all IGMP packets received on an interface in that bridge group are snooped, and multicast forwarding is done on the basis of the multicast group address and not on the basis of a multicast MAC address.

Multicast data traffic is forwarded only to the users subscribed to the particular multicast group. Table 3-3 shows how IGMP control packets are handled.

Table 3-3. IGMP control packet handling with IGMP snooping enabled

Control packet type	System action
IGMP-QUERY	Generic queries received from the "router" interface are forwarded to all DSL users.
	Group-specific queries received from the "router" interface are forwarded to users who have joined that group.
IGMP-REPORT(v1/v2)	Reports received from users are forwarded only to the "router" interface.

Note The Stinger system does not generate any IGMP queries or reports. It is up to the upstream router to generate queries and handle the reports, and it is up to the end users to send reports.

The following commands modify `bridge-group` 22 (VLAN 478) to enable IGMP snooping. A designated "router" interface is required for this feature to work:

```
admin> read bridge-group 22
admin> set igmp-snooping = yes
admin> set lan-router-interface-address shelf = 1
admin> set lan-router-interface-address slot = 8
admin> set lan-router-interface-address item = 2
```

```
admin> set lan-router-interface-address logical-item = 478
admin> write -f
```

When IGMP snooping is enabled and a Join is received from a subscriber interface, the system snoops the packet and makes an entry in its bridge table, along with the IP multicast address. This Join is forwarded only on the "router" interface and not on the other interfaces in the bridge group.

Similarly, when a multicast data packet arrives from the LAN router interface, it is snooped, checked for the IP multicast address and forwarded only on subscriber interfaces from which a Join was received.

If both IGMP snooping and port blocking are enabled and the system receives traffic for `224.0.0.*` from the "router" interface, it is forwarded on all the ports in the bridge group. However, if traffic for `224.0.0.*` is received from a subscriber interface, it is forwarded only to the "router" interface.

VLAN and connection settings

For details about the VLAN and bridging parameters used in the `vlan-ethernet` and `connection` profiles for an N:1 VLAN bridging configuration, see "Overview of vlan-ethernet and connection settings" on page 3-2. For background information about configuring subscriber PVCs, see the *Stinger ATM Configuration Guide*.

How address limiting works

Limiting the number of source MAC addresses learned on a transparent bridging interface restricts the number of users that can access the network through a single CPE, and prevents a type of denial-of-service attack in which a user overloads the bridge table by sending heavy traffic from many different source MAC addresses. When the address limit has been reached on a connection, the system logs a message to that effect, to enable the administrator to check for denial-of-service attacks or, if the connection attempts are valid, to reconfigure the limit on a connection.

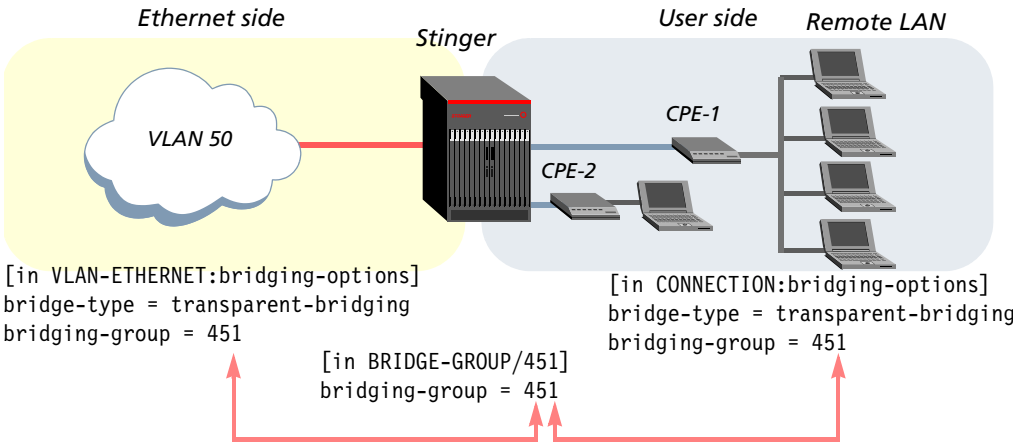
The transparent bridging algorithm enables the system to learn the MAC addresses of devices sending traffic across an interface and enter the addresses in an internal bridge table. With the MAC address learning limit, you can limit the number of addresses the system will learn across the connection. If the system receives traffic from an unknown source address after the system has learned the maximum number of MAC addresses, it discards the traffic and does not add the source address to the bridge table.

If MAC address aging is also enabled, when one of the devices on an interface becomes inactive for a specified interval, the system ages the address out of its bridge table. As soon as the bridge table contains less than the maximum number of addresses for the interface, the system can again accept traffic from an unknown source MAC address on that interface and add the address to its bridge table. For details about address aging, see "Sample bridge-group configuration with MAC address aging" on page 3-8.

Sample N:1 VLAN bridging configuration with address limiting

Figure 3-4 shows a Stinger system bridging multiple user PVCs onto a VLAN. MAC address limiting will be enforced for CPE-1.

Figure 3-4. Sample N:1 VLAN bridging



With the sample configuration shown below, the system learns up to three MAC addresses on the CPE-1 connection. For the first three PCs that send traffic, the system forwards the traffic and learns the source MAC addresses. If the fourth PC attempts to send traffic, the user's connection is rejected. Later, if one of the bridged PCs does not send traffic for a duration equal to the `mac-entry-age-time` setting, the system ages out that MAC address from its bridge table. If the fourth PC sends traffic at that time, the system learns its address and forwards the traffic.

To configure this N:1 VLAN bridging setup, first verify that bridging is enabled on the physical interface. For example:

```
admin> get ethernet { 1 8 2 } bridging-enabled
[in ETHERNET/{ shelf-1 first-control-module 2}:bridging-enabled]
bridging-enabled = yes
```

If bridging is not enabled, enable it as described in "Enabling layer 2 bridging for VLAN operations" on page 2-3. Then, follow these steps:

1 Create a bridge group. The following group sets the address age-out interval to three minutes and uses DSL port blocking. (For more detail, see "Creating and configuring bridge groups" on page 3-6.)

```
admin> new bridge-group
admin> set enable = yes
admin> set bridging-group = 451
admin> set lan-router-interface-address shelf = 1
admin> set lan-router-interface-address slot = 8
admin> set lan-router-interface-address item = 2
admin> set lan-router-interface-address item = 50
admin> set mac-entry-age-time = 180
admin> write -f
```

2 Create a `vlan-ethernet` profile.

```
admin> new vlan-ethernet
admin> set interface-address physical-address shelf = 1
admin> set interface-address physical-address slot = 8
admin> set interface-address physical-address item-number = 2
admin> set interface-address logical-item = 50
admin> set enabled = yes
admin> set bridging-options bridging-group = 451
admin> set bridging-options bridge = yes
admin> set bridging-options bridge-type = transparent-bridging
admin> write -f
```

3 Create `connection` profiles for bridged PVCs to the CPE devices. The profiles must specify the right bridge group number. The profile for CPE-1 also limits the number of MAC addresses to three.

```
admin> new connection cpe-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options ip-routing = no
admin> set bridging-options bridge = yes
admin> set bridging-options bridging-group = 451
admin> set bridging-options bridge-type = transparent-bridging
admin> set bridging-options mac-address-learning-limit = 3
admin> set atm-options vpi = 0
admin> set atm-options vci = 36
admin> set atm-options nailed-group = 51
admin> write -f
admin> new connection cpe-2
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options ip-routing = no
admin> set bridging-options bridge = yes
admin> set bridging-options bridging-group = 451
admin> set bridging-options bridge-type = transparent-bridging
admin> set atm-options vpi = 0
admin> set atm-options vci = 38
admin> set atm-options nailed-group = 57
admin> write -f
```

Configuring stacked VLANs

VLAN stacking is a method of encapsulating one VLAN within another VLAN. It allows a carrier to partition the network among several network service providers (NSPs), while allowing each NSP to utilize VLANs to their full extent.

Each NSP can be assigned one or more VLANs ("backbone VLANs" or "NSP VLANs"), and within each NSP VLAN, up to 4095 unique instances of 802.1Q VLAN IDs are available, with each ID representing an ATM PVC from a DSL subscriber.

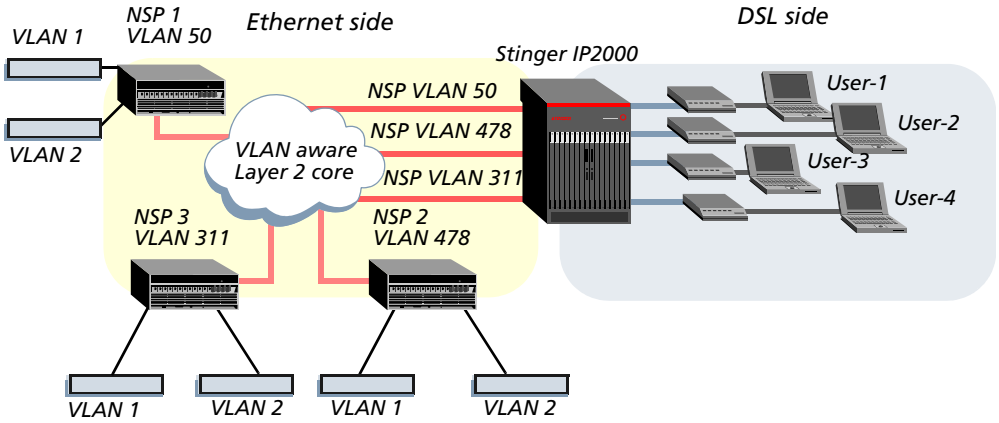
Stinger IP2000 systems support VLAN stacking for both untagged Ethernet frames and VLAN tagged traffic received on subscriber interfaces. The two methods differ in terms of subscriber `connection` profiles.

Note For stacked VLAN connections, the Stinger IP2000 does not bridge frames received from one DSL connection to another, even when the connections are configured with the same `bridging-group` value. This applies even to broadcast and multicast frames. Broadcast frames received from WAN interfaces are tagged with both the NSP VLAN and user VLAN ID and bridged to the appropriate NSP VLAN. Broadcast frames received from Gigabit Ethernet interface with a single VLAN tag are not bridged to the WAN interfaces.

Bridging untagged frames to stacked VLANs

In the sample stacked VLAN setup shown in Figure 3-5, a layer-2 core is partitioned among three NSPs.

Figure 3-5. Stacked VLAN: Bridging untagged frames from DSL interfaces



When the system receives untagged frames from an ATM PVC on a DSL interface, it tags the frames with the user's VLAN ID. This VLAN tag represents the user's connection within the context of an NSP. The system then embeds the tag in another, second-tier VLAN ID, which represents the user's NSP. The Ethernet frame from the ATM PVC is then bridged over the Gigabit Ethernet interface.

When system receives VLAN stacked frames on the Gigabit Ethernet interface, the NSP and subscriber VLAN tags are used to determine the DSL interface on which to bridge the frames. The two VLAN tags are removed before bridging the packet onto the WAN interface.

Overview of VLAN stacking settings for untagged frames

Each NSP requires one or more VLAN configurations on the Gigabit Ethernet interface, and each DSL subscriber requires a `connection` profile with a VLAN ID that is unique within the context of the user's destination NSP VLAN. Following are the parameters, shown with default settings, for VLAN stacking of untagged frames:

```
[in VLAN-ETHERNET/{ { shelf-1 first-control-module 2} 50}:bridging-options]
bridge-type = vlan-circuit

[in CONNECTION/"":bridging-options]
bridge-type = vlan-circuit
vlan-stack-user-vlan-id = 0

[in ETHERNET/{ shelf-1 second-control-module 2 }]
vlan-stack-tag-type = 91:00
```

Parameter	Setting
`bridge-type`	Type of bridging. Valid values are `transparent-bridging`, `no-bridging`, `vlan-circuit`, and `stacked-vlan`. For VLAN stacking, the `stacked-vlan` setting is required.
`vlan-stack-user-vlan-id`	Subscriber's 802.1Q VLAN ID (from 0 to 4095) to be used in stacked-VLAN frames for incoming traffic that contains untagged Ethernet frames. The value must be unique within the NSP VLAN. With the default zero value, VLAN stacking is disabled for the connection. When set to a nonzero value, VLAN stacking is enabled and the specified value is added to the frames as a VLAN tag that represents the user's connection within the context of an NSP. This parameter is applicable only in `connection` profiles. Although it also appears in both `ethernet` and `vlan-ethernet` profiles, the field is not applicable to those contexts.
`vlan-stack-tag-type`	Two-byte hexadecimal value to be inserted in the EtherType field for stacked-VLAN frames. All stacked VLAN frames being transmitted/received on the Ethernet port will use this value in their EtherType field. The default value is `0x9100`. Because VLAN stacking is not yet a standardized technology, an EtherType value has not been standardized to represent stacked-VLAN frames, unlike the value 0x8100 used for IEEE 802.1Q VLAN frames, for example. So for the sake of interoperability, this value is configurable. If the layer-2 core network is using a specific EtherType value for stacked VLAN frames, you must set the `vlan-stack-tag-type` parameter to that value.

Sample configuration bridging untagged frames

The following example configures an NSP VLAN for "NSP 1" in Figure 3-5 (page 3-13), and two bridged PVCs to be directed to the NSP VLAN. In this example, the NSP VLAN is assigned VLAN number 50, and the user VLAN IDs are 471 and 473 within the NSP VLAN.

The following sets of sample commands configure a VLAN with multiple user PVCs.

1 Enable bridging on the Gigabit Ethernet port, as described in "Enabling layer 2 bridging for VLAN operations" on page 2-3. (If the layer-2 core network is using a specific EtherType value for stacked VLAN frames, you must also set the `vlan-stack-tag-type` parameter to that value.)

2 Create a `vlan-ethernet` profile for the NSP VLAN. In this example, the VLAN is assigned VLAN ID 50 and `bridging-group` 9.

```
admin> new vlan-ethernet { { 1 8 2 } 50 }
admin> set enabled = yes
admin> set bridging-options bridging-group = 9
admin> set bridging-options bridge = yes
admin> set bridging-options bridge-type = stacked-vlan
admin> write -f
```

3 Create `connection` profiles for users of the NSP VLAN. These profiles must use the same `bridging-group` number as the NSP VLAN, and must specify a nonzero `vlan-stack-user-vlan-id` value that is unique within the NSP VLAN. In this example, the user VLAN IDs are 471 and 473.

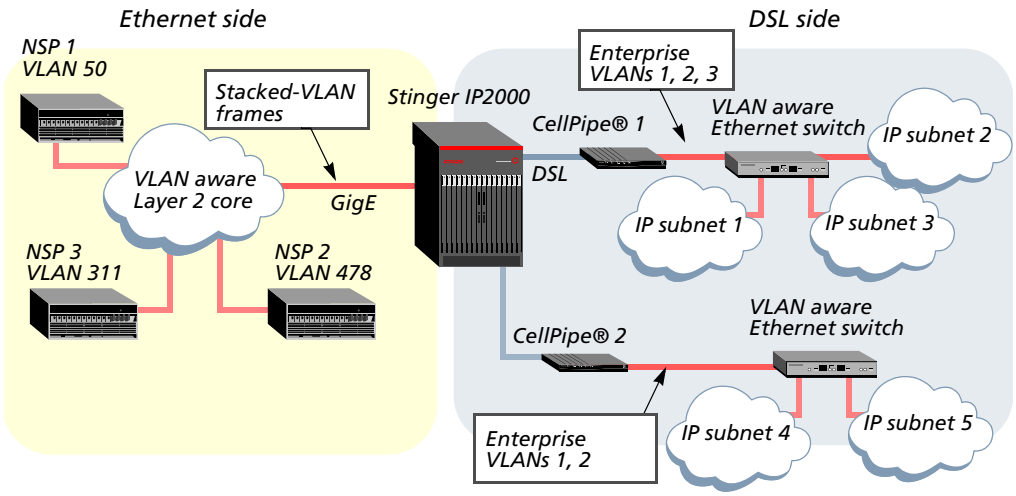
```
admin> new connection vlan-user-1
admin> set encapsulation-protocol = atm
admin> set ip-options ip-routing-enabled = no
admin> set bridging-options bridging-group = 9
admin> set bridging-options bridge = yes
admin> set bridging-options bridge-type = stacked-vlan
admin> set bridging-options vlan-stack-user-vlan-id = 471
admin> set atm-options vci = 60
admin> set atm-options nailed-group = 125
admin> write -f
admin> new connection
admin> set station = vlan-user-2
admin> set encapsulation-protocol = atm
admin> set ip-options ip-routing-enabled = no
admin> set bridging-options bridging-group = 9
admin> set bridging-options bridge = yes
admin> set bridging-options bridge-type = stacked-vlan
admin> set bridging-options vlan-stack-user-vlan-id = 473
admin> set atm-options vci = 60
```

```
admin> set atm-options nailed-group = 127
admin> write -f
```

Bridging enterprise VLAN tagged frames to stacked VLANs

Stinger systems also support bridging of VLAN-tagged traffic received from the DSL side to stacked VLANs. The sample VLAN setup in Figure 3-6 (page 3-16)shows a Stinger IP2000 with three stacked VLAN configurations on the Gigabit Ethernet interface.

Figure 3-6. Stacked VLAN: Bridging enterprise VLAN-tagged frames



The CellPipe® units are operating in LLC bridged mode, and interacting with an enterprise VLAN-aware Ethernet switch in which enterprise VLANs are defined. Each enterprise IP subnet maps to a unique VLAN ID within the context of an individual DSL connection, and the Stinger IP2000 bridges the data between DSL interfaces and the Gigabit Ethernet backbone.

In the Stinger IP2000, the basic VLAN stacking configuration on the Ethernet side does not change, but another layer of complexity is required to accommodate tagged frames from the DSL side.

Instead of adding a `vlan-stack-user-vlan-id` tag (as it does for untagged frames), when VLAN tagged frames are received from a DSL interface, the system must change the incoming VLAN tag to a VLAN user ID (and optionally, a new priority value) that represents the user's connection within the context of an NSP. The system then embeds the new (modified) VLAN information in another, second-tier VLAN tag, which represents the user's NSP. The Ethernet frame is then bridged over the Gigabit Ethernet interface.

The `vlan-stack-tag-type` parameter applies exactly as described for untagged frames in "Overview of VLAN stacking settings for untagged frames" on page 3-14.

Overview of VLAN stacking settings for tagged frames

Each NSP requires one or more VLAN configurations on the Gigabit Ethernet interface. To enable the system to modify VLAN tags in inbound tagged frames, the DSL `connection` profile must specify a `flow-services` profile that defines the

mapping between enterprise priorities and VLAN IDs in inbound traffic and NSP priorities and VLAN user IDs for outbound traffic on the Gigabit Ethernet port. The profile can also be configured to influence switching or forwarding decisions on the basis of traffic flows. Each flow of traffic through an interface can be treated differently based on a set of flow classification rules.

Following are parameters, shown with default settings, for VLAN stacking of tagged frames:

```
[in FLOW-SERVICES/""]
name* = ""
service-type = none
flow-list = [ { { 0 } { 0 } } { { 0 } { 0 } } { { 0 } { 0 } } { { 0 } { 0 } }+

[in FLOW-SERVICES/"":flow-list]
flow-list[1] = { { 0 0 00:00 } { 0 } }
...
flow-list[32] = { { 0 0 00:00 } { 0 } }

[in FLOW-SERVICES/"":flow-list[1]]

layer2-classifier = { 0 0 00:00 }
service-options = { 0 }

[in FLOW-SERVICES/"":flow-list[1]:layer2-classifier]
vlan-id = 0
ethernet-priority = 0
ethernet-type = 00:00

[in FLOW-SERVICES/"":flow-list[1]:service-options]
vlan-id = 0

[in CONNECTION/"":session-options]
flow-services = ""
```

Parameter	Setting	
`name`	Name of this `flow-services` profile, up to 31 characters. The name is used to identify the profile and apply it to a `connection` profile.	
`service-type`	Type of service for the traffic flow of the connection.	
	`none`	With the default value of `none`, the profile has no effect on traffic flows if applied to a `connection` profile.
	`vlan-stack-user-tag`	Enable the system to map the VLAN tag in the incoming traffic flow to a different VLAN user ID for stacked VLANs.

Parameter	Setting	
`flow-list`	An array of 32 indexed subprofiles used to configure a set of flow classification rules and corresponding service parameters applicable to those flows. The settings in the subprofiles are interpreted on the basis of the `service-type` setting. In a profile for tagged frames, the `flow-list` subprofiles must be configured in order (without a gap in the sequence of subprofile index numbers) beginning with `flow-list` 1, followed by `flow-list` 2, and so forth until all required `flow-list` subprofiles have been specified.	
`layer2-classifier`	A `flow-list` subprofile for defining classification rules to identify a specific layer 2 traffic flow.	
	`vlan-id`	Enterprise VLAN ID in the VLAN-tagged frames of the inbound traffic flow (a number from 0 to 4095). This is the "source" VLAN ID that will be mapped to another ID specified in the `service-options` subprofile.
	`ethernet-priority`	802.1Q/P priority value (p-bit value) of the frame's VLAN tag. A number from 0 to 7.
	`ethernet-type`	Ethernet type. A two-byte hexadecimal number representing the Ethernet protocol.
`service-options`	A `flow-list` subprofile for defining service-specific parameters applicable to the layer 2 traffic flow.	
	`vlan-id`	User VLAN ID (from 1 to 4095) to be inserted in stacked-VLAN frames. This is the "destination" VLAN ID that represents the user's connection within the context of an NSP, and must be unique within the NSP VLAN.
`flow-services`	Name of a `flow-services` profile that defines the proper VLAN tag mapping for this connection.	

Note In some cases, there may be a requirement to preserve the enterprise VLAN IDs received in incoming traffic and simply prefix them with the NSP VLAN ID. That can be done, as long as the enterprise VLAN IDs are unique within the context of the NSP VLAN, by specifying the enterprise VLAN ID in both the `layer2-classifier` and `service-options` subprofiles.

Sample configuration for mapping tagged frames

The following example configures an NSP VLAN for "NSP 2" in Figure 3-6 (page 3-16), and the bridged PVC from the unit labeled `CellPipe® 1`, to be directed to the NSP VLAN. The VLAN IDs in this example are shown in Table 3-4.

Table 3-4. Enterprise VLAN tagged frames mapped to a stacked VLAN

Enterprise VLAN ID	NSP VLAN ID	NSP VLAN user ID
1	478	1001
2	478	2002
3	478	3003

To configure the stacked VLAN and mapping between enterprise and NSP VLAN IDs, first verify that bridging is enabled on the physical interface. For example:

```
admin> get ethernet { 1 8 2 } bridging-enabled
[in ETHERNET/{ shelf-1 first-control-module 2}:bridging-enabled]
bridging-enabled = yes
```

If bridging is not enabled, enable it as described in "Enabling layer 2 bridging for VLAN operations" on page 2-3. Then, follow these steps:

1 Create a `vlan-ethernet` profile for the NSP VLAN. In this example, the VLAN is assigned VLAN ID 478 and `bridging-group` 30.

```
admin> new vlan-ethernet { { 1 8 2 } 478 }
admin> set enabled = yes
admin> set bridging-options bridging-group = 30
admin> set bridging-options bridge = yes
admin> set bridging-options bridge-type = stacked-vlan
admin> write -f
```

2 Create a `flow-services` profile that maps VLAN tags 1, 2, and 3 on the DSL interface to user ID values that are unique within the NSP VLAN. In this example, the user VLAN IDs are 1001, 2002, and 3003.

```
admin> new flow-services
admin> set name = enterprise1-vlans
admin> set service-type = vlan-stack-user-tag
admin> set flow-list 1 layer2-classifier vlan-id = 1
admin> set flow-list 1 service-options vlan-id = 1001
admin> set flow-list 2 layer2-classifier vlan-id = 2
admin> set flow-list 2 service-options vlan-id = 2002
admin> set flow-list 3 layer2-classifier vlan-id = 3
admin> set flow-list 3 service-options vlan-id = 3003
admin> write -f
```

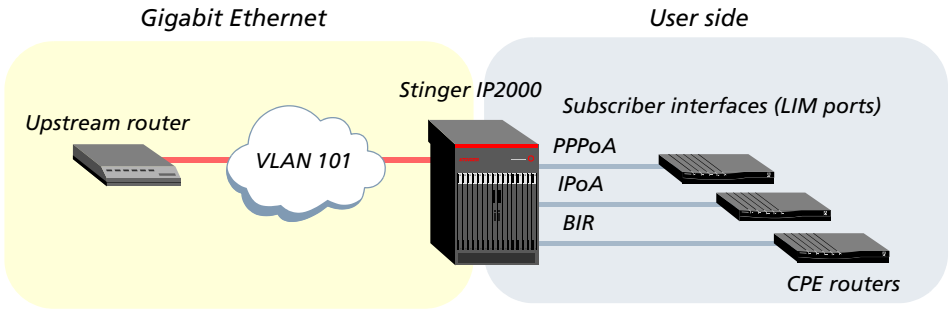
3 Create a `connection` profile for the inbound CellPipe® connection. The profile must use the same `bridging-group` number as the NSP VLAN, and must specify the `flow-services` profile.

```
admin> new connection cellpipe-1
admin> set encapsulation-protocol = atm
admin> set ip-options ip-routing-enabled = no
admin> set bridging-options bridging-group = 30
admin> set bridging-options bridge = yes
admin> set bridging-options bridge-type = stacked-vlan
admin> set session-options flow-services = enterprise1-vlans
admin> set atm-options vci = 60
admin> set atm-options nailed-group = 125
admin> write -f
```

Configuring routed VLANs

A routed VLAN interface is the interface to which the router's IP address on the VLAN is attached. In a Stinger IP2000 system, a routed VLAN interface is always mapped to a virtual IP interface on the Gigabit Ethernet port. Packets received on a routed VLAN interface are routed based on the IP address, and packets are sent through the routed VLAN interface based on an IP routing decision.

The system can route traffic received from a VLAN to any routed, terminated user connection, such as IP over ATM (RFC 2684), bridged IP routing (BIR), DHCP requests and responses, PPP over ATM (PPPoA), PPP over Ethernet (PPPoE).



To configure a routed VLAN, follow these steps:

1 Create terminated routed `connection` profiles for subscribers.

For details see Chapter 4, “IP Routing Configuration.”

2 Create a VLAN on the Gigabit Ethernet interface.

The `vlan-ethernet` parameters for routed VLAN are the same as for VLAN bridging (N:1). See “Overview of vlan-ethernet and connection settings” on page 3-2.

3 Create a virtual IP interface to the configured VLAN, specifying the VLAN ID.

Note You can enable multicast forwarding on the virtual IP interface, to allow it to handle both routed and multicast traffic. For details about that configuration, see "Configuring MBONE interfaces" on page 8-3.

Creating a virtual IP interface for a routed VLAN

Following are the parameters, shown with default settings for a virtual IP interface, for configuring a routed VLAN:

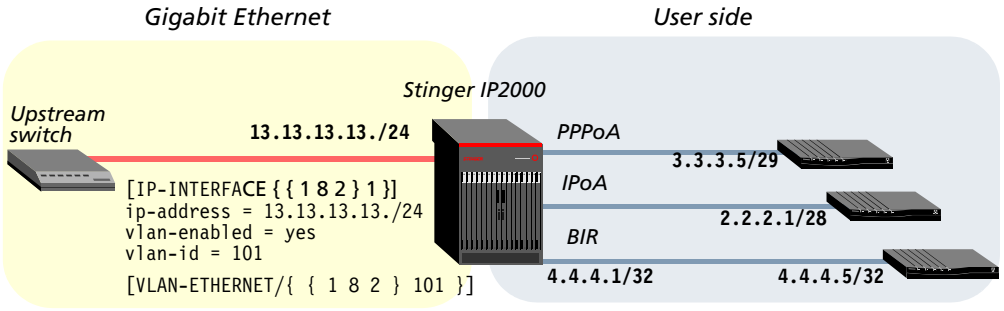
```
[in IP-INTERFACE/{ { shelf-1 first-control-module 2 } 1 }]
interface-address* = { { shelf-1 first-control-module 2 } 1 }
ip-address = 0.0.0.0/0
vlan-enabled = no
vlan-id = 0
```

Parameter	Setting
`interface-address`	Physical address of the Gigabit Ethernet interface in the system, followed by the virtual IP interface number, using the following format: { *shelf-n slot-n port-n* } *item-n* } The *slot-n* is `first-control-module` or `second-control-module`), and *port-n* is 2 for the Gigabit Ethernet port. The *item-n* value is a number from 1 to 16, identifying the virtual IP interface.
`ip-address`	IP address of the virtual IP interface.
`vlan-enabled`	Enables or disables IEEE 802.1Q VLAN tagging on the virtual IP interface. Set this parameter to `yes` for a routed VLAN interface.
`vlan-id`	VLAN ID for this virtual interface. This is the IEEE 802.1Q tag value to be added to the IP packets transmitted on the virtual interface. The valid range is from 0 to 4095, but for full compatibility with IEEE 802.1Q, Lucent recommends that you do not use the *vlan-id* values of 0, 1 or 4095.

Sample routed VLAN configuration

Figure 3-7 shows a basic representation of a routed VLAN configuration, which enables the system to handle a routed data stream with a VLAN tag, and forward the data stream to CPE routers based on the CPE IP address.

Figure 3-7. Sample routed VLAN



To configure the routed VLAN interface, follow these steps:

1 Create a `vlan-ethernet` profile. This example uses VLAN ID 101.

```
admin> new vlan-ethernet { { 1 8 2 } 101}
admin> set enabled = yes
admin> set bridging-options bridge-type = no-bridging
admin> write -f
```

2 Create an `ip-interface` profile with a valid IP address and VLAN enabled. Specify the same VLAN ID used in the `vlan-ethernet` profile (101 in this example).

```
admin> new ip-interface { { 1 8 2 } 1 }
admin> set ip-address = 13.13.13.13/24
admin> set vlan-enabled = yes
admin> set vlan-id = 101
admin> write -f
```

3 Configure the terminated PVCs. The system routes packet streams to the virtual IP interface associated with the VLAN, so the VLAN is used to carry routed traffic.

For example, the following commands configure a PPPoA, IPoA, and BIR connection.

```
admin> new connection pppoa-1
admin> set active = yes
admin> set encapsulation-protocol = ppp
admin> set ip-options remote-address = 3.3.3.5/29
admin> set ppp-options send-auth-mode = chap-ppp-auth
admin> set ppp-options bi-directional-auth = required
admin> set ppp-options send-password = sendpw
admin> set ppp-options recv-password = recvpw
admin> set telco-options call-type = off
admin> write -f
admin> new connection ipoa-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 2.2.2.1/28
```

```
admin> set atm-options vci = 37
admin> set atm-options nailed-group = 301
admin> write -f
admin> new connection bir-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 4.4.4.5/32
admin> set ip-options local-address = 4.4.4.1/32
admin> set atm-options atm1483type = aal5-llc
admin> set atm-options vci = 111
admin> set bir-options enable = yes
admin> set atm-options nailed-group = 155
admin> write -f
```

Administrative tools for VLAN

Commands that provide administrative information about VLAN are available only in the debug environment. If you are managing the system remotely, some of this information is also available through the `ip2kstats` MIB. For details, see "VLAN-related diagnostics" on page A-14 and "SNMP MIB for GMAC and VLAN statistics" on page A-24.

IP Routing Configuration

4

This chapter describes IP routing features that are typically configured on a Stinger IP2000. Some parameters in IP-related profiles are not used by the IP2000, or are not relevant to its primary applications. Those parameters are not described in this chapter, but are documented in the *Stinger Reference*.

Introduction to the IP router software

When you reset the system, an IP routing table is constructed that contains all the routes known to the system, including the following:

- Routes for the local Ethernet interfaces (configured `ip-interface` profiles)
- Routes for active WAN IP sessions
- Routes for inactive WAN IP sessions (configured `connection` profiles)
- Routes defined in `ip-route` profiles or RADIUS route profiles

If the Routing Information Protocol (RIP) is enabled on one or more interfaces, the system adds routes as it learns them from routing-update packets. In addition, the system is continuously updating its routing table by adding routes for links that become active and removing routes for inactive sessions. If a nailed connection goes down, the system removes the route from its routing table.

Routes and interfaces

An IP route specifies a destination address, a gateway to the network, and an interface that leads to the gateway. It can also specify metrics and other values associated with the route.

A route defined in a profile is a *static route*. A *dynamic route* is learned from RIP updates sent by other routers. Dynamic updates provide access to many more routes than those actually configured in the system, and are updated automatically as routes change. However, dynamic updates cause additional routing overhead, so they are disabled by default.

An *interface* is a point of ingress to or egress from the system. For example, a local interface is an Ethernet port and a WAN interface is a nailed or switched connection. An *IP interface* is the logical IP address that enables IP data to be sent and received.

Displaying the routing table

For details about the `netstat` command, see the *Stinger Reference*. The following command displays the system's routing table:

```
admin> netstat -r
Destination       Gateway         IF          Flg   Pref Met     Use        Age
0.0.0.0/0         1.112.26.1     ie0         SGP     60   1     343        2274
127.0.0.0/8       -               bh0         CP       0   0       0        2274
127.0.0.1/32      -               local       CP       0   0       0        2274
127.0.0.2/32      -               rj0         CP       0   0       0        2274
1.112.0.0/16      -               ie0         C        0   0    6497        2274
1.112.26.146/32   -               local       CP       0   0    3635        2274
224.0.0.0/4       -               mcast       CP       0   0     179        2274
224.0.0.1/32      -               local       CP       0   0       0        2274
224.0.0.2/32      -               local       CP       0   0       0        2274
224.0.0.9/32      -               local       CP       0   0       0        2274
255.255.255.255/32-               ie0         CP       0   0      61        2274
Total Routes = 11      Hidden Routes = 0
```

For each route in the table, the `Destination` and `Gateway` fields show the destination address and the address of the next-hop router used to reach that destination. The zero destination address is the default route. If the system does not find a route for a packet's destination, it forwards the packet to the default route rather than dropping the packet. Note that the system uses the most specific route (having the longest prefix) that matches a given destination. Direct routes do not show a gateway address.

An asterisk (*) in the flags column indicates a hidden route, which is not included in routing updates sent by the system and is not used for forwarding packets. Hidden routes are used only for display purposes.

The `IF` field shows the name of the interface through which a packet addressed to the entry's destination will be sent. The route to the `mcast` interface name encapsulates the multicast forwarder for the entire class D address space. (For more information, see Chapter 8, "IP Multicast Configuration.")

Routes to the local unit display the `local` interface name. Packets to the 224.0.0.1 and 224.0.0.2 interfaces can be multicast and received like normal multicast packets, but upon receiving such a packet, the router does not forward it to another link layer device. Effectively, these packets have a maximum transmission unit (MTU) of 1.

Displaying the interface table

To display the interface table, use the `–i` option on the `netstat` command line:

```
admin> netstat -i
Name       MTU   Net/Dest          Address         Ipkts   Ierr Opkts    Oerr
ie0        1500  1.112.0.0/16      1.112.26.146     5542    0     1636    0
ie1        1500  -                   -                 0    0        0    0
ie1-1      1500  -                   -                 0    0        0    0
lo0        1500  127.0.0.1/32        127.0.0.1      1629    0     1629    0
rj0        1500  127.0.0.2/32        127.0.0.2         0    0        0    0
bh0        1500  127.0.0.3/32        127.0.0.3         0    0        0    0
wanabe     1500  127.0.0.3/32        127.0.0.3         0    0        0    0
local     65535  127.0.0.1/32        127.0.0.1      1892    0     1892    0
mcast     65535  224.0.0.0/4         224.0.0.0       180    0      180    0
tunnel0    1500   1.112.0.0/16      1.112.26.146       0    0        0    0
vr0_main   1500   1.112.26.146/32   1.112.26.146       0    0        0    0
sip0      65535  -                   -                 0    0        0    0
```

The entries named `ie0`, `ie1`, or `ie1-`*N* represent Ethernet interfaces, where *N* represents the logical-item number of the interface. When the logical-item number is zero (the physical interface), it does not appear in the interface name. The other names in the interface table have the following significance:

- The `lo0` (loopback) interface is the local loopback.
- The `rj0` (reject) and `bh0` (blackhole) interfaces are used in the pool-summary feature.
- The `wanabe` interface is an inactive RADIUS dial-out profile.
- The `local` interface is the local machine.
- The `mcast` interface is the multicast interface, which represents the multicast forwarder for the entire class D address space. For details, see Chapter 8, "IP Multicast Configuration."
- The `tunnel` interface is a single pseudo-interface that is used only when the system is terminating tunnels. (The number terminating the tunnel interface name is an internal number that can change from one software version to the next.)
- The `vr0_main` interface represents the router itself.
- The `sip0` interface is the soft IP interface. For details, see "Defining a soft interface for increased accessibility" on page 4-8.
- The numbered WAN (`wan`*N*) interfaces are WAN connections, which are entered in the interface table as they become active.

IP2000 performance statistics

The IP2000 controller collects statistics on the number of packets and octets transmitted and received on each LIM interface. These counters are represented in the output of the `netstat -i` and `ifstat` commands, and are accessible to an external management utility.

The per-interface statistics for connections terminated on the IP2000 are cleared on the LIM when they are displayed on the controller by using the `netstat -i` or `ifstat` command.

For example, in the following output, the `Ipkts`, `Ierr`, `Opkts`, and `Oerr` statistic counters display the sum of the packets in transit as seen by the IP2000 added to the current packets counted by the TAOS interface manager.

```
admin> netstat -i
Name     MTU   Net/Dest          Address          Ipkts   Ierr   Opkts Oerr
ie0      1500  10.1.26.0/24    10.1.26.1     605504    0         0    0
ie1      1500  15.1.1.0/24     15.1.1.1           0    0         0    0
lo0      1500  127.0.0.1/32    127.0.0.1          8    0         8    0
rj0      1500  127.0.0.2/32    127.0.0.2          0    0         0    0
bh0      1500  127.0.0.3/32    127.0.0.3          0    0         0    0
wanabe   1500  127.0.0.3/32    127.0.0.3          0    0         0    0
local   65535  127.0.0.1/32    127.0.0.1      58935    0     58935    0
mcast   65535  224.0.0.0/4     224.0.0.0          0    0         0    0
tunnel0  1500  10.1.26.0/24    10.1.26.1          0    0         0    0
vr0_main 1500  10.1.26.1/32    10.1.26.1          0    0         0    0
sip0     65535  -               -                 0    0         0    0
wan11    1524  200.200.200.254 2.2.2.1           7    0        10    0
```

Similarly, in the following output, the `in_oct`, `in_errs`, `out_octet`, and `out_err` statistic counters display the sum of the packets in transit as seen by the IP2000 added to the current packets counted by the TAOS interface manager.

```
admin> ifstat 1
in_oct 0 in_errs 0 out_octet 0 out_err 0
```

The MIB II interface stat counters also now display the correct values when viewed from an external management utility.

IP address syntax

The system uses dotted decimal format (not hexadecimal) for IP addresses. If no subnet mask is specified, the system uses a default mask based on the address class. For example, a class C address, such as 198.5.248.40, has 24 network bits, leaving 8 bits for the host portion of the address. If no subnet mask is specified for a class C address, the system uses the default mask of 24 bits.

A subnet address includes a prefix length, which specifies the number of network bits in the address. For example, the following address specifies a 29-bit subnet:

```
ip-address = 198.5.248.40/29
```

In this address, 29 bits of the address are used to specify the network. The three remaining bits are used to specify unique hosts on the subnet. With three bits used to specify hosts on a 29-bit subnet, eight different bit combinations are possible. Of those eight possible host addresses, two are reserved:

000 — Reserved for the network (base address)
001
010
100
110
101
011
111 — Reserved for the broadcast address of the subnet

Note Be careful with zero subnets (subnets with the same base address as a class A, B, or C network). Early implementations of TCP/IP did not allow them. For example, the subnet 192.32.8.0/30 was illegal because it had the same base address as the class C network 192.32.8.0/24, while the subnet 192.32.8.4/30 was legal. Modern implementations of TCP/IP support zero subnets, and the Stinger implementation of RIP treats these subnets the same as any other network. However, you must treat zero subnets consistently throughout your network. Otherwise, you will encounter routing problems.

Table 4-1 shows subnet masks and prefix lengths for a class C network number.

Table 4-1. Decimal subnet masks and corresponding prefix lengths

Subnet mask	Number of host addresses	Prefix length
255.255.255.0	254 hosts + 1 broadcast, 1 network base	/24
255.255.255.128	126 hosts + 1 broadcast, 1 network base	/25
255.255.255.192	62 hosts + 1 broadcast, 1 network base	/26
255.255.255.224	30 hosts + 1 broadcast, 1 network base	/27
255.255.255.240	14 hosts + 1 broadcast, 1 network base	/28
255.255.255.248	6 hosts + 1 broadcast, 1 network base	/29
255.255.255.252	2 hosts + 1 broadcast, 1 network base	/30
255.255.255.254	Invalid mask (no hosts)	/31
255.255.255.255	1 host—a host route	/32

The broadcast address of any subnet has the host portion of the IP address set to all ones. The network address (or base address) represents the network itself, because the host portion of the IP address is all zeros. For example, supposing the IP configuration assigns the following address to a remote router:

`198.5.248.120/29`

The Ethernet network attached to that router has the following address range:

`198.5.248.120 – 198.5.248.127`

A host route is a special-case IP address with a prefix length of /32. For example:

`198.5.248.40/32`

Host routes are to a single host, rather than to a router or subnet.

Configuring `ip-interface` profiles for Ethernet ports

The system creates an `ip-interface` profile for an Ethernet port when it first detects the presence of the port. For example, the following output shows the default `ip-interface` profiles for the soft interface (the profile with the zero index) and the IP2000 controller:

```
admin> dir ip-interface
    21  07/24/2004 13:55:31  { { any-shelf any-slot 0 } 0 }
    31  07/24/2004 22:46:34  { { shelf-1 first-control-module 1 } 0 }
    21  07/24/2004 13:57:01  { { shelf-1 first-control-module 2 } 0 }
    36  07/24/2004 17:34:13  { { shelf-1 first-control-module 2 } 1 }
```

```
21  07/24/2004 13:55:31  { { shelf-1 second-control-module 1 } 0 }
21  07/24/2004 13:57:01  { { shelf-1 second-control-module 2 } 0 }
```

The profile for the Gigabit Ethernet interface on the first IP2000 controller (in slot 8) uses the following index:

```
{ { shelf-1 first-control-module 2 } 0 }
```

This index consists of a physical address and a logical-item number in the following format:

{ { *shelf-num slot-num item-num* } *logical-item-num* }

The logical item addresses a specific logical interface. It is zero except when multiple (virtual) interfaces have been configured on the physical port. For more details, see "Defining a local virtual IP interface" on page 4-7.

Overview of typical local interface settings

For information about enabling IP multicast forwarding on the Gigabit Ethernet interface, see Chapter 8, "IP Multicast Configuration."

Following are the parameters, shown with default settings, used to configure the IP2000 Gigabit Ethernet port as an IP interface:

```
[in IP-INTERFACE/{ { shelf-1 first-control-module 2 } 0 }]
interface-address* = { { shelf-1 first-control-module 2 } 0 }
ip-address = 0.0.0.0/0
rip-mode = routing-off
rip2-use-multicast = yes
directed-broadcast-allowed = yes
vlan-enabled = no
vlan-id = 0
```

Parameter	Setting
interface-address	Address of the interface in the Stinger unit, or, if the item number is not zero, the virtual interface address.
ip-address	IP address of the LAN interface. If the LAN IP address includes a subnet specification, you must create a static route to another LAN router to enable the system to reach local networks beyond its own subnets. For details, see "Configuring ip-route profiles" on page 4-25.
rip-mode	Enables or disables RIP updates on the interface. You can enable RIP to receive routing table updates, send them, or both. Running RIP-2 and RIP-v1 on the same network in such a way that the routers receive each other's advertisements is *not recommended*.
rip2-use-multicast	Enables or disables use of the multicast address (224.0.0.9) rather than the broadcast address for RIP updates.
directed-broadcast-allowed	Enables or disables forwarding of directed broadcast traffic onto the interface and its network.

Parameter	Setting
vlan-enabled vlan-id	Configure a routed VLAN interface. See "Configuring routed VLANs" on page 3-20.

Configuring a local IP interface

The following command lists the ip-interface profiles created by the system for a Stinger unit with redundant IP2000 controllers:

```
admin> dir ip-interface
    21  07/24/2004 13:55:31  { { any-shelf any-slot 0 } 0 }
    31  07/24/2004 22:46:34  { { shelf-1 first-control-module 1 } 0 }
    21  07/24/2004 13:57:01  { { shelf-1 first-control-module 2 } 0 }
    21  07/24/2004 13:55:31  { { shelf-1 second-control-module 1 } 0 }
    21  07/24/2004 13:57:01  { { shelf-1 second-control-module 2 } 0 }
```

The next command assigns an IP address to the Gigabit Ethernet port of the first controller (installed in slot 8):

```
admin> read ip-interface { { 1 8 2 } 0 }
admin> set ip-address = 1.1.1.2/29
admin> write -f
```

In this example configuration, the Gigabit Ethernet interface is connected to the 1.1.1 subnet. RIP is off by default, so to enable the interface to communicate with routers on other local subnets, either the system must have a static route configuration to another router in its own subnet, or the interface must enable RIP. For an example of configuring a static route, see "Configuring ip-route profiles" on page 4-25.

The following commands configure the interface to receive RIP-2 updates on the multicast address (the multicast address is the default):

```
admin> read ip-interface { { 1 8 2 } 0 }
admin> set rip-mode = routing-recv-v2
admin> write -f
```

You can verify that the system can transfer IP packets across the interface by pinging another host on the same network segment, as shown in the following example:

```
admin> ping 1.1.1.19
PING 1.1.1.19: 56 Data bytes
64 bytes from 1.1.1.19: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 1.1.1.19: icmp_seq=3 ttl=255 time=0 ms
^C
```

Defining a local virtual IP interface

You can configure up to 16 ip-interface profiles for each IP2000 module as a whole, with each profile specifying one IP address. For details about using a virtual IP interface for a management VLAN, see "Configuring routed VLANs" on page 3-20.

The system creates the default profile for an interface and assigns it the zero logical-item number. To configure another IP address on a LAN interface, create an ip-interface profile with a nonzero logical-item number in its interface address.

For example, the following commands create a virtual interface for the Gigabit Ethernet port:

```
admin> read ip-interface { { 1 8 2 } 1 }
admin> set ip-address = 1.1.1.1/29
admin> write -f
```

The logical-item numbers do not have to be consecutive, but they must each be unique.

Defining a soft interface for increased accessibility

You can configure a soft IP interface, which is an internal IP interface that is always active and reachable, as long as one of the system's IP interfaces is up. The `ip-interface` profile with the zero index is reserved for the soft interface.

Note Do not specify the IP address of a physical LAN interface as the soft interface address.

The following commands set the soft interface IP address to 1.1.1.128:

```
admin> read ip-interface { { 0 0 0 } 0 }
admin> set ip-address = 1.1.1.128
admin> write -f
```

If RIP is enabled, the system advertises the soft interface address as a host route (with a prefix length of 32 bits) using the loopback interface. If RIP is not enabled, routers one hop away from the unit must have a static route to the soft interface address.

To verify that other hosts in your network have a route to the soft address, run `ping` or `traceroute` from the other hosts. For example:

```
host1% ping 1.1.1.128
PING 1.1.1.128 (1.1.1.128): 56 Data bytes
64 bytes from 1.1.1.128: icmp_seq=0 ttl=255 time=0 ms
64 bytes from 1.1.1.128: icmp_seq=7 ttl=255 time=0 ms
^C
```

Disabling directed broadcasts to protect against denial-of-service

Denial-of-service attacks known as *smurf* attacks typically use ICMP Echo Request packets with a spoofed source address and packets directed to IP broadcast addresses. These attacks are intended to degrade network performance, possibly to the point that the network becomes unusable.

To prevent the IP router from being used as an intermediary in this type of denial-of-service attack launched from another network, you must disable the router from forwarding directed broadcasts it receives from another network. You must explicitly disable directed broadcasts on *all* IP interfaces in the system (including the management interface). In a system with redundant controllers, disable the feature on both controllers, so the unit is still protected following a switchover. The following commands configure the Gigabit Ethernet interface on the first controller:

```
admin> read ip-interface { { 1 8 2 } 0 }
admin> set directed-broadcast-allowed = no
admin> write -f
```

Configuring ip-global network features

The IP router has many configuration settings that affect its operations. The settings that determine its routing policies include security, RIP options, IP route cache options, and other options. These settings are available only in the ip-global profile. They have no counterpart in RADIUS.

Of the many settings in the ip-global profile, some are intended for remote access service and are not directly relevant to IP2000 operations. However, there are many more options you can choose to configure in this profile. For details about all ip-globalparameters and subprofiles, see the *Stinger Reference*.

Setting a system address

The system IP address is the source address used for all packets generated by the system. It must be the real address of one of the unit's LAN IP interfaces, or the soft interface address (see "Defining a soft interface for increased accessibility" on page 4-8.) Following is the parameter for specifying a system address:

```
[in IP-GLOBAL]
system-ip-addr = 0.0.0.0
```

With the default zero address, the Stinger unit uses the IP address assigned to the management IP address (configured in the ip-interface profile with the {1 f 1 } index), or if that address is not specified, the Gigabit Ethernet interface({ 1 f 2 }) as the source address for packets it generates. However, explicitly setting the system address simplifies access control. For example, most RADIUS servers keep a database of known remote access server (RAS) clients and their authentication keys. If you do not specify a system address, the RADIUS database must include a complete list of all the system's interface addresses. If you specify a system address, it is used for all RADIUS request packets.

Following is an example of setting the system-ip-addr parameter to the Ethernet interface address:

```
admin> get ip-interface { { 1 8 2 } 0 } ip-address
ip-address = 2.2.2.2
admin> read ip-global
admin> set system-ip-addr = 2.2.2.2
admin> write -f
```

Configuring DNS

Domain Name System (DNS) is a TCP/IP service for centralized management of address resolution. You enable DNS lookups by specifying a domain name and the IP addresses of one or more local servers.

Some sites maintain multiple DNS servers, each one dedicated to a particular client or location. In addition, some servers support a list feature that enables them to return multiple addresses for a hostname in response to a DNS query. For information about those DNS features, see the *Stinger Reference*.

Overview of typical DNS settings

Following are the parameters (shown with default settings) for configuring DNS to allow lookups:

```
[in IP-GLOBAL]
domain-name = ""
dns-primary-server = 0.0.0.0
dns-secondary-server = 0.0.0.0
sec-domain-name = ""
```

Parameter	Setting
domain-name	Primary domain name to use for DNS lookups. The system appends this domain name to hostnames when performing lookups.
dns-primary-server	Address of the primary local DNS server to use for lookups.
dns-secondary-server	Address of the secondary local DNS server to use for lookups. Used only if the primary server is not found.
sec-domain-name	Secondary domain name to use for DNS lookups if the hostname is not found in the primary domain.

Specifying domain names for lookups

The following commands specify a primary and secondary domain name for DNS lookups:

```
admin> read ip-global
admin> set domain-name = abc.com
admin> set sec-domain-name = eng.abc.com
admin> write -f
```

If a lookup fails with the first domain name, the router tries again with the secondary domain name.

Setting RIP options

The following parameters (shown with default settings) define how the system handles RIP updates:

```
[in IP-GLOBAL]
rip-policy = Poison-Rvrs
summarize-rip-routes = no
rip-trigger = yes
rip-pref = 100
dialout-poison = no
rip-queue-depth = 0
ignore-def-route = yes
suppress-host-routes = no
```

Parameter	Setting
`rip-policy`	Policy for sending update packets that include routes received on the same interface.
`summarize-rip-routes`	Enables or disables summarization of subnet information in RIP-v1 updates. This setting has no effect on RIP-2 updates.
`rip-trigger`	Enables or disables RIP triggering. With a `yes` setting (the default), RIP updates include only changed routes.
`rip-pref`	Default preference for routes learned from RIP updates. When choosing the routes to put in the routing table, the unit first compares their preference values, preferring the lowest number. If the preference values are equal, the router compares the metric values, using the route with the lowest metric. Specify a number from `0` through `255`. A value of `255` prevents the use of the route. The default preferences for different types of routes are `0` (zero) for connected routes, `10` for OSPF routes, `30` for routes learned from ICMP redirects, and `100` for routes learned from RIP and static routes.
`dialout-poison`	Enables or disables advertisement of dial-out routes when no trunks are available. Stinger units do not dial out, so leave this parameter at its default setting.
`ignore-def-route`	Enables or disables exclusion of advertised default routes from the routing table.
`rip-queue-depth`	Maximum number of RIP packets to be held for processing. Valid values are 0 to 1024. The default (0) means that the router will not drop any RIP packets, no matter how far behind it gets.
`suppress-host-routes`	Enables or disables suppression of host routes for interfaces with a subnet mask of less than 32 bits.

RIP policy for propagating updates back to the originating subnet

You can specify a split-horizon or poison-reverse policy for outgoing update packets that include routes received on the same interface on which the update is sent. Split-horizon means that the router does not propagate routes back to the subnet from which they were received. Poison-reverse means that it propagates routes back to the subnet from which they were received, but with a metric of 16 (infinite metric).

The following set of commands specifies the split-horizon policy:

```
admin> read ip-global
admin> set rip-policy = split
admin> write -f
```

RIP triggering

RIP triggering enables the router to tag routes that have been updated in the routing table and send updates that include only the changed routes. The result is reduced processing overhead for both the TAOS router and its neighbors.

With the default value (`yes`), the router tags changes to its routing table and includes only the tagged routes in its next update. Changes occur when a call arrives or disconnects, RIP learns a route from another router, or the administrator modifies a route-related profile. The router broadcasts updates 5 to 8 seconds after the first change in the routing table is detected. The delay helps to prevent constant updates during peak traffic conditions.

If `rip-trigger` is set to `no`, the router sends full table updates every 20 to 40 seconds. To prevent RIP routers on a network from synchronizing and sending large updates in unison, the full table update is no longer broadcast at fixed 30-second intervals.

Limiting the size of UDP packet queues

When the router is very busy and receives a flood of UDP packets from SNMP requests or RIP updates, a backlog of packets waiting for processing can create enough delay in routing to cause sporadic problems with time-sensitive packets, such as LCP negotiation or frame relay management packets.

To prevent such problems, UDP processing runs at a lower priority than processing of routed packets. On a system busily routing packets, UDP processing might be delayed, and a backlog of UDP packets builds up. The `rip-queue-depth` parameter in the `ip-global` profile and the `queue-depth` parameter in the `snmp` profile specify the maximum size of this backlog.

When you set one of these parameters to specify a queue depth, the system is more likely to drop UDP packets when it is busy routing packets. However, time-sensitive routed packets are less likely to be delayed and system memory is used more efficiently.

In following sample commands sets both queue depths to 50. Fifty of each type of packet is held for processing, and if additional packets of either type are received when the queue is full, they are dropped.

```
admin> read ip-global

admin> set rip-queue-depth = 50

admin> write -f

admin> read snmp

admin> set queue-depth = 50

admin> write -f
```

The `netstat` command output shows the queue depth of various UDP ports, and the total packets received and total packets dropped on each port. The total packets received count includes dropped packets. In the following example, the SNMP queue depth was set to 32:

```
admin> netstat udp
udp:
Socket  Local Port  InQLen  InQMax    InQDrops    Total Rx
  0        1023       0       1           0           0
  1       route       0      50           0         509
  2        echo       0      32           0           0
  3         ntp       0      32           0           0
  4        1022       0     128           0           0
  5        SNMP      32      32        5837       20849
```

Ignoring default routes when updating the routing table

Lucent Technologies recommends enabling the `ignore-def-route` parameter to prevent routing updates from modifying the default route in the routing table. The following set of commands protects the default route from RIP updates:

```
admin> read ip-global
admin> set ignore-def-route = yes
admin> write -f
```

Suppressing host-route advertisements

If you set the `suppress-host-routes` parameter to `yes`, routes are suppressed according to the following rules:

- If a `connection` profile includes a subnet mask of less than 32 bits in the `remote-address` setting, host routes for the interface are suppressed while the session is being negotiated, and after the session is established, only network routes are advertised for the interface.
- If a `connection` profile includes a subnet mask of /32 in the `remote-address` setting, host routes for the interface are not suppressed. (Pool addresses also have a 32-bit mask, so they are not suppressed.)

The following set of commands configures the router to suppress host routes for connections that specify a subnet mask of less than 32 bits:

```
admin> read ip-global
admin> set suppress-host-routes = yes
admin> write -f
```

Configuring and using address pools

An address pool is a range of contiguous addresses on a local IP network or subnet. Pool addresses are available for assignment to incoming connections that request an address. When the call terminates, the address is returned to the pool, making it available again for assignment.

If you designate a subnet for IP address pools, you must make sure that other IP hosts on the local network know the route to that subnet. You must also make sure that the pools do not overlap (do not contain duplicate addresses).

For related information, see “Defining address pools for a virtual router” on page 6-7.

Overview of settings for defining pools

You can define up to 128 address pools locally in the `ip-global` profile. Those pools can be used to assign addresses to connections authenticated locally (in `connection` profiles) or by RADIUS. If you are using RADIUS authentication, you can choose to define address pools in RADIUS instead of, or in addition to, those defined locally. If you have the RADIPAD program installed, you can use it to manage address pools centrally on a single RADIUS server.

Settings in the **ip-global** *profile*

The following parameters (shown with default values) configure address pools locally:

```
[in IP-GLOBAL]
pool-summary = no
pool-base-address = [ 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0+
assign-count = [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 +
pool-name = [ "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "+
must-accept-address-assign = no
```

Parameter	Setting
pool-summary	Set/clear the Pool Summary flag. For details, see "Example of configuring summarized address pools" on page 4-17.
pool-base-address	Base address of a pool of contiguous addresses on a local network or subnet.
assign-count	Number of addresses in the pool.
pool-name	A pool name, required only when TACACS+ authentication is in use. If TACACS+ authentication is not in use, the name is treated as a comment.
must-accept-address-assign	Enables or disables rejection of an assigned IP address by an incoming caller during PPP negotiation.

Settings in RADIUS pseudo-user profiles

You can define address pools in a RADIUS pools pseudo-user profile. The first line of pools pseudo-user profile uses the following format:

```
pools-name Password = "ascend", Service-Type = Outbound-User
```

The *name* argument is the system name (specified by the name parameter in the system profile). Subsequent lines in the profile define IP address pools by using the Ascend-IP-Pool-Definition (217) attribute. The value of the Ascend-IP-Pool-Definition attribute uses the following syntax:

"*pool-num base-addr assign-count*"

Syntax element	Description
pool-num	Pool number. If you use the same number to designate two pools, one locally and one in RADIUS, the RADIUS definition takes precedence. So if you have defined some pools in the ip-global profile and do not wish to override them, start numbering the pools at the next number. For example, if you defined 10 pools in the ip-global profile, start with number 11 in RADIUS. Otherwise, start with 1.
base-addr	The base address in a pool of contiguous addresses on the local network or subnet.
assign-count	Number of addresses included in the pool.

Global RADIUS pools (RADIPAD)

RADIUS IP Address Daemon (RADIPAD) is a program that works with RADIUS authentication to manage IP address pools centrally, so that connections can all acquire an address from a global pool, regardless of which system answers the call.

RADIPAD runs on one RADIUS server on the network. A Stinger unit sends an authentication request to RADIUS, and if the user profile contains an attribute to allocate an IP address from the global pool, RADIUS sends a request to RADIPAD to acquire the address.

The Stinger unit does not communicate directly with RADIPAD, so it does not require additional configuration to use RADIPAD. To configure RADIPAD, you define the global pools of addresses, specify which RADIUS server is running RADIPAD, and (optionally) specify which systems can obtain addresses from those pools. You can then create RADIUS user profiles that acquire an IP address from the global pool.

At startup, `syslog` notes RADIUS requests to release RADIUS-allocated IP addresses. Some versions of the RADIUS server might time out the request, resulting in log messages indicating the release of global-pool addresses.

Defining global pools

Global address pools are defined in a `global-pools` pseudo-user profile on the server running RADIPAD. The first line of a `global-pools` pseudo-user profile uses the following format:

`global-pools-`*`name`* `Password = "ascend", Service-Type = Outbound-User`

The *name* argument is a designation for any class of users. You can create multiple global pool profiles for multiple user classes. For example, you could create profiles named `global-pool-ppp`, `global-pool-slip`, and so forth. Subsequent lines in the profile define IP address pools by using the Ascend-IP-Pool-Definition (217) attribute. This is the same attribute described in "Settings in RADIUS pseudo-user profiles" on page 4-14, and it follows the same rules for global pools. In addition, when the Stinger unit assigns an address from a pool managed by the RADIPAD daemon, RADIPAD tries to allocate an address from the pools in order, by pool number, and chooses an address from the first pool with an available IP address.

Specifying the RADIPAD host

Each RADIUS server must specify the host running RADIPAD and (optionally) the systems that can access the global pools. These settings are defined in a `radipa-hosts` pseudo-user profile, which uses the following format in the first line of the profile:

`radipa-hosts Password = "ascend", Service-Type = Outbound-User`

Subsequent lines in the profile use the following attribute-value pairs to define which hosts can assign addresses from the pools managed by RADIPAD:

RADIUS attribute	Value
Ascend-Assign-IP-Client (144)	Address of a system that is allowed to access the global address pools managed by RADIPAD. You can specify multiple instances of this attribute. If no client addresses are specified, all units listed in the RADIUS clients file can access RADIPAD pools.

RADIUS attribute	Value
Ascend-Assign-IP-Server (145)	Address of the server running RADIPAD. Only one instance of this attribute can appear in the profile, and it must specify the correct IP address.

For example:

```
radipa-hosts Password ="ascend", Service-Type = Outbound-User
   Ascend-Assign-IP-Server = 10.31.4.34,
   Ascend-Assign-IP-Client = 10.31.4.10,
   Ascend-Assign-IP-Client = 10.31.4.11
```

You can specify only one RADIPAD server, but you can include multiple clients. The sample profile indicates that two systems (10.31.4.10 and 10.31.4.11) can access RADIPAD pools as clients.

Preventing the use of class boundary addresses

If you define address pools that contain more than 254 addresses, be aware that the system allocates the class boundary addresses (*n.n.n*.0 and *n.n.n*.255) as valid connection addresses. According to CIDR, this is permitted because the pool is not a /24 network. However, some client systems, such as Windows, do not tolerate the class boundary addresses well. For example, because Windows assumes a /24 network, it broadcasts NetBIOS over IP name service to the .255 address, which could overwhelm a connection assigned the .255 host address.

To prevent client software from using a host address for broadcasts, you must explicitly apply a filter that prevents the system from using the class boundary addresses. For example, if you are using RADIUS authentication, you can apply a data filter, in the `answer-defaults` profile, that drops packets from any source to pool address *n.n.n*.0 or *n.n.n*.255.

Examples of configuring address pools

For a pool that is not summarized, each assigned address is advertised as its own host route. Such a pool can start at any base address. Addresses do not accept a subnet mask component, because they are always advertised as host routes.

The following commands define three address pools, each containing 50 addresses. Pool 1 contains 10.2.3.4 through 10.2.3.54. Pool 2 contains 11.5.7.51 through 11.5.7.101. Pool 3 contains 12.7.112.15 through 12.7.112.65.

```
admin> read ip-global
admin> set pool-base-address 1 = 10.2.3.4
admin> set pool-base-address 2 = 11.5.7.51
admin> set pool-base-address 3 = 12.7.112.15
admin> set assign-count 1 = 50
admin> set assign-count 2 = 50
admin> set assign-count 3 = 50
admin> write -f
```

Following is a comparable RADIUS pools profile (for use by a single RADIUS server):

```
pools-taos01 Password = "ascend", Service-Type = Outbound-User
   Ascend-IP-Pool-Definition = "1 10.2.3.4 50",
   Ascend-IP-Pool-Definition = "2 11.5.7.51 50",
   Ascend-IP-Pool-Definition = "3 12.7.112.15 50"
```

Following is a comparable global pools definition (for use with RADIPAD):

```
global-pool-ppp Password ="ascend", Service-Type = Outbound-User
   Ascend-IP-Pool-Definition = "1 10.2.3.4 50",
   Ascend-IP-Pool-Definition = "2 11.5.7.51 50",
   Ascend-IP-Pool-Definition = "3 12.7.112.15 50"
```

Although some client software assumes a default subnet mask of 255.255.255.0 for PPP interfaces, you can define pools on subnets wider than /24. For example, the following commands define an address pool on a /23 subnet:

```
admin> read ip-global
admin> set pool-base-address 1 = 10.55.178.1
admin> set assign-count 1 = 510
admin> write -f
```

This pool definition translates to 10.55.178.0/23 (a subnet mask of 255.255.254.0). Following are comparable RADIUS definitions:

```
pools-taos01 Password = "ascend", Service-Type = Outbound-User
   Ascend-IP-Pool-Definition = "1 10.55.178.1 510"
global-pool-ppp Password ="ascend", Service-Type = Outbound-User
   Ascend-IP-Pool-Definition = "1 10.55.178.1 510"
```

Example of configuring summarized address pools

The pool-summary feature reduces routing overhead associated with address pools. Instead of advertising each address assigned from a pool as a host route, the system suppresses the host route advertisements and instead advertises a static route to the pool itself.

To use summarized pools locally or in RADIUS, you must set the pool-summary flag to yes in the ip-global profile, and you must define all pools to be network-aligned.

Setting the pool-summary flag

The following commands enable the pool-summary flag:

```
admin> read ip-global
admin> set pool-summary = yes
admin> write -f
```

Defining network-aligned pools

Following are the rules for network-aligned address pools:

- The specified number of addresses in the pool must be two less than the total number of addresses in the pool. (Add 2 to the assign-count value for the total number of addresses in the subnet, and calculate the mask for the subnet on the basis of this total.)

`assign-count` + 2 = number of subnet hosts

- The specified base address of the pool must be the first host address. (Subtract 1 from the `pool-base-address` value for the base address for the subnet.)

  `pool-base-address` `- 1 = network-aligned subnet address`

The following commands enable the `pool-summary` flag and define a network-aligned pool:

```
admin> read ip-global
admin> set pool-summary = yes
admin> set assign-count 1 = 62
admin> set pool-base-address 1 = 10.12.253.1
admin> write -f
```

In the preceding sample configurations, the `assign-count` parameter is set to 62. When you add 2 to this value, you get 64. The subnet mask for 64 addresses is 255.255.255.192 (256 – 64 = 192). The prefix length for a 255.255.255.192 mask is /26.

The `pool-base-address` parameter is set to 10.12.253.1. When you subtract 1 from this value, you get 10.12.253.0, which is a valid network-aligned base address for the 255.255.255.192 subnet mask. (Note that 10.12.253.64, 10.12.253.128, and 10.12.253.192 are also valid zero addresses for the same mask.) The resulting address pool subnet is 10.12.253.0/26.

Following is a comparable RADIUS `pools` profile (for use by a single RADIUS server).

```
pools-taos01 Password = "ascend", Service-Type = Outbound-User
   Ascend-IP-Pool-Definition = "1 10.12.253.1 62"
```

Following is a comparable global pools definition (for use with RADIPAD):

```
global-pool-ppp Password ="ascend", Service-Type = Outbound-User
   Ascend-IP-Pool-Definition = "1 10.12.253.1 62"
```

The system still creates (but does not advertise) a host route for each assigned address in the pool. Host routes take precedence over subnet routes, so packets whose destination matches an assigned IP address from the pool are routed properly. However, because the system advertises the entire pool as a route, and only privately knows which IP addresses in the pool are active, a remote network might improperly send the Stinger unit a packet for an inactive IP address. If that occurs, the packets are routed to the Reject (`rj0`) interface (127.0.0.2). Packets routed to the Reject interface are bounced back to the sender with an `ICMP unreachable` message.

Examples of assigning an address from a pool

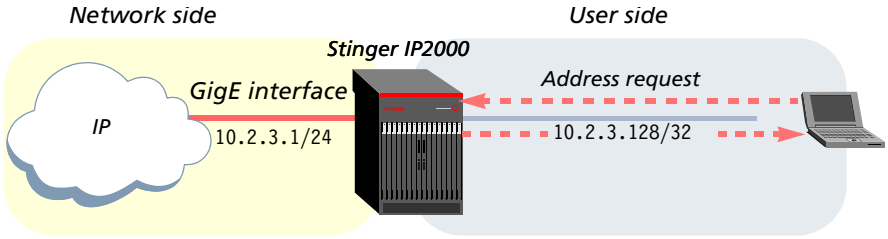
When an incoming call requests an IP address, the Stinger unit assigns one from a pool. A host requests an address if its client software has settings such as the following:

```
Username=victor
Accept Assigned IP=Yes
IP address=Dynamic (or Assigned or N/A)
Netmask=255.255.255.255 (or None or N/A)
Default Gateway=None or N/A
Name Server=10.2.3.55
```

```
Domain suffix=abc.com
Baud rate=38400
Hardware handshaking ON
VAN Jacobson compression ON
```

Figure 4-1 shows a remote host requesting and being assigned an IP address.

Figure 4-1. Remote CPE requiring dynamic IP address assignment



The following commands enable dynamic address assignment systemwide:

```
admin> read answer-defaults
admin> set ip-answer assign-address = yes
admin> write -f
```

During PPP negotiation, a CPE can reject an IP address offered by the router and present the caller's own IP address for consideration. For security purposes, many sites set `must-accept-address-assign` to `yes` to ensure that the Stinger unit terminates such a call, as shown in the following example:

```
admin> read ip-global
admin> set must-accept-address-assign = yes
admin> write -f
```

For address assignment to occur, the Stinger unit must have an address available for assignment, the `answer-defaults` profile must enable dynamic assignment, the client profile must specify dynamic assignment, and the client's PPP software must be configured to acquire its IP address dynamically.

The following commands configure a profile to acquire an address from the first pool that has available addresses:

```
admin> new connection victor
admin> set active = yes
admin> set encapsulation-protocol = ppp
admin> set ppp recv-password = localpw
admin> set ip-options address-pool = 0
admin> write -f
```

Following is a comparable RADIUS profile:

```
victor Password = "localpw"
   Service-Type = Framed-User,
   Framed-Protocol = PPP,
   Ascend-Assign-IP-Pool = 0
```

Following is a comparable RADIUS profile that acquires an address from any global pool managed by the RADIPAD daemon:

```
victor Password = "localpw"
   Service-Type = Framed-User,
   Framed-Protocol = PPP,
   Ascend-Assign-IP-Pool = 65535,
   Ascend-Assign-ip-global-Pool = "global-pool-ppp"
```

IP pool chaining

Because the addresses within a pool must be contiguous, many sites have defined a large number of pools, with each pool containing only a small range of addresses. For example, the following RADIUS profile defines six pools, each containing 10 addresses:

```
pools-JFAN-TNT Password = "ascend"
    Service-Type = Outbound,
    Ascend-IP-Pool-Definition = "1 11.168.6.10 10",
    Ascend-IP-Pool-Definition = "2 12.168.6.10 10",
    Ascend-IP-Pool-Definition = "3 13.168.6.10 10",
    Ascend-IP-Pool-Definition = "7 17.168.6.10 10",
    Ascend-IP-Pool-Definition = "8 18.168.6.10 10",
    Ascend-IP-Pool-Definition = "9 19.168.6.10 10"
```

In earlier versions of the software, you could allow a connection to acquire an address from any pool (by assigning the pool number 0 in the connection's profile) or from a single specified pool, such as pool 1. IP pool chaining enables you to allow a connection to acquire an address from any pool within a chain.

When IP pool chaining is enabled, contiguous pools are treated as one *pool space* with shared addresses. When the system assigns an address to an end user, it begins searching for an available address in the first pool of the chain and stops when it either finds an available address or encounters a null pool definition. So, the pools within a chain must be defined in a contiguous sequence. For example, the following profile contains two IP pool chains (pools 1, 2, 3 and pools 7, 8, 9), with each pool chain containing 30 addresses:

```
pools-JFAN-TNT Password = "ascend", Service-Type = Outbound
    Ascend-IP-Pool-Chaining = IP-Pool-Chaining-Yes,
    Ascend-IP-Pool-Definition = "1 11.168.6.10 10",
    Ascend-IP-Pool-Definition = "2 12.168.6.10 10",
    Ascend-IP-Pool-Definition = "3 13.168.6.10 10",
    Ascend-IP-Pool-Definition = "7 17.168.6.10 10",
    Ascend-IP-Pool-Definition = "8 18.168.6.10 10",
    Ascend-IP-Pool-Definition = "9 19.168.6.10 10"
```

Note To support IP pool chaining in RADIUS profiles, the RADIUS server must support vendor-specific attributes (VSA) and the system must be configured in VSA compatibility mode. For details, see "Pool chaining in RADIUS" on page 4-23.

IP pool chaining is supported both for RADIUS-defined address pools and for pools defined locally in the `ip-global` profile. For example, the following settings in the `ip-global` profile enable pool chaining and define a pool chain (pools 1 and 2) that contains 252 addresses:

```
[in IP-GLOBAL]
pool-chaining = yes
pool-base-address = [ 172.20.31.1 172.20.33.1 0.0.0.0 153.37.21.1 0.0+
assign-count = [ 126 126 0 30 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0+
```

Pool chaining in local profiles

Whether pool chains are defined locally or in RADIUS, the pool addresses are available for dynamic assignment regardless of where the connection's profile is authenticated.

Overview of local profile settings

Following are the parameters, shown with default settings, relevant to IP pool chaining:

```
[in IP-GLOBAL]
pool-chaining = no
pool-base-address = [ 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0+
assign-count = [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0+
[in CONNECTION/"":ip-options]
address-pool = 0
```

Parameter	Setting
`pool-chaining`	Enables or disables IP pool chaining. With the `yes` setting, the system treats contiguous IP address pools as a single extended pool space when searching for an available address to assign to a connection.
`pool-base-address`	An array of up to 128 IP addresses to be used as the first address in a pool. These values are used with the `assign-count` values to define address pools locally. A pool chain contains all of the pools defined in sequence within the array, such as 1, 2, 3. To end a pool chain, leave a null value in the array.
`assign-count`	An array of up to 128 numbers that specify the number of addresses in a pool that starts with the corresponding `pool-base-address`.
`address-pool`	Number of an address pool from which to acquire an address. When pool chaining is enabled, a pool number within a chain includes addresses defined in all other pools within the chain. For example, if pools 1, 2, and 3 are in a pool chain, setting this parameter to 1 has the same effect as setting it to 2 or 3.

Example of local pool chain definition

The following commands define five address pools, which form two pool chains. Notice that the pool numbers (their indexes in the `pool-base-address` and `assign-count` arrays) are contiguous within a chain.

```
admin> read ip-global
admin> set pool-chaining = yes
admin> set pool-base-address 1 = 10.1.1.1
admin> set pool-base-address 2 = 11.1.1.1
admin> set pool-base-address 3 = 12.1.1.1
admin> set assign-count 1 = 50
admin> set assign-count 2 = 50
admin> set assign-count 3 = 50
admin> set pool-base-address 7 = 13.1.1.1
admin> set pool-base-address 8 = 14.1.1.1
admin> set assign-count 7 = 50
admin> set assign-count 8 = 50
admin> write -f
```

The following commands enable dynamic address assignment systemwide:

```
admin> read answer-defaults
admin> set ip-answer assign = yes
admin> write -f
```

The following commands configure profiles to acquire an address from the first pool chain. When the end users initiate a session request, they can acquire an address from 10.1.1.1 to 10.1.1.51, from 11.1.1.1 to 11.1.1.51, or from 12.1.1.1 to 12.1.1.51. If no addresses are available within those ranges, the connection is refused.

```
admin> new connection jfan
admin> set active = yes
admin> set encapsulation-protocol = ppp
admin> set ppp-options recv-password = localpw
admin> set ip-options address-pool = 2
admin> write -f
admin> new connection ravi
admin> set active = yes
admin> set encapsulation-protocol = ppp
admin> set ppp-options recv-password = localpw
admin> set ip-options address-pool = 1
admin> write -f
```

Following are comparable RADIUS profiles:

```
jfan Password = "localpw"
    Service-Type = Framed-User,
    Framed-Protocol = PPP,
    Ascend-Assign-IP-Pool = 2
ravi Password = "localpw"
    Service-Type = Framed-User,
```

```
Framed-Protocol = PPP,
Ascend-Assign-IP-Pool = 1
```

Pool chaining in RADIUS

Whether pool chains are defined locally or in a RADIUS pool's pseudo-user profile, the pool addresses are available for dynamic assignment regardless of where the connection's profile is authenticated.

Overview of RADIUS profile settings

RADIUS servers use the following attribute-value pairs to define and apply pool chains:

RADIUS attribute	Value
Ascend-IP-Pool-Chaining (85)	Enables or disables IP pool chaining in a pseudo-user profile that defines address pools. If this attribute is set to IP-Pool-Chaining-Yes (1), the system treats contiguous IP address pools as a single extended pool space when searching for an available address to assign to a connection. With a value of IP-Pool-Chaining-No (0), the system treats each address pool as a separate space. **Note** When this attribute is specified in a RADIUS profile, its value overrides the Pool-Chaining setting in the `ip-global` profile.
Ascend-IP-Pool-Definition (217)	Address pool definition in a pseudo-user profile. The value has the following syntax: *`pool-number base-addr assign-count`* The *`pool-number`* value is an integer that identifies the pool. A pool chain contains all of the pools defined in sequence, such as 1, 2, 3. To end a pool chain, leave a gap in the sequence of *`pool-number`* values. The *`base-addr`* value is an IP address to be used as the first address in a pool, and the *`assign-count`* value specifies the number of addresses in a pool.
Ascend-Assign-IP-Pool (218)	Number of the address pool from which the RADIUS user profile should acquire an address. When pool chaining is enabled, a pool number within a chain includes addresses defined in all other pools within the chain. For example, if pools 1, 2, and 3 are in a pool chain, setting this value to 1 has the same effect as setting it to 2 or 3.

To use these attributes, the RADIUS server must support vendor-specific attributes (VSAs) and the system must be configured in VSA compatibility mode. Following are the relevant settings:

```
[in EXTERNAL-AUTH]
auth-type = radius

[in EXTERNAL-AUTH:rad-auth-client]
auth-radius-compat = vendor-specific
```

For details about these settings, see the *Stinger Reference*.

Example of pool chaining in RADIUS

The following pseudo-user profile defines five address pools, which form two pool chains. Notice that the pool numbers are contiguous within a chain.

```
pools-JFAN-TNT Password = "ascend"
    Service-Type = Outbound,
    Ascend-IP-Pool-Chaining = IP-Pool-Chaining-Yes,
    Ascend-IP-Pool-Definition = "1 10.1.1.1 50",
    Ascend-IP-Pool-Definition = "2 11.1.1.1 50",
    Ascend-IP-Pool-Definition = "3 12.1.1.1 50",
    Ascend-IP-Pool-Definition = "7 13.1.1.1 50",
    Ascend-IP-Pool-Definition = "8 14.1.1.1 50"
```

The following commands configure local `connection` profiles to acquire an address from the first pool chain. When the end users initiate a session request, they can acquire an address from 13.1.1.1 to 13.1.1.51, or from 14.1.1.1 to 14.1.1.51. If no addresses are available within those ranges, the connection is refused.

```
admin> new connection hanif

admin> set active = yes

admin> set encapsulation-protocol = ppp

admin> set ppp-options recv-password = localpw

admin> set ip-options address-pool = 7

admin> write -f

admin> new connection hasnain

admin> set active = yes

admin> set encapsulation-protocol = ppp

admin> set ppp-options recv-password = localpw

admin> set ip-options address-pool = 8

admin> write -f
```

Following are comparable RADIUS user profiles:

```
hanif Password = "localpw"
    Service-Type = Framed-User,
    Framed-Protocol = PPP,
    Ascend-Assign-IP-Pool = 7

hasnain Password = "localpw"
    Service-Type = Framed-User,
    Framed-Protocol = PPP,
    Ascend-Assign-IP-Pool = 8
```

Configuring ip-route profiles

Any profile that specifies how to reach an IP device or subnet (such as an ip-interface, connection, or RADIUS user profile) specifies a static IP route to that destination. However, you can also configure static routes explicitly, to extend or fine-tune the routing table.

Overview of typical static route settings

You can define static routes in ip-route profiles or in RADIUS. For details about RADIUS pseudo-user and user profile route definitions, using the Framed-Route (22) or Ascend-Private-Route (104) attribute-value pair, see the *TAOS RADIUS Guide and Reference*.

Following are the local parameters (shown with default settings) for configuring a static route:

```
[in IP-ROUTE/""]
name* = ""
dest-address = 0.0.0.0/0
gateway-address = 0.0.0.0
metric = 8
private-route = no
active-route = yes
```

Parameter	Settings
name	Name of the profile (up to 31 characters).
dest-address	Destination IP address. The default value is 0.0.0.0, which represents the default route.
gateway-address	IP address of a next-hop router used to reach the specified destination. A next-hop router is directly connected to the same Ethernet segment, or is one hop away on a WAN link. When the Gigabit Ethernet interface is connected to a subnet and RIP is not enabled on the interface, the system must be informed about the gateway-address of other backbone routers that can route beyond the subnet.
metric	RIP metric (0–15) for the route. Among routes with the same destination address, the higher the metric, the less likely that the system will choose the route.
private-route	Enables or disables including the route in RIP updates.
active-route	Enables or disables entering the route in the routing table. (Setting the parameter to no is a useful way to make a route temporarily inactive, so you can reinstate the route later.)

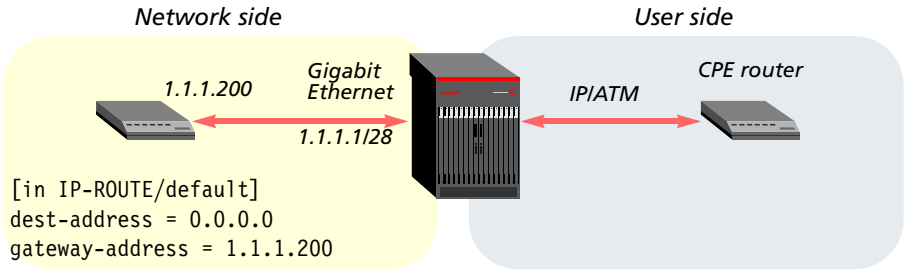
Offloading routing overhead to an external router

To offload routing overhead from the Stinger unit, you can define a default route to a router on the IP2000 Gigabit Ethernet interface. A default route is a special-case static route that acts as a catch-all for packets for which the Stinger unit cannot find a route. A default route has the zero address as its destination and points to a specific gateway address. The system routes all packets with unknown destinations to the specified gateway. If no default route is defined, the system drops those packets.

The system creates an `ip-route` profile named `default`, but the profile is not valid until you specify a gateway address, so the route is not active until you assign an address and activate the route. You can create a default route by modifying the `default` profile, or by creating one or more `ip-route` profiles that specify a zero destination and a valid gateway address.

Figure 4-2 shows a router that resides on the same subnet as the IP2000 Gigabit Ethernet IP interface. In this example, the system offloads part of its routing overhead by using a default route to the LAN router.

Figure 4-2. Default route to a local IP router



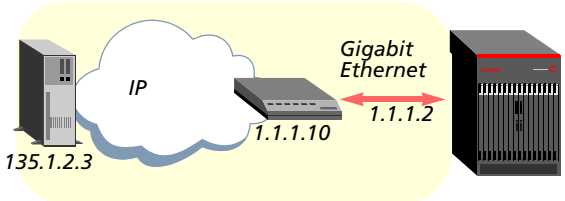
The following commands define a default route to the local router:

```
admin> read ip-route default
admin> set gateway-address = 1.1.1.200
admin> write -f
```

Creating a static route to a subnet

When RIP is turned off on an IP interface, the router cannot reach subnets beyond other routers on that interface unless it has a static route to the subnet. To enable access to subnets beyond the local segment, you must configure a static route. Figure 4-3 shows an example.

Figure 4-3. Static route to a subnet



The following commands configure a static route to the remote subnet:

```
admin> new ip-route subnet
admin> set dest-address = 135.1.2.3
admin> set gateway-address = 1.1.1.10
admin> write -f
```

Overview of routed subscriber connection features

The system creates a routing interface for local `connection` profiles when it starts up. For interfaces that use pool addresses or are defined in RADIUS user profiles, the system creates a routing interface when a session becomes active.

Subscriber connections can be PPP over ATM (PPPoA), PPP over Ethernet (PPPoE), IP over ATM (IPoA), or bridged IP routing (BIR) connections.

Source interface local addresses

You can define a Stinger local address to be used by multiple subscriber connections across DSL interfaces.

Note This feature is intended for use in `connection` or RADIUS profiles on Stinger DSL interfaces only.

The `source-if` setting provides several advantages over use of the `local-address` field as a means of assigning a connection-specific local address. For example, with `source-if` fewer IP addresses are consumed, because you can use the same local source address for many subscriber connections. In addition, the `source-if` address is used as the source address in most system-generated packets, which provides a security enhancement for sites that prefer not to reveal the configured `system-ip-addr` to packet analysis software.

Packets that use the specified source address

The system uses the assigned `source-if` address for all packets it generates except those in which upper protocol layers have already set the source IP address in a packet as, for example, in ICMP packets generated by the `ping -x` command. For those packets, the source address is set as it was in previous software versions to the value of the `system-ip-addr` parameter in the `ip-global` profile, or to the management interface IP address.

CPE client considerations

The subscriber connections can be routed IPoA, PPPoA, PPPoE, or BIR connections. For PPP connections, the system uses the assigned source address during IP NCP negotiation.

Multiple BIR connections can be configured with the same `local-address` setting. Support for that feature remains unchanged.

Soft IP interface requirement

With the `source-if` feature, for the system to bring up a connection to a CPE, it must have a configured soft IP interface for the `source-if` interface. You configure a soft IP interface by creating a new `ip-interface` with the default zero physical address and a

nonzero logical item number. For example, the next commands create the logical soft interface and set its address to 10.0.0.254:

```
admin> new ip-interface
admin> set interface-address logical-item = 1
admin> set ip-address = 10.0.0.254
admin> write -f
```

To assign this source address to a subscriber connection, you specify the name of the interface in the `source-if` parameter. The name uses the following format:

`sip`*N*

The *N* is the `logical-item` value of the profile. For example, the default soft interface is `sip0`, and the soft interface created in the preceding commands is `sip1`. You can display interface names by using the `netstat -i` command. For example:

```
admin> netstat -i
Name     MTU   Net/Dest           Address          Ipkts    Ierr Opkts  Oerr
ie0      1500 210.210.210.0/24   210.210.210.127   14044    0   5808    0
lo0      1500 127.0.0.1/32       127.0.0.1          5798    0   5798    0
rj0      1500 127.0.0.2/32       127.0.0.2             0    0      0    0
bh0      1500 127.0.0.3/32       127.0.0.3             0    0      0    0
wanabe   1500 127.0.0.3/32       127.0.0.3             0    0      0    0
local   65535 127.0.0.1/32       127.0.0.1          6079    0   6079    0
mcast   65535 224.0.0.0/4        224.0.0.0             0    0      0    0
tunnel0  1500 127.0.0.5/32       127.0.0.5             0    0      0    0
vr0_main 1500 127.0.0.4/32       127.0.0.4             0    0      0    0
sip0    65535 -                  -                     0    0      0    0
sip1    65535 10.0.0.254/32      10.0.0.254            0    0      0    0
```

Overview of configuration settings

Following are the relevant parameters, shown with default settings:

```
[in IP-INTERFACE/{ { any-shelf any-slot 0 } 0 }
ip-address = 0.0.0.0/0

[in IP-INTERFACE/{ { any-shelf any-slot 0 } 0 }:interface address
physical-address = { any-shelf any-slot 0 }
logical-item = 0

[in CONNECTION/"":ip-options]
source-if = ""
```

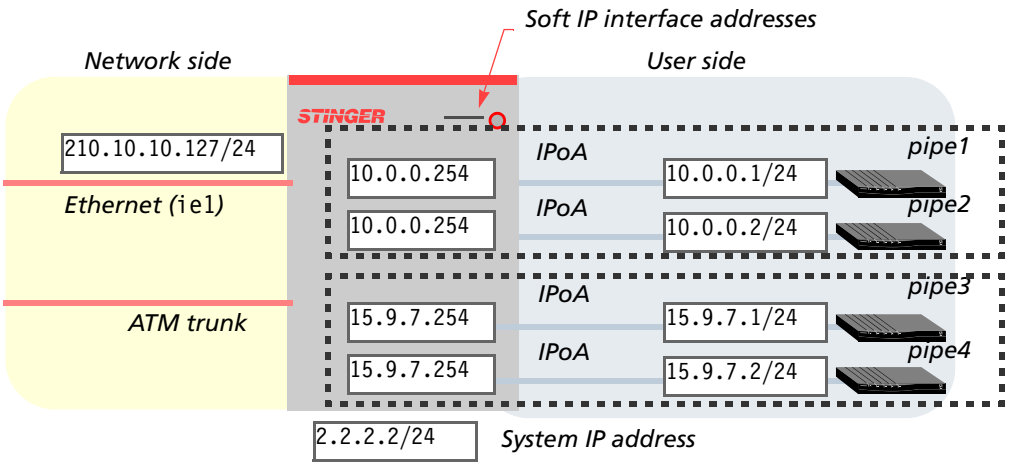
Parameter	Setting
`ip-address`	Address to be used as the source interface address for multiple subscriber connections. If you change this address while active calls refer to it through the `source-if` field, all of the calls are terminated and then reestablished.
`physical-address`	For a soft IP interface, this field must be set to its default value of `{ any-shelf any-slot 0 }`.
`logical-item`	For a soft IP interface to be used with `source-if`, this field must be set to a nonzero number.

Parameter	Setting
`source-if`	Name of the configured soft IP interface whose IP address will be the local address for this connection. The name uses the format `sip`*N*, where *N* is the `logical-item` value of the soft interface's `ip-interface` profile. You can display interface names by using the `netstat -i` command. If the corresponding soft IP interface does not exist, the call disconnects and the system logs a disconnect code. The `source-if` and `local-address` parameters cannot both be set in the same profile. If you change the setting of this parameter for a call that is already in the UP state, the call is terminated and then reestablished. For PPPoA/PPPoE calls, the client must reinitiate the PPP negotiation. The `Ascend-IP-Source-If` (309) attribute provides this functionality in a RADIUS user profile.

Sample configuration with a source interface address

Figure 4-4 shows a sample configuration with groups of subscriber connection sharing a `source-if` address. In this case, packets generated by the Stinger unit will not use the 2.2.2.2 `system-ip-addr` configured in the `ip-global` profile. A different source address will be used for each of the CPE groups.

Figure 4-4. Sample configuration using two soft IP interface addresses



The following commands configure the two soft IP interface profiles:

```
admin> new ip-interface { { 0 0 0 } 1 }
admin> set ip-address = 10.0.0.254/24
admin> write -f
admin> new ip-interface { { 0 0 0 } 2 }
admin> set ip-address = 15.9.7.254/24
admin> write -f
```

The next command displays interface names. Notice that the last two lines of the command output specify the `sip1` and `sip2` interfaces, which were created by the preceding set of commands.

```
admin> netstat -i
Name     MTU   Net/Dest            Address          Ipkts    Ierr Opkts  Oerr
ie0      1500 12.9.56.0/24     12.9.56.81       14044    0   5808    0
ie1      1500 210.210.210.0/24   210.210.210.127   14044    0   5808    0
lo0      1500 127.0.0.1/32       127.0.0.1          5798    0   5798    0
rj0      1500 127.0.0.2/32       127.0.0.2             0    0      0    0
bh0      1500 127.0.0.3/32       127.0.0.3             0    0      0    0
wanabe   1500 127.0.0.3/32       127.0.0.3             0    0      0    0
local   65535 127.0.0.1/32       127.0.0.1          6079    0   6079    0
mcast   65535 224.0.0.0/4        224.0.0.0             0    0      0    0
tunnel0  1500 127.0.0.5/32       127.0.0.5             0    0      0    0
vr0_main 1500 127.0.0.4/32       127.0.0.4             0    0      0    0
sip0    65535 -                  -                     0    0      0    0
sip1    65535 10.0.0.254/32      10.0.0.254            0    0      0    0
sip2    65535 15.9.7.254/32      15.9.7.254            0    0      0    0
```

The next set of commands configures `connection` profiles for the CPE units labeled `pipe1` and `pipe2` in Figure 4-4. These two CPE units share the same `source-if` address.

```
admin> new connection pipe1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 10.0.0.1/24
admin> set ip-options source-if = sip1
admin> set atm-options vci = 37
admin> set atm-options nailed-group = 301
admin> write -f
admin> new connection pipe2
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 10.0.0.2/24
admin> set ip-options source-if = sip1
admin> set atm-options vci = 37
admin> set atm-options nailed-group = 302
admin> write -f
```

The next set of commands configures `connection` profiles for the CPE units labeled `pipe3` and `pipe4` in Figure 4-4:

```
admin> new connection pipe3
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 15.9.7.1/24
admin> set ip-options source-if = sip2
```

```
admin> set atm-options vci = 37
admin> set atm-options nailed-group = 303
admin> write -f
admin> new connection pipe4
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 15.9.7.2/24
admin> set ip-options source-if = sip2
admin> set atm-options vci = 37
admin> set atm-options nailed-group = 304
admin> write -f
```

Anti-spoofing protection for IPoA, BIR, PPPoA, and PPPoE connections

IP spoofing is a method of sending IP packets with a faked IP source address to gain unauthorized access to systems, usually to cause problems such as denial-of-service attacks. With this software version, a simple anti-spoofing method is provided to protect the system from IP spoofing attempts across PPPoA, PPPoE, IPoA, and BIR connections.

Two levels of anti-spoofing protection are provided.

- Level 1: Source IP address checks

  At level 1, the system checks the source address of a packet against the remote address of the `connection` or RADIUS user profile, and drops packets whose source address doesn't match. The source IP address check is recommended for a connection to a CPE that connects to a single subnet, or for a restrictive access that allows only packets from the CPE itself or a host on its subnet across the WAN interface.

  The profile's address can be specific to a single host or a LAN, and the address can be statically specified in the profile or dynamically allocated via DHCP or address pools. However, source address checks cannot be used with DHCP over BIR connections.

Note Using source IP address checks or reverse path checks on a DHCP over BIR connection prevents the virtual interface from coming up.

- Level 2: Reverse path checks

  At level 2, the system performs the source address check of level 1, but also verifies reachability of the source address across the same interface on which the packet was received. If a route to the packet's source address is not found, the system drops the packet. The reverse path check is recommended for connections to a router that services different networks, or for a less restrictive access that allows addresses beyond the CPE subnet to be accepted on the WAN interface.

  Routes to source addresses can be configured explicitly through static routes or learned dynamically through any of the available routing protocols.

Anti-spoofing works on DSL and trunk interfaces, but does not work on the Gigabit Ethernet interface.

Overview of anti-spoofing settings

For PPP calls (PPPoE or PPPoA), anti-spoofing settings must be configured in the PPP

`connection` profile (not in the ATM circuit `connection` profile between the LIM interface and the internal ATM interface of the LIM). For IPoA or BIR connections, anti-spoofing settings must be configured in their respective `connection` profiles.

Following are the relevant parameters, shown with default values:

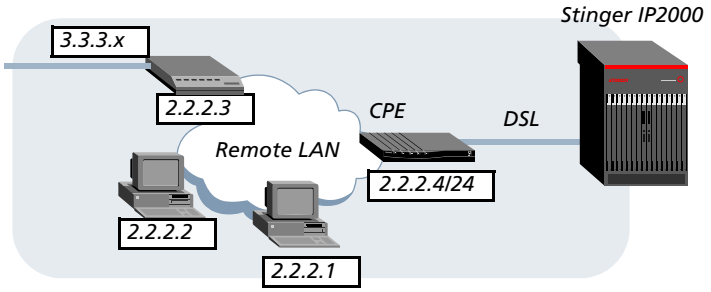
```
[in CONNECTION/"":ip-options]
source-ip-check = no
reverse-path-check = no
```

Parameter	RADIUS attribute	Setting
`source-ip-check`	Ascend-Source-IP-Check (96)	Enables or disables level 1 anti-spoofing protection. When it is enabled, the system compares the source address in packets received on the WAN interface to the profile's remote address, and drops nonmatching packets. By default, this protection is disabled. This setting and the `reverse-path-check` setting are mutually exclusive. Only one of these fields can be enabled in a profile. Use this setting to restrict traffic on the interface to packets from the profile's remote host or subnet.
`reverse-path-check`	Ascend-Reverse-Path-Check (310)	Enables or disables level 2 anti-spoofing protection. When it is enabled, the system checks the routing table to verify that it can reach the packet's source address across the same interface on which it received the packet, and drops nonmatching packets. By default, this protection is disabled. This setting and the `source-ip-check` setting are mutually exclusive. Only one of these fields can be enabled in a profile. Use this setting to allow all packets from the profile's remote host or subnet, as well as packets received from networks that are reachable through that interface.

Sample anti-spoofing configuration

With the sample network shown in Figure 4-5, the CPE must have a `connection` profile in the Stinger system. The profile can enable `source-ip-check` (level 1 protection) or `reverse-path-check` (level 2 protection) on a PPPoA call.

Figure 4-5. Sample network with two levels of anti-spoofing protection



- If `source-ip-check` is enabled for the subscriber connection, packets from 2.2.2.x/24 subnet will be accepted, but packets from 3.3.3.x network will be dropped.
- If `reverse-path-check` is enabled for the subscriber connection and a static route for 3.3.3.x is configured (or a routing protocol is enabled on the WAN interface to the CPE), packets from both the 2.2.2.x/24 subnet and the 3.3.3.x network will be accepted.

For example, the following commands enable an SDSL interface in slot 4 and configure an ATM circuit from the interface to the LIM's ATM internal interface. This is a required configuration for a PPPoA interface, but should not be configured for anti-spoofing protection.

```
admin> read sdsl { 1 4 1 }
admin> set enabled = yes
admin> write -f
admin> new connection cir-141
admin> set active = yes
admin> set atm-options atm1483type = aal5-vc
admin> set atm-options vci = 36
admin> set atm-options nailed-group = 151
admin> set atm-connect-options nailed-group = 2097
admin> write -f
```

The following commands create a PPPoA `connection` profile for level 1 anti-spoofing. Only packets from the CPE and hosts on its subnet will be accepted.

```
admin> new connection pppoa-1
admin> set active = yes
admin> set encapsulation-protocol = ppp
admin> set telco-options call-type = off
admin> set ppp-options recv-password = localpw
admin> set ip-options remote-address = 2.2.2.4/24
admin> set ip-options source-ip-check = yes
admin> write -f
```

Configuring IPoA subscriber connections

The CPE devices described in this section are IP-capable DSL devices that transmit IP over ATM (IPoA). IP over ATM is sometimes referred to as RFC 2684 traffic.

The CPE devices require a terminating PVC to the Stinger unit. A terminating PVC is not switched through the system, it terminates on the IP2000 controller and its data stream is passed up to the IP router for further handling.

The profile for a terminating connection must specify the IP address of the far-end router, and it can set a number of other routing-related values. The profile must also specify the ATM characteristics of the connection (for example, a VPI and VCI assignment and a nailed group representing the interface to use). The *Stinger ATM Configuration Guide* describes the ATM aspects of the configuration in detail.

Typical `atm-options` settings for terminating PVCs

For a discussion of ATM settings and quality of service (QoS) contracts, see the *Stinger ATM Configuration Guide*. Following are the ATM-related parameters, shown with default settings, for ATM terminating PVCs:

```
[in CONNECTION/""]

station = ""
active = no
encapsulation-protocol = atm-circuit

[in CONNECTION/"":atm-options]
atm1483type = aal5-llc
vpi = 0
vci = 35
nailed-group = 1
```

Parameter	RADIUS attribute	Setting
`station`	User-Name (1)	Name of the far-end device.
active	N/A	Enables or disables the profile.
`encapsulation-protocol`	Framed-Protocol (7)	Encapsulation protocol to use for the connection. Must specify ATM for terminating PVCs.
`atm1483type`	Framed-Protocol (7)	Stinger systems support the two encapsulation methods for carrying routed PDUs in the payload field of ATM adaptation layer (AAL) type 5, which are defined in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5.*
		The `aal5llc` setting indicates LLC encapsulation, which is used for non-PPP terminated connections such as RFC 2684 PVCs that terminate in the system. For PPP connections (PPPoE and PPPoA), the `aal5vc` setting is used, which indicates AAL5 VC multiplexing.
`vpi`	Ascend-ATM-Vpi (94)	VPI value for the PVC. For a discussion of valid values, see the *Stinger ATM Configuration Guide*.
`vci`	Ascend-ATM-Vci (95	VCI value for the PVC. For a discussion of valid values, see the *Stinger ATM Configuration Guide*.

Parameter	RADIUS attribute	Setting
`nailed-group`	Ascend-ATM-Group (64)	Nailed-group number of the interface used by the connection. You can obtain the nailed-group assigned to any interface by using the `which -n` command.

Typical `ip-options` settings for terminating PVCs

For information about enabling IP multicast forwarding on client or remote MBONE interfaces, see Chapter 8, "IP Multicast Configuration."

Following are the IP options (shown with default settings) for configuring an IP routed RFC 2684 connection to a DSL CPE:

```
[in CONNECTION/"":ip-options]
ip-routing-enabled = yes
vj-header-prediction = no
remote-address = 0.0.0.0/0
local-address = 0.0.0.0/0
routing-metric = 1
private-route = no
rip = routing-off
```

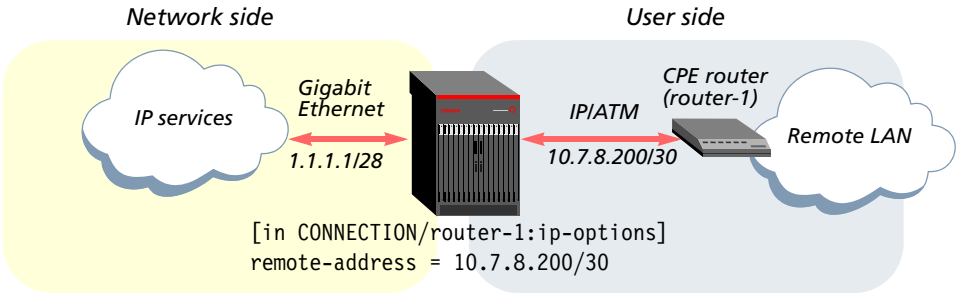
Parameter	RADIUS attribute	Setting
`ip-routing-enabled`	Ascend-Route-IP (228)	Enables or disables IP routing on the interface.
`vj-header-prediction`	Framed-Compression (13)	Enables or disables Van Jacobson prediction for TCP packets on incoming calls using encapsulation protocols that support Van Jacobson compression.
`remote-address`	Framed-IP-Address (8) Framed-IP-Netmask (9)	IP address of the remote CPE device.
`local-address`	Ascend-PPP-Address (253) Ascend-IF-Netmask (153)	Local IP address of a numbered interface connection. For a more flexible alternative to this setting, see "Source interface local addresses" on page 4-27.
`routing-metric`	Ascend-Metric (225)	RIP metric (1–15) for the specified route. If preference values are equal, the higher the metric, the less likely that the router will use the route.
`private-route`	Ascend-Route-Preference (126)	Include or exclude this route in RIP updates.
`rip`	Framed-Route (22)	Enables or disables RIP.

For details about parameters, see the *Stinger Reference*. For details about the attribute-value pairs used to configure IP options in RADIUS profiles, see the *TAOS RADIUS Guide and Reference*.

Sample RFC 2684 (IPoA) terminating PVC

Figure 4-6 shows a CPE router connection using IP over ATM.

Figure 4-6. Router-to-router IP connection



The default settings for the `ip-options` subprofile enable IP routing and Van Jacobson header compression and turn RIP off. Those settings are typically appropriate for a DSL interface, but they are not required.The following example shows configuration of a `connection` profile for the DSL CPE router in Figure 4-6:

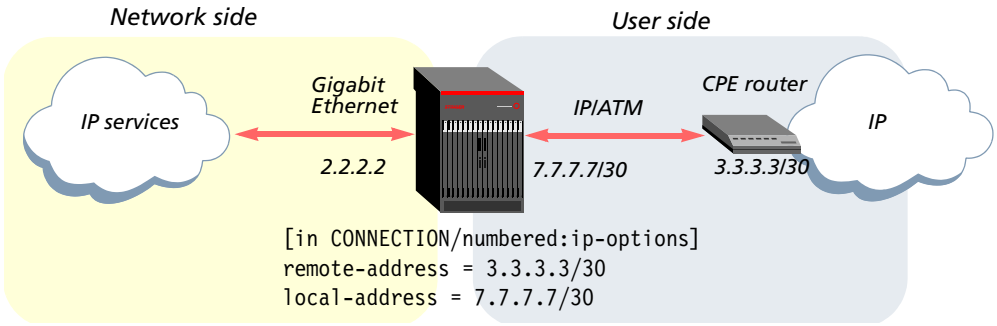
```
admin> read connection router-1

admin> set active = yes

admin> set encapsulation-protocol = atm

admin> set ip-options remote-address = 10.7.8.200/30

admin> set atm-options vpi = 8

admin> set atm-options vci = 100

admin> set atm-options nailed-group = 201

admin> write -f

permconn-st-1 Password = "ascend"
    Service-Type = Outbound,
    Framed-Protocol = ATM-1483,
    User-Name = "router-1",
    Framed-IP-Address = 10.7.8.200,
    Framed-IP-Netmask = 255.255.255.252,
    Ascend-ATM-Group = 201,
    Ascend-Route-IP = Route-IP-Yes,
    Ascend-ATM-Vpi = 8,
    Ascend-ATM-Vci = 100
```

Example of a numbered interface using `local-address`

A numbered-interface configuration assigns each side of the connection a unique address that applies only to that connection. Figure 4-7 shows a numbered-interface connection. The Stinger unit's real, physical Ethernet interface has the IP address 2.2.2.2. The other two addresses represent the local and remote addresses of the numbered-interface connection.

See "Source interface local addresses" on page 4-27 for information about local addresses that can be used by multiple subscriber connections across DSL interfaces.

Figure 4-7. A numbered-interface connection



Some applications such as SNMP use the `local-address` value internally to keep track of the circuit. The `local-address` value must be unique to the connection and to the network.

Note Do not assign a local address that belongs to one of the Stinger unit's real physical LAN interfaces. Doing so causes routing problems.

The following set of commands specifies a `connection` profile for the numbered interface:

```
admin> new connection numbered
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 3.3.3.3/30
admin> set ip-options local-address = 7.7.7.7/30
admin> set atm-options vci = 36
admin> set atm-options nailed-group = 211
admin> write -f
```

Following is a comparable RADIUS profile:

```
permconn-st-2 Password = "ascend"
    Service-Type = Framed-User,
    Framed-Protocol = ATM-1483,
    User-Name = "numbered",
    Ascend-ATM-Group = 211,
    Ascend-Route-IP = Route-IP-Yes,
    Ascend-ATM-Vpi = 0,
    Ascend-ATM-Vci = 36,
    Framed-IP-Address = 3.3.3.3,
    Framed-IP-Netmask = 255.255.255.252,
    Ascend-PPP-Addr = 7.7.7.7,
    Ascend-IF-Netmask = 255.255.255.252
```

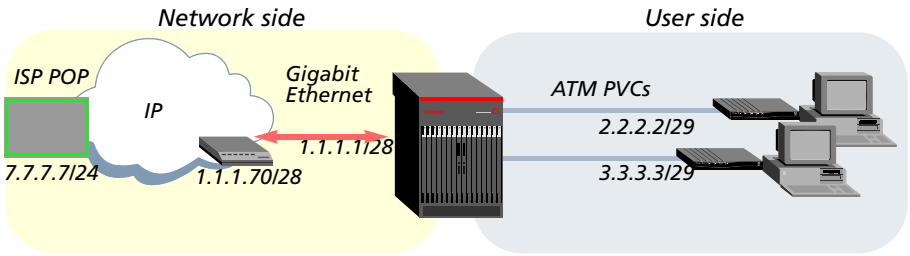
In this example, the interface is assigned a 30-bit subnet, so four bit combinations are available for host assignments. Of the four possible host addresses, the one that is evenly divisible by 4 is the network or base address (the address that specifies zeros in the host bits). This address is added to the routing table. The other host addresses are assigned a /32 subnet mask and added as host routes. You can suppress

advertisement of the host routes associated with a numbered interface by using the `suppress-host-routes` parameter, which is described in the *Stinger Reference*.

Example of routing a terminated PVC across Gigabit Ethernet

You can forward RFC 2684 PVCs from DSL subscribers onto the Gigabit Ethernet IP interface to be further routed to a specific IP destination such as an Internet service provider (ISP), as shown in Figure 4-8.

Figure 4-8. Forwarding terminating PVCs on the Gigabit Ethernet interface



This configuration requires a terminating PVC for each DSL subscriber. In this example, the Stinger does not maintain a large routing table itself. It uses a static route configuration to forward IP traffic across Gigabit Ethernet to another router, which routes the traffic on toward the ISP point of presence.

The following commands create a `connection` profile for each of the DSL subscribers in Figure 4-8:

```
admin> new connection user-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 2.2.2.2/29
admin> which -n { 1 2 1 }
Nailed group corresponding to port { shelf-1 slot-2 1 } is 51
admin> set atm-options nailed-group = 51
admin> write -f
admin> new connection user-2
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 3.3.3.3/29
admin> which -n { 1 2 2 }
Nailed group corresponding to port { shelf-1 slot-2 2 } is 52
admin> set atm-options nailed-group = 52
admin> write -f
```

The following command displays the `ip-interface` profile for the IP2000 Gigabit Ethernet interface, which shows that the address has been specified and RIP is not enabled:

```
admin> get ip-interface { { 1 8 2 } 0 }
[in IP-INTERFACE/{ { shelf-1 first-control-module 2 } 0 }]
```

```
interface-address* = { { shelf-1 first-control-module 2 } 0 }
ip-address = 1.1.1.1/28
proxy-mode = Off
rip-mode = routing-off
route-filter = ""
rip2-use-multicast = yes
ospf = { no 0.0.0.0 normal 10 40 5 simple ******* 0 1 16777215 type-1 c+
multicast-allowed = no
igmp-options = { 2 125 100 10 2 }
multicast-rate-limit = 100
multicast-group-leave-delay = 0
multicast-group-leave-delay-msec = 0
multicast-service-profile = ""
multicast-max-groups = 0
directed-broadcast-allowed = yes
vrouter = ""
management-only-interface = no
vlan-enabled = no
vlan-id = 0
```

The following set of commands configures a static route to the ISP's destination address, specifying a next-hop router on the Gigabit Ethernet interface:

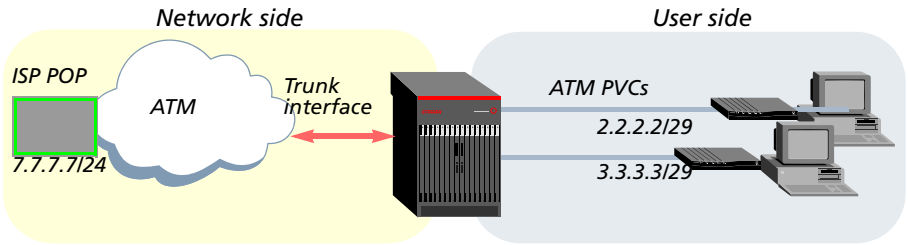
```
admin> read ip-route isp-dest
admin> set dest-address = 7.7.7.7/24
admin> set gateway-address = 1.1.1.70
admin> set active-route = yes
admin> write -f
```

With this example configuration, when packets destined for 7.7.7.7/24 are received on the terminating PVCs, the IP2000 consults its own routing table and forwards the packets onto its Gigabit Ethernet interface to the next-hop router specified as the `gateway-address`.

Example of using IP routing to aggregate PVCs onto a trunk VC

You can use IP routing to aggregate many RFC 2684 PVCs from DSL subscribers onto a single virtual circuit to a specific IP destination such as an ISP. Instead of configuring an ATM circuit for each subscriber, you use PVCs that terminate on the IP2000 and use IP routing to direct the traffic out on a terminating PVC to the ISP. This greatly simplifies provisioning new DSL subscribers that route to the same ISP.

Figure 4-9. Aggregating PVCs onto a single virtual circuit using IP routing



The following commands create a `connection` profile for each of the DSL subscribers in Figure 4-9:

```
admin> new connection user-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 2.2.2.2/29
admin> set atm-options vpi = 0
admin> set atm-options vci = 36
admin> set atm-options nailed-group = 51
admin> write -f
admin> new connection user-2
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 3.3.3.3/29
admin> set atm-options vpi = 0
admin> set atm-options vci = 37
admin> set atm-options nailed-group = 52
admin> write -f
```

The next command configures the PVC to the ISP:

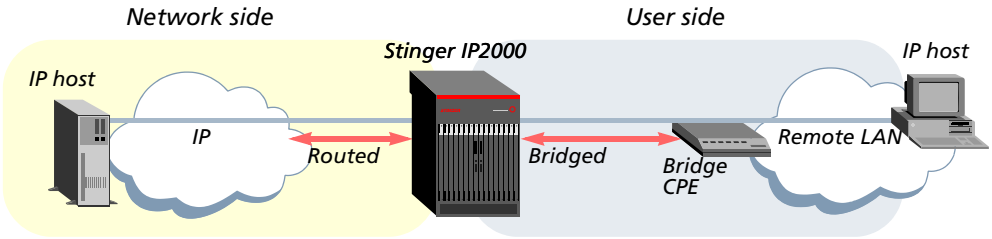
```
admin> new connection isp
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 7.7.7.7/24
admin> set atm-options vpi = 0
admin> set atm-options vci = 35
admin> which -n { 1 17 2}
Nailed group corresponding to port { shelf-1 trunk-module-1 2 } is 802
admin> set atm-options nailed-group = 802
admin> write -f
```

This `connection` profile creates a static route to the ISP's destination address across the trunk interface. When packets destined for 7.7.7.7/24 are received on the terminating PVCs, the IP2000 consults its own routing table and forwards the packets onto the trunk interface to the ISP.

Configuring BIR subscriber connections

With bridged IP routing (BIR), a Stinger IP2000 can establish an IP routed connection to an IP host through a customer premises equipment (CPE) bridge device. A BIR connection can use a line interface module (LIM) or trunk interface. A sample setup with a BIR interface on a LIM port is shown in Figure 4-10.

Figure 4-10. BIR interface on a LIM port



On the BIR interface, the system receives IP packets encapsulated in bridged frames. The IP2000 decapsulates the packets and passes them up the protocol stack to the IP router. To the IP host, the session appears to be an ordinary IP connection.

BIR configurations require the use of numbered interfaces, which assign both the remote and local side of the connection a unique IP address. The remote address can specify a subnet or an individual remote IP host. Typically, the local address for the Stinger unit is a unique address on the remote subnet. For details about numbered interfaces, see "Example of a numbered interface using local-address" on page 4-36. For related information, see "Source interface local addresses" on page 4-27.

Overview of `bir-options` and `ip-options` settings

In addition to the many possible IP routing parameters in `connection` and RADIUS profiles, described in "Configuring IPoA subscriber connections" on page 4-34, the following parameters apply to BIR interfaces. The parameters are shown with default settings.

```
[in CONNECTION/"":bir-options]
enable = no
proxy-arp = no
[in CONNECTION/"":ip-options]
ip-routing-enabled = yes
remote-address = 0.0.0.0/0
local-address = 0.0.0.0/0
```

Parameter	RADIUS attribute	Setting
`enable`	Ascend-BIR-Enable (70)	Enables or disables BIR on this interface.
`proxy-arp`	Ascend-BIR-Proxy (71)	Enables or disables proxy Address Resolution Protocol (ARP), which causes the Stinger IP2000 to respond as proxy for ARP requests from local hosts for remote hosts on the far end of the link.
`ip-routing-enabled`	Ascend-Route-IP (228)	Enables or disables IP routing on the interface.
`remote-address`	Framed-IP-Address (8) Framed-IP-Netmask (9)	IP address of the remote device, which can include a subnet specification. If the address does not include a subnet mask, the router assumes the default subnet mask based on address class.

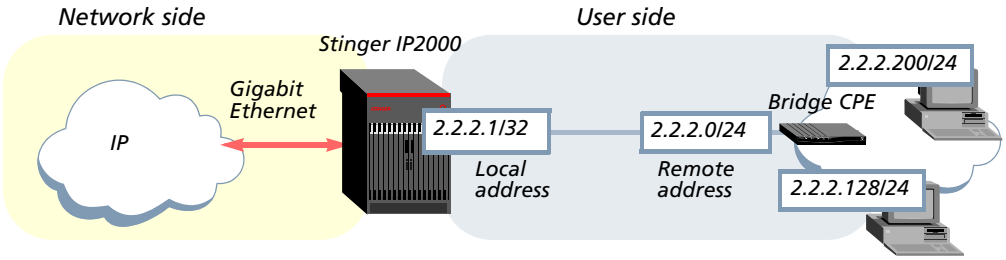
Parameter	RADIUS attribute	Setting
`local-address`	Ascend-PPP-Address (253) Ascend-IF-Netmask (153)	IP address assigned to the local side of a numbered-interface connection. This is a requirement for BIR interfaces. For a more flexible alternative setting, see "Source interface local addresses" on page 4-27.

Sample subnet (BIR/24) configuration

A BIR subnet configuration specifies a remote subnet address, and can be used to transmit bridged data to multiple IP hosts on that subnet.

When the Stinger IP2000 receives a packet destined for a BIR subnet interface, it examines the network bits of the destination address and forwards the packet to the related CPE. For example, Figure 4-11 shows two bridging CPE devices connected to an IP class C subnet. With this example, if the IP2000 receives a packet addressed to 2.2.2.200 or 2.2.2.128, it examines only the first 24 bits of the address, and forwards the packets to the bridge CPE.

Figure 4-11. BIR subnet configuration on LIM interface



The following commands configure a BIR subnet interface through the DSL CPE bridge in Figure 4-11:

```
admin> new connection bir-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 2.2.2.0/24
admin> set ip-options local-address = 2.2.2.1/32
admin> set bir-options enable = yes
admin> set bir-options proxy-arp = yes
admin> set atm-options atm1483type = aal5-llc
admin> set atm-options vci = 101
admin> which -n { 1 2 1 }
Nailed group corresponding to port { shelf-1 slot-2 1 } is 51
admin> set atm-options nailed-group = 51
admin> write -f
```

Following is a comparable definition in a RADIUS profile:

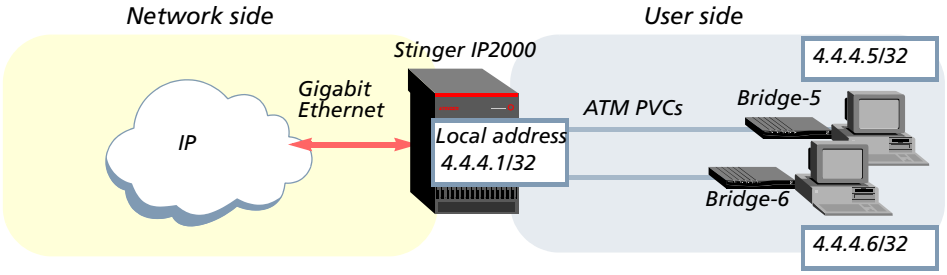
```
permconn-cpe-1 Password = "ascend"
    Service-Type = Outbound,
```

```
Framed-Protocol = ATM-1483,
User-Name = "bir-1",
Ascend-Route-IP = Route-IP-Yes,
Framed-IP-Address = 2.2.2.0,
Framed-IP-Netmask = 255.255.255.0,
Ascend-PPP-Addr = 2.2.2.1,
Ascend-IF-Netmask = 255.255.255.255,
Ascend-ATM-Group = 51,
Ascend-ATM-Vci = 101,
Ascend-BIR-Enable = BIR-Enable-Yes,
Ascend-BIR-Proxy = BIR-Proxy-Yes
```

Sample host route (BIR/32) configurations

When a Stinger IP2000 receives a packet to a BIR/32 interface, it examines the full 32 bits of the destination address and forwards the packet to the related CPE. Figure 4-12 shows two bridging DSL CPE devices, each supporting one host. In this example, the IP hosts have addresses on the same IP network, but that is not a requirement.

Figure 4-12. BIR/32 configurations



In Figure 4-12, the `local-address` value is the same for both BIR interfaces. This is recommended for host routes to the same IP network because it simplifies configuration of the remote hosts, all of which can point to the same local address as the gateway.

The following commands configure a BIR/32 interface through the CPE labeled *Bridge-5* in Figure 4-12:

```
admin> new connection bir-5
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 4.4.4.5/32
admin> set ip-options local-address = 4.4.4.1/32
admin> set atm-options atm1483type = aal5-llc
admin> set atm-options vci = 111
admin> set bir-options enable = yes
admin> which -n { 1 2 5 }
Nailed group corresponding to port { shelf-1 slot-2 5 } is 55
admin> set atm-options nailed-group = 55
admin> write -f
```

The following commands modify the `connection` profile immediately above to configure a BIR/32 interface through the CPE labeled *Bridge-6*:

```
admin> set station = bir-6
(New index value; will save as new profile CONNECTION/bir-6.)
admin> set ip-options remote-address = 4.4.4.6/32
admin> set atm-options vci = 112
admin> which -n { 1 2 6 }
Nailed group corresponding to port { shelf-1 slot-2 6 } is 56
admin> set atm-options nailed-group = 56
admin> write -f
```

Following are comparable definitions in RADIUS profiles:

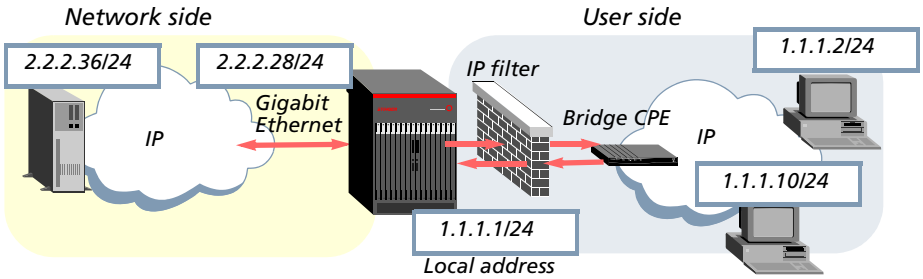
```
permconn-cpe-5 Password = "ascend"
    Service-Type = Outbound,
    Framed-Protocol = ATM-1483,
    User-Name = "bir-5",
    Ascend-Route-IP = Route-IP-Yes,
    Framed-IP-Address = 4.4.4.5,
    Framed-IP-Netmask = 255.255.255.255,
    Ascend-PPP-Addr = 4.4.4.1,
    Ascend-IF-Netmask = 255.255.255.255,
    Ascend-ATM-Group = 55,
    Ascend-ATM-Vci = 111,
    Ascend-BIR-Enable = BIR-Enable-Yes
permconn-cpe-6 Password = "ascend"
    Service-Type = Outbound,
    Framed-Protocol = ATM-1483,
    User-Name = "bir-6",
    Ascend-Route-IP = Route-IP-Yes,
    Framed-IP-Address = 4.4.4.6,
    Framed-IP-Netmask = 255.255.255.255,
    Ascend-PPP-Addr = 4.4.4.1,
    Ascend-IF-Netmask = 255.255.255.255,
    Ascend-ATM-Group = 56,
    Ascend-ATM-Vci = 112,
    Ascend-BIR-Enable = BIR-Enable-Yes
```

Sample use of filters with BIR connections

You can apply an IP filter to restrict outbound packets on a BIR interface. However, IP filters are not applied to the inbound packet stream on a BIR interface. For details about defining IP filters, see "Filter Configuration" on page 10-1.

Figure 4-13 shows a sample BIR interface to a subnet that supports two IP hosts.

Figure 4-13. Bidirectional filtering on a BIR interface



The filter defined in this example is applied to the BIR interface. The input filter rules affect packets received on the BIR interface, and output filter rules affect packets destined for the user-side subnet.

The first input filter rule shown below specifies that if the destination IP address in a packet is 2.2.2.0/24, the protocol is 17 (UDP), and the source UDP port is less than 50, the packet is discarded. So, packets that match this rule will not reach the server at 2.2.2.36. The second input filter is an explicit default rule that forwards all other IP packets received on the BIR interface.

```
admin> new filter udp-filter
admin> set input-filters 1 valid-entry = yes
admin> set input-filters 1 forward = no
admin> set input-filters 1 Type = ip-filter
admin> set input-filters 1 ip-filter protocol = 17
admin> set input-filters 1 ip-filter dest-address-mask = 255.255.255.0
admin> set input-filters 1 ip-filter dest-address = 2.2.2.36
admin> set input-filters 1 ip-filter Src-Port-Cmp = less
admin> set input-filters 1 ip-filter source-port = 50
admin> set input-filters 2 valid-entry = yes
admin> set input-filters 2 forward = yes
admin> set input-filters 2 Type = ip-filter
```

The first output filter rule shown below specifies that if the source IP address in a packet is 2.2.2.36/24, the protocol is 17, and the source UDP port is less than 50, the packet is discarded. So, packets that match this rule will not reach the IP hosts across the BIR interface. The second output filter is an explicit default rule that forwards all other IP packets destined for the remote subnet through the BIR interface.

```
admin> set output-filters 1 valid-entry = yes
admin> set output-filters 1 forward = no
admin> set output-filters 1 Type = ip-filter
admin> set output-filters 1 ip-filter source-address-mask = 255.255.255.0
admin> set output-filters 1 ip-filter source-address = 2.2.2.36
admin> set output-filters 1 ip-filter protocol = 17
admin> set output-filters 1 ip-filter Src-Port-Cmp = less
admin> set output-filters 1 ip-filter source-port = 50
```

```
admin> set output-filters 2 valid-entry = yes
admin> set output-filters 2 forward = yes
admin> set output-filters 2 Type = ip-filter
admin> write -f
```

The following commands create a BIR profile to the bridge CPE in Figure 4-13, and apply the sample filter:

```
admin> new connection bir-1-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 1.1.1.0/24
admin> set ip-options local-address = 1.1.1.1/24
admin> set session-options data-filter = udp-filter
admin> set bir-options enable = yes
admin> set atm-options nailed-group = 101
admin> write -f
```

Configuring DHCP relay for IPoA and BIR connections

DHCP relay enables the Stinger system to transfer messages between a client requesting a configuration for an IPoA or BIR connection and a DHCP server. To enable DHCP requests on BIR connections, the system creates one virtual interface for each remote DHCP client.

RFC compliance and caveats

Stinger support for DHCP relay with option 82 is compliant with the following RFCs, with the exception noted for RFC 1542:

- RFC 951, *Bootstrap Protocol (BOOTP)*
- RFC 1542, *Clarifications and Extensions for the Bootstrap Protocol*
- RFC 2131, *The Dynamic Host Configuration Protocol (DHCP)*
- RFC 2132, *DHCP Options and BOOTP Vendor Information Extensions*
- RFC 3046, *DHCP Relay Agent Information Option*

Note When virtual interfaces are in use on a BIR connection, the system is forced to adopt behavior that is noncompliant with section 4.1 of RFC 1542, which indicates that if a relay agent receives a DHCP packet for which the destination IP address is not one of the interfaces of the relay-agent, the packet must be forwarded without any changes. Instead, the Stinger catches the packet and adds the relay-agent information to it.

DHCP option 82

DHCP option 82, the relay agent Information option, is used to associate a unique identifier with a broadband device such as a DSL CPE or Integrated Access Device (IAD). DHCP servers that recognize this option can use the option 82 identifier to enforce conditions on address or configuration access.

Note Option 82 specifies that the identifier can be associated with the virtual circuit to a remote device, or with the remote router itself. However, the Stinger currently supports only the virtual circuit identifier.

The identifier consists of the specified IP address concatenated to the connection name on which the DHCP request packet was received. The connection name is used by the relay agent to route back the packet.

Virtual IP interfaces on BIR connections

The system creates virtual IP interfaces for a BIR connection only when all of the following conditions are verified:

- The system receives a DHCP reply packet containing option 82 from the DCHP server.
- The `ip-global` profile `allow-multiple-interface-creation` field is set to `yes`.
- The BIR `connection` profile has DHCP relay enabled.
- The number of the created virtual interface is less than the configured one.
- If the `message-type` option is present in the DHCP packet, it must be set to DHCP-ACK. If the option is not present, the system assumes DCHP-ACK.

Direct routes associated with the virtual interface are set up during the interface creation. These dynamically created interfaces will be typed `virtual`, like virtual interfaces on an Ethernet port.

Virtual interface IP address assignments

Virtual interface addresses have the following characteristics:

- The IP address of the virtual interface is obtained from the `connection` profile if the `local-address` setting is not null. Otherwise, it is extracted from option 3 (the router option) of the DHCP packet.
- The destination IP address of a virtual interface is obtained from the `yiaddr` field of the DHCP reply message.
- The netmask is obtained from option 1 (the netmask option) if present in the DHCP reply packet. If option 1 is not provided, the netmask will be the default mask associated with the IP address class.
- The broadcast address is obtained from option 28 (the broadcast option) if present in the DHCP reply packet. If option 28 is not provided, the broadcast address will be calculated from the destination IP address and the netmask. The host part of the destination address will have all bits set to 1.

IP address lease time

The address for a virtual IP interface is leased via DHCP for a number of seconds. The lease time is extracted from the DCHP packet, option 51, and can be extended for an interface by setting the `additional-lease-delay` parameter. The maximum lease time cannot exceed 42,949,672 seconds, which translates to11930 hours (497 days). This limit is set because Stinger ticks are calibrated to 10 ms with a maximum value of a 32 bits/100 = 42949672. The Stinger interprets a lease time of 0 as an infinite lease.

If a virtual interface is up and active for a given period, and a DHCP-ACK is received during its active period, then the lease time is updated by the new value. The interface then remains up for the new received lease time.

When virtual interfaces are activated

The virtual interfaces are activated if one of the following conditions occurs. Activating a virtual interface implies creation of associated routes.

- The interface has just been created.
- The interface was inactive and a DHCP reply has been received for the same host, same IP address.
- All associated virtual interfaces are activated when the main interface was inactive and becomes active. If the leased time has expired for an associated virtual interface, that virtual interface is deleted.

How the system selects an interface for incoming packets

The virtual input interface selection for incoming packets is done in the following order:

1 If the source IP address located in the packet matches one of virtual interface remote IP addresses (taking into account the netmask), then the matching virtual interface is selected as the input interface.

2 If the destination IP address located in the packet matches the virtual interface local IP address (taking into account the netmask), then the matching virtual interface is selected as input interface.

3 If there is no match, the input interface will be the main interface.

When virtual interfaces are in use, the statistics are globally counted by the main interface even if the packets are received on virtual interfaces.

Deactivating and deleting virtual IP interfaces

All virtual interfaces are deactivated if the main interface becomes inactive. Deactivating a virtual interface implies a suppression of associated routes.

The interfaces are deleted if one of the following condition occurs:

- The main interface is deleted.
- The lease time has elapsed.

  The information about lease time is copied into the virtual interface. Both the shelf and the slot have this information. However, normally the LIM has the responsibility of releasing the interface if the time is elapsed. The copy on the shelf is present if the slot is reset and an update is required.
- A DHCP-RELEASE or DHCP-DECLINE message is relayed by the relay-agent.

Note If you must manually delete a virtual interface on a BIR connection, see “ifmgr” on page A-15.

DHCP relay configuration settings

DHCP relay configuration requires settings in the `ip-global` profile. In addition, LAN and WAN interfaces have settings for enabling or disabling `bootp-relay`. It is enabled by default on LAN interfaces, and disabled by default on WAN interfaces.

Overview of `ip-global` DHCP relay settings

Note Option 82 specifies that the identifier can be associated with the virtual circuit to a remote device, or with the remote router itself. However, the Stinger currently supports only the virtual circuit identifier. Although the `relay-agent-information` subprofile contains both `circuit-id` and `remote-id` configurations, only the `circuit-id` configuration is used.

The following `ip-global` parameters (shown with default settings) configure DHCP relay. All settings within the `relay-agent-information` subprofile (and its subprofiles) apply only when DHCP option 82 is in use.

```
[in IP-GLOBAL:bootp-relay]
active = no
bootp-servers = [ 0.0.0.0 0.0.0.0 ]

[in IP-GLOBAL:bootp-relay:relay-agent-information]
enable = no
allow-multiple-interface-creation = no
circuit-id = { no 0.0.0.0 1 no }
remote-id = { no 0.0.0.0 1 no }
giaddr-selection = system-ip-address

[in IP-GLOBAL:bootp-relay:relay-agent-information:circuit-id]
enable = no
if-ip = 0.0.0.0
version = 1
dhcp-allow-any-src-port = no
```

Parameter	Setting
`bootp-relay active`	Enables or disables DHCP relay. When this parameter is set to `yes`, the IP2000 forwards requests from a client on one network (such as a remote interface) to a DHCP server on another network interface.
`bootp-relay bootp-servers[1]/[2]`	These indexed parameters specify the IP address of one DHCP server. Only one address is required. **Note** Redundancy of DHCP servers is not currently supported.
`relay-agent-information enable`	Globally enable or disable DHCP option 82.
`relay-agent-information allow-multiple-interface-creation`	When set to `yes`, the relay agent can create new virtual IP interfaces on BIR connections each time a new DHCP request has been successfully serviced. The `relay-agent-information` field must also be set to `yes`, and the `remote-address` field in the BIR `connection` profile must be null. (The `local-address` field can be set to a valid IP address.) If this parameter is set to `no`, only one interface is associated with a BIR connection.

Parameter	Setting
`relay-agent-information giaddr-selection`	This parameter specifies which address to use as the gateway address to populate the `giaddr` field in DHCP packets. The setting applies only when option 82 is configured. **Note** When DHCP over BIR is used without option 82, the `giaddr` field is set to the `remote-ip-address` configured in the BIR connection profile. In this case, the DHCP server must know the route to that address. By default, the system address is used as the gateway address, which is also the source address of the DHCP packets. Changing this parameter changes the contents of the `giaddr` field, not the source address of the packets. Specifying the soft address or local interface address enables the DHCP server to reply to an address rather than the system address. For more details, see "Sample DHCP relay configurations for BIR connections" on page 4-56. Following are valid settings: `system-ip-address` — Use the configured system IP address (the default). If the system IP address is set to the address of the management interface, the DHCP server must be accessible across that interface. `soft-ip-address` — Use the IP address configured in the `ip-interface` profile with the zero index ({{`a a 0` }}), which is the soft interface. If the soft interface is not configured, the system uses the null address (0.0.0.0). The system does not consider virtual interfaces with the zero index ({{`a a 0`}*x*}). `local-ip-address` — Use the IP address of the interface on which the DHCP server is accessible.
`relay-agent-information: remote-id`	The `remote-id` subprofile is currently not used.
`relay-agent-information: circuit-id:enable`	Enables or disables the circuit identifier suboption of DHCP option 82. If the circuit ID is enabled, the IP2000 encodes the `station` value (the hostname) of the profile that defines the PVC on which the DHCP client-to-server packet was received. This ensures that DHCP responses are sent back to the proper circuit.

Parameter	Setting
`relay-agent-information: circuit-id:if-ip`	IP address of one of the IP2000 IP interfaces. If both IDs are enabled, only one interface IP address is needed. If this field is empty, the Stinger uses the system address (`ip-global:system-ip-addr`) if that value has been defined. To interoperate with DHCP servers that zero-delimit suboption fields, the specified IP address cannot contain a zero octet.
`relay-agent-information: circuit-id:version`	Relay agent version ID. To interoperate with DHCP servers that zero-delimit suboption fields, change this from the default 1 to a value of 257 or higher.
`relay-agent-information: circuit-id:dhcp-allow-any-src-port`	If set to `yes`, enables the system to process DHCP requests from a nonstandard source port, and to add option 82 to such requests.

Overview of **`ip-interface`** and **`connection`** DHCP settings

Following are the parameters, shown with default settings, for enabling the use of DHCP across a LAN or WAN interface:

```
[in IP-INTERFACE/{ { shelf-1 first-control-module 1 } 0 }]
bootp-relay-enable = yes
```

```
[in CONNECTION/"":ip-options:bootp-relay-options]
enable = no
max-dynamic-interface = 0
additional-lease-delay = 15
```

Parameter	Setting
`bootp-relay-enable`	Enables or disables relay of DHCP requests and replies received on the LAN interface. The LAN interface on which the DHCP server is accessible must have this parameter set to `yes`, which is the default setting. Setting this parameter to `no` prevents relay of DHCP packets received on the interface. It does not prevent the system from sending relayed packets on the interface. This parameter has no impact if the Stinger is either a DHCP client or server.
`enable`	Enables or disables DHCP relay requests on the interface. This parameter applies to DHCP relay requests on both BIR connections and IP terminated virtual circuits.
`max-dynamic-interface`	Maximum number of dynamic interfaces that can be created by the DHCP ACK packet on the connection. If null, dynamic interface creation is disabled on this interface. This parameter applies only to BIR connections.

Parameter	Setting
`additional-lease-delay`	Number of seconds to keep the virtual interface up after the lease time (extracted from the DCHP packet, option 51) has expired. A non-null value allows the DHCP client to request a new IP address after the previous IP address lease time has expired without a shutdown of the virtual interface. Typically, the DHCP server reallocates the same IP address to a client. The maximum lease time is 42,949,672 seconds, which translates to11930 hours (497 days). The Stinger interprets a lease time of 0 as an infinite lease. This parameter applies only to BIR connections.

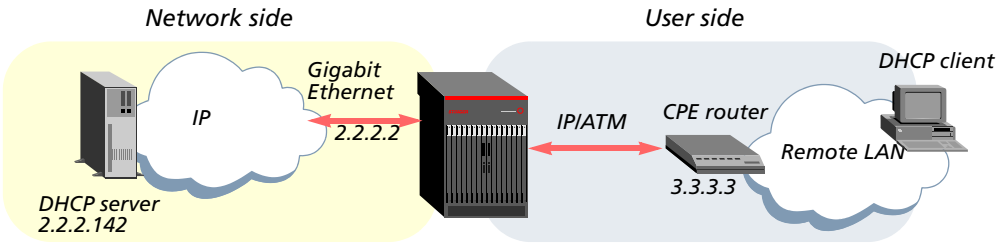
Sample DHCP relay configurations for IPoA connections

To support centralized assignment of the IP addresses of host PCs via DHCP, both the remote CPE router and the IP2000 must be configured as DHCP relay agents. The sample configurations in this section show how to use DHCP relay with or without option 82.

Sample configuration using DHCP relay without option 82

Figure 4-14 shows a high-speed circuit to a remote CPE router with a LAN interface to one or more host PCs.

Figure 4-14. DHCP relay for an IPoA terminated PVC



In this example, the remote PC is configured to use DHCP to obtain an IP address. The CPE router on the remote LAN is configured statically with the IP address 3.3.3.3 and is configured to enable DHCP relay. The CPE router DHCP relay configuration specifies a DHCP server at 2.2.2.142 (the DHCP server beyond the DHCP relay agent on the IP2000).

The IP2000 is configured as DHCP relay agent with the DHCP server at 2.2.2.142 across its Gigabit Ethernet interface (2.2.2.2). The DHCP server is configured to recognize the CPE router at 3.3.3.3.

The following commands configure the IP2000 as a DHCP relay agent:

```
admin> read ip-global
admin> set bootp-relay active = yes
admin> set bootp-relay bootp-servers 1 = 2.2.2.142
admin> write -f
```

The next commands configure the `connection` profile for the CPE router:

```
admin> new connection cpe-router
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 3.3.3.3/30
admin> set ip-options bootp-relay-options enable = yes
admin> set atm-options nailed-group = 251
admin> write -f
```

Sample configuration using DHCP relay with option 82

This example builds on the sample DHCP relay configuration shown above (Figure 4-14). The `connection` profile to the CPE router in that sample configuration does not require any changes to support option 82.

For sites that support option 82, the DHCP server configuration typically requires the presence of an ID in DHCP queries. For example, the DHCP server in this example is configured to recognize the CPE router at 3.3.3.3 across interface 7.7.7.7, and to require a circuit ID. If the DHCP request forwarded to the server by the IP2000 does not contain the circuit ID, the server refuses to return an address.

The following commands configure the system to enable the circuit identifier suboption of DHCP option 82 and specify the Gigabit Ethernet address as the ID (Figure 4-14):

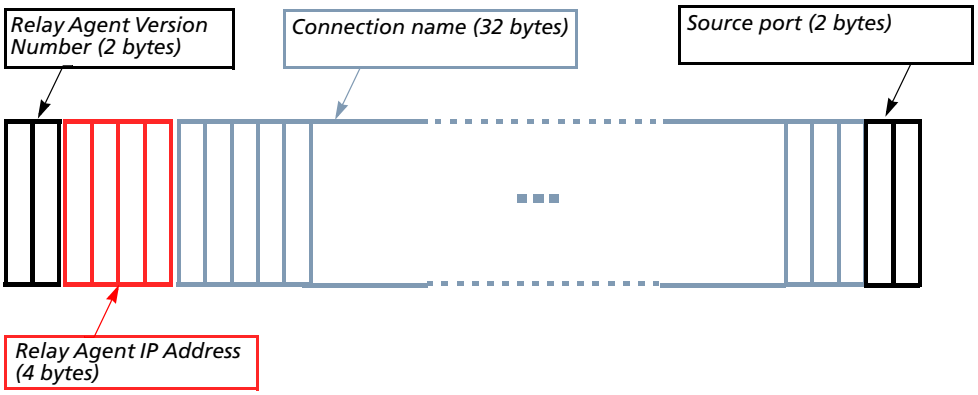
```
admin> read ip-global
admin> set bootp-relay relay-agent-information circuit-id enable = yes
admin> set bootp-relay relay-agent-information circuit-id if-id = 2.2.2.2
admin> write -f
```

Interoperation with DHCP servers that zero-delimit suboption fields

Some DHCP servers treat option-82 suboption information fields as zero-delimited strings rather than as an array of bytes. When a suboption information field (`circuit-id` and `remote-id`) contains a zero byte in the middle of the field, these DHCP servers consider the zero byte to be the end of the option-82 string.

Figure 4-15 shows the format Stinger systems use for the DHCP option-82 suboption information field (`circuit-id` and `remote-id`):

Figure 4-15. Option 82 information field formats



The two-byte version number of the relay-agent information ID is set by default to 1, which is coded internally as `0x00 0x01`.

The four-byte relay-agent IP address is set to an IP address in use in the system.

To avoid conflicts with DHCP servers that zero-delimit suboption fields, the version number setting is configurable by the operator. In addition, operators must be sure not to assign an IP address that contains a zero byte. For example, the following commands set the circuit identifier suboption `version` field to 257:

```
admin> read ip-global
admin> set bootp-relay relay-agent-information circuit-id version = 257
admin> write -f
```

Allowing non-standard DHCP source ports

To enable the Stinger to process DHCP requests from a nonstandard source port, and to add option 82 (which includes the source port number) to such requests, you can configure the system to allow any DHCP source port. For example:

```
admin> read ip-global
admin> set bootp-relay relay-agent-information circuit-id dhcp-allow-any-
src-port = yes
admin> write -f
```

DHCP issues on LAN management interfaces

The management interface of the Stinger unit is the IP interface used to access the system for management or configuration tasks, usually via `telnet` or an SNMP manager. It might be the 10/100 Ethernet port, the Gigabit Ethernet port of the IP2000, or an ATM trunk interface.

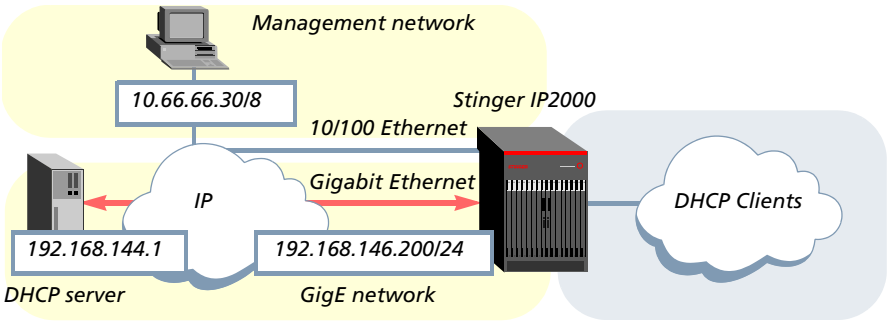
Often the system address is set to the IP address of the management interface. By default, this is also the address that populates the `giaddr` field in DHCP packets the system relays out on a LAN interface.

If the system address is set to the management interface and the DHCP server is not accessible across that interface, you should specify either the soft IP address or the local interface address to populate the `giaddr` field of DHCP packets. This enables the server to reply to an address rather than the system address.

You can also disable forwarding of DHCP packets received on the management network interface, or any LAN interface.

The sample configuration in this section uses the network setup and IP addresses shown in Figure 4-16:

Figure 4-16. Sample DHCP usage with LAN management interface



The following commands configure the management network interface. DHCP relay is blocked on this interface.

```
admin> read ip-interface { { 1 first-control-module 1 } 0 }
admin> set ip-address = 10.66.66.30/8
admin> set bootp-relay-enable = no
admin> write -f
```

The following commands configure the Gigabit Ethernet network interface. This is the interface on which the DHCP server is accessible, so DHCP relay must be enabled (as it is by default).

```
admin> read ip-interface { { 1 first-control-module 2 } 0 }
admin> set ip-address = 192.168.144.200/24
admin> write -f
```

The following commands set the system address to the management network address, configure the DHCP server address, and set the `giaddr` address to the local interface on which the DHCP server is accessible.

```
admin> read ip-global
admin> set system-ip-addr = 10.66.66.30/8
admin> set bootp-relay active = yes
admin> set bootp-relay bootp-servers 1 = 192.168.144.1
admin> set bootp-relay relay-agent enable = yes
admin> set bootp-relay relay-agent giaddr-selection = local-ip-address
admin> write -f
```

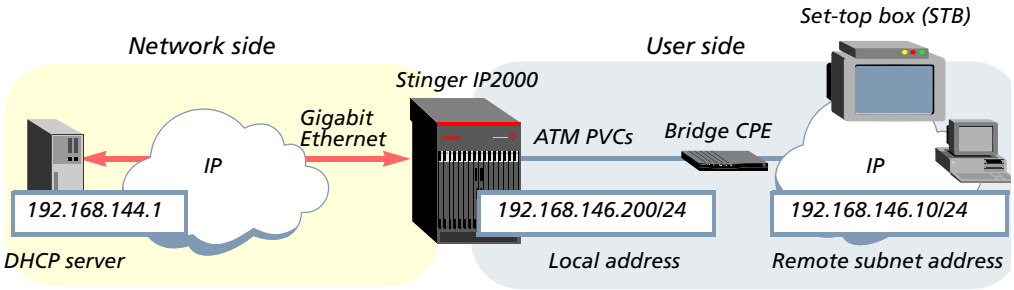
Sample DHCP relay configurations for BIR connections

The sample configurations in this section are presented in order of increasing complexity. They are intended to show how the system behaves with specific global and connection settings.

Note In each sample configuration, connection-specific configurations other than the settings shown are meaningless and forbidden.

All configurations in this section use the basic network setup and IP addresses shown in Figure 4-17:

Figure 4-17. Sample DHCP setup on BIR connection



Sample configuration with no DHCP relay on a BIR connection

In this sample configuration, the following conditions apply:

- Relay-agent is disabled on this interface. If the system receives DHCP requests on the interface, it discards them.
- Each remote host on the BIR connection has a static IP configuration.

The following set of commands configures the system to act as DHCP relay agent, but does not enable option 82 or multiple interface creation on BIR connections:

```
admin> read ip-global
admin> set bootp-relay active = yes
admin> set bootp-relay bootp-servers 1 = 192.168.144.1
admin> set bootp-relay relay-agent-information enable = no
admin> set bootp-relay relay-agent-information allow-multiple-interface = no
admin> set bootp-relay relay-agent-information circuit-id enable = no
admin> set bootp-relay relay-agent-information circuit-id if-ip = 0
admin> write -f
```

The following commands configure the BIR connection and disable DHCP relay requests:

```
admin> read connection
admin> set station = bir-host-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 192.168.146.10/24
admin> set ip-options local-address = 192.168.146.200/24
```

```
admin> set ip-options bootp-relay-options enable = no
admin> set ip-options bootp-relay-options max-dynamic-interface = 0
admin> set atm-options nailed-group = 3
admin> set bir-options enable = yes
admin> write -f
```

Sample configuration enabling relay agent on a BIR connection

In this sample configuration, the following conditions apply:

- Relay-agent is enabled on this interface. DHCP option 82 is not enabled.
- The `giaddr` field of the DHCP packet will contain the `local-address` of the interface on which the request was made. This address will be used to route back the DCHP reply.
- Several hosts acting as DHCP clients can be configured on this BIR connection, but each host must be on the same IP network. In this example, the IP network is 192.168.146.0.
- All hosts share the same gateway located at 192.168.146.200.
- Some hosts on this connection can have a static IP configuration.
- The Stinger has only one interface associated with the BIR connection.

The following set of commands configures the system to act as DHCP relay agent, but does not enable option 82 or multiple interface creation on BIR connections.

```
admin> read ip-global
admin> set bootp-relay active = yes
admin> set bootp-relay bootp-servers 1 = 192.168.144.1
admin> set bootp-relay relay-agent-information enable = no
admin> set bootp-relay relay-agent-information allow-multiple-interface = no
admin> set bootp-relay relay-agent-information circuit-id enable = no
admin> set bootp-relay relay-agent-information circuit-id if-ip = 0
admin> write -f
```

The following commands enable relay agent requests on the BIR connection, but do not enable option 82.

```
admin> read connection
admin> set station = bir
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 192.168.146.10/24
admin> set ip-options local-address = 192.168.146.200/24
admin> set ip-options bootp-relay-options enable = yes
admin> set ip-options bootp-relay-options max-dynamic-interface = 0
admin> set atm-options nailed-group = 3
admin> set bir-options enable = yes
admin> write -f
```

Sample configuration with option 82

In this sample configuration, the following conditions apply:

- Relay-agent is enabled on this interface.
- DHCP option 82 is enabled. The `giaddr` field of the DHCP packet will contain the Stinger `system-ip-address`, and the option 82 identifier will be inserted in the DHCP packet. Option 82 contains enough information to route back the DHCP reply.
- Several hosts acting as DHCP clients can be configured on this BIR connection.
- All hosts share the same gateway located at 192.168.146.200.
- Some hosts on this connection can have a static IP configuration.
- The Stinger has only one interface associated with the BIR connection.

The following set of commands configures the system to act as DHCP relay agent, and enables DHCP option 82 in the system. The option 82 identifier inserted into DHCP packets will be associated with the virtual circuit to the remote device.

```
admin> read ip-global
admin> set bootp-relay active = yes
admin> set bootp-relay bootp-servers 1 = 192.168.144.1
admin> set bootp-relay relay-agent-info enable = yes
admin> set bootp-relay relay-agent-info allow-multiple-interface = no
admin> set bootp-relay relay-agent-info circuit-id enable = yes
admin> set bootp-relay relay-agent-info circuit-id if-ip = 192.168.146.200
admin> write -f
```

The following commands enable relay agent requests on the BIR connection:

```
admin> read connection
admin> set station = bir
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 192.168.146.10/24
admin> set ip-options local-address = 192.168.146.200/24
admin> set ip-options bootp-relay-options enable = yes
admin> set ip-options bootp-relay-options max-dynamic-interface = 0
admin> set atm-options nailed-group = 3
admin> set bir-options enable = yes
admin> write -f
```

Sample configuration with option 82 and multiple interface creation

In this sample configuration, the following conditions apply:

- Relay-agent is enabled on this interface.
- DHCP option 82 is enabled. The `giaddr` field of the DHCP packet will contain the Stinger `system-ip-address`, and the option 82 identifier will be inserted in the DHCP packet. Option 82 contains enough information to route back the DHCP reply.
- Several hosts acting as DHCP clients can be configured on this BIR connection.
- All hosts share the same gateway located at 192.168.146.200.
- Static IP configuration for hosts on the BIR connection is forbidden.
- The Stinger has the same number of interfaces associated with the BIR connection as there are remote hosts. In this example, the BIR connection supports up to three virtual interfaces for three DHCP clients.

The system creates a new interface for each fully serviced DHCP request. In this sample configuration, the IP address is obtained from the `local-address` defined in the `connection` profile of the configured main interface, and the remote address is obtained dynamically as described in "Virtual interface IP address assignments" on page 4-47. Other Interface parameters are cloned from the configured main interface. The system sets up direct routes to the virtual interfaces during interface creation, and assigns the type `virtual` to the new interfaces.

The following set of commands configures the system to act as DHCP relay agent, enables option 82, and enables creation of virtual interfaces for BIR connections.

```
admin> read ip-global
admin> set bootp-relay active = yes
admin> set bootp-relay bootp-servers 1 = 192.168.144.1
admin> set bootp-relay relay-agent-info enable = yes
admin> set bootp-relay relay-agent-info allow-multiple-interface = yes
admin> set bootp-relay relay-agent-info circuit-id enable = yes
admin> set bootp-relay relay-agent-info circuit-id if-ip = 192.168.146.200
admin> write -f
```

The next commands enable relay agent requests on the BIR connection. Note that the `remote-address` field must be null. For details about addresses are assigned to the virtual IP interfaces as they are created, see "Virtual interface IP address assignments" on page 4-47.

```
admin> read connection
admin> set station = bir
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 0.0.0.0/0
admin> set ip-options local-address = 192.168.146.200/24
admin> set ip-options bootp-relay-options enable = yes
admin> set ip-options bootp-relay-options max-dynamic-interface = 3
```

```
admin> set atm-options nailed-group = 3
admin> set bir-options enable = yes
admin> set bir-options proxy-arp = yes
admin> write -f
```

Sample configuration using the DHCP router option

This sample configuration is just like the previous one ("Sample configuration with option 82 and multiple interface creation" on page 4-59) except that the IP address (the local address) of the virtual interface is obtained from the router option (option 3). Using the DHCP router option enables a host to use a particular gateway, and so allows the DCHP server to distributes addresses on different subnets for hosts on the same physical connection.

If the information is not provided in the DHCP packet, the system discards the DHCP reply and does not create the virtual interface. If more than one address is specified via the router option in the reply packet, the system uses the first one.

The following set of commands configures the system to act as DHCP relay agent, enables option 82, and enables creation of virtual interfaces for BIR connections.

```
admin> read ip-global
admin> set bootp-relay active = yes
admin> set bootp-relay bootp-servers 1 = 192.168.144.1
admin> set bootp-relay relay-agent-info enable = yes
admin> set bootp-relay relay-agent-info allow-multiple-interface = yes
admin> set bootp-relay relay-agent-info circuit-id enable = yes
admin> set bootp-relay relay-agent-info circuit-id if-ip = 192.168.146.200
admin> write -f
```

The next commands enable relay agent requests on the BIR connection. Note that the `remote-address` field must be null. For details about addresses are assigned to the virtual IP interfaces as they are created, see "Virtual interface IP address assignments" on page 4-47.

```
admin> read connection
admin> set station = bir
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 0.0.0.0/0
admin> set ip-options local-address = 0.0.0.0/0
admin> set ip-options bootp-relay-options enable = yes
admin> set ip-options bootp-relay-options max-dynamic-interface = 3
admin> set atm-options nailed-group = 3
admin> set bir-options enable = yes
admin> write -f
```

Configuring broadband RAS subscriber access

With broadband remote access server (BRAS) support, a Stinger IP2000 can terminate PPP over ATM (PPPoA) and PPP over Ethernet (PPPoE) sessions received on a LIM DSL interface.

When a DSL interface receives PPPoA or PPPoE packet streams, the system switches the inbound traffic to the LIM's ATM internal interface on the basis of an ATM circuit configuration. The IP2000 controller then uses IP routing to forward the packet streams to the appropriate egress interface. For background information about ATM circuit configurations, see the *Stinger ATM Configuration Guide*.

Note PPPoA and PPPoE sessions are supported on DSL interfaces only, but are not currently supported over T1/E1/IMA or IDSL ports. PPPoA and PPPoE are not supported across trunk or Ethernet interfaces.

Recommended `call-type` setting for PPP sessions

The system assigns nailed group 1 to LIM slot 1 port 1 ({ 1 1 1 }), and also assigns nailed group 1 to permanent, leased PPP connections. Because the default `call-type` in Stinger systems is nailed (`ft1`), a conflict occurs when you configure PPPoA or PPPoE connections on any LIM interface, and { 1 1 1 } is enabled.

Following are representative default settings showing the conflicting nailed group assignments:

```
[in DS1-ATM/{ shelf-1 slot-1 1 }:line-config]
nailed-group = 1

[in CONNECTION/"":telco-options]
call-type = ft1
nailed-groups = 1
```

Note Because these defaults conflict, it is recommended to set the `call-type` value to `off` for all PPP connections.

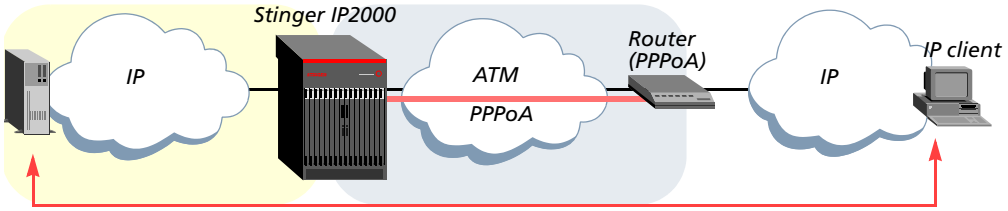
With the default `call-type` setting of `ft1` in a PPP profile, if { 1 1 1 } is enabled (with or without connections), the system generates rolling LOG errors, LOG warning, and LOG information messages. This condition is prevented by setting `call-type` to `off`.

In addition, when `call-type` is set to `off`, the system is able to terminate PPP sessions automatically following a period of client inactivity, and reestablish them when the client becomes active again.

Overview of PPPoA and PPPoE topologies

A PPPoA connection uses the ATM adaptation layer 5 (AAL5) protocol as a framing mechanism across point-to-point virtual circuits, as described in RFC 2364, *PPP over AAL5*. Only VC-multiplexed PPPoA is currently supported. The PPPoA session is between the router and the Stinger unit, as shown in Figure 4-18. The PPPoA session enables an IP client on the far side of the PPPoA router to connect through the Stinger unit to the IP cloud beyond it.

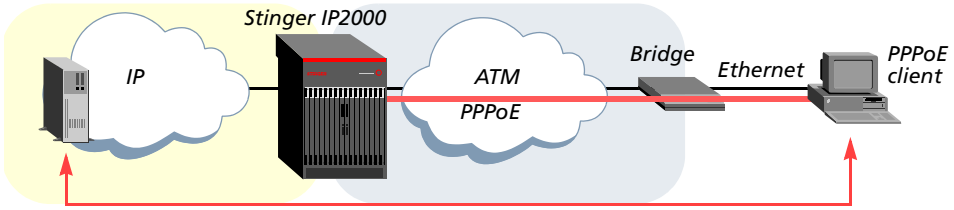
Figure 4-18. PPPoA topology



For PPPoE, the connection uses Ethernet-bridged framing, as defined in RFC 2516, *A Method for Transmitting PPP Over Ethernet (PPPoE)*. PPPoE is negotiated in two phases: a discovery phase and a session phase. During the discovery phase, the PPPoE client negotiates with the Stinger to obtain information it requires. When the session has been established, the client sends PPP packets encapsulated in Ethernet-bridged frames.

The PPPoE sessions shown in Figure 4-19 are between a PPPoE client connected through the bridge, and the Stinger unit. The PPPoE session enables the client to connect through the Stinger unit to the IP cloud beyond it.

Figure 4-19. PPPoE topology



Required setup for PPPoA and PPPoE connections

To enable establishment of PPPoA or PPPoE connections using IP routing, you must complete the following steps:

1 Configure the `answer-defaults` profile to accept PPP session requests and require their authentication. You do this once, to enable the system to process subsequent PPP session requests. See "Configuring the answer-defaults profile for PPP sessions" on page 4-62.

2 Configure an ATM circuit `connection` profile between the DSL interface on which the PPPoE or PPPoA connection will be established and the LIM internal interface. See "Terminating traffic on a LIM internal interface" on page 4-64.

3 Configure a `connection` or RADIUS profile for each PPPoE or PPPoA session, as described in "Example of configuring a PPPoA connection" on page 4-65 or "Example of configuring a PPPoE connection" on page 4-67.

If you are using RADIUS to externally authenticate PPP sessions, see the *TAOS RADIUS Guide and Reference*.

Configuring the `answer-defaults` profile for PPP sessions

To control access, PPP sessions typically require password authentication each time a PPP session is opened. Stinger units support RFC 1334 *PPP Authentication Protocol (PAP)* and RFC 1994 *Challenge Handshake Authentication Protocol (CHAP)*, which must

be negotiated with the client on the basis of settings in the answer-defaults profile and the client's connection or RADIUS profile.

The values in the answer-defaults profile are applied before the system locates the connection or RADIUS profile associated with the request. If the PPP client's profile contains a similar parameter with a different value, the connection-specific value overrides the answer-defaults value when building the session. If no similar value is found in the connection or RADIUS profile, the answer-defaults setting is used.

Following are some answer-defaults parameters, shown with default settings, that affect PPP authentication and session timeout:

```
[in ANSWER-DEFAULTS]
profiles-required = yes

[in ANSWER-DEFAULTS:ppp-answer]
enabled = yes
receive-auth-mode = no-ppp-auth
bi-directional-auth = none
substitute-send-name = ""

[in ANSWER-DEFAULTS:session-info]
idle-timer = 120
max-call-duration = 0
```

Parameter	Setting
profiles-required	A setting of yes (the default) prevents unauthenticated sessions. If set to no, the system builds a temporary profile for session requests for which it cannot locate a configured profile.
enabled	The enabled parameter must be set to yes (the default) for the system to answer PPP session requests.
receive-auth-mode	With the default no-ppp-auth setting, the Stinger unit does not request authentication. If set to a non-default value, the Stinger unit requests an authentication protocol, and the client must accept one of the options the system offers.
bi-directional-auth	Support for bidirectional CHAP. If set to allowed or required, the system negotiates bidirectional CHAP if the client's connection profile specifies the proper settings.
substitute-send-name	System name to send to clients for bidirectional CHAP authentication, if different from the name setting in the system profile.
idle-timer	With a call-type setting of off in a client profile, the system uses the idle-timer value to terminate the session after a default interval of 2 minutes. You can configure a different default interval here or in the client's session-options subprofile, by specifying the maximum number of consecutive seconds a session can remain idle before it is terminated.

Parameter	Setting
max-call-duration	Maximum number of minutes of connect time for a PPP session. The default zero value disables the timer.

Most sites change the default setting of the receive-auth-mode parameter to ensure authentication of a PPP request before a session can be established. For example:

```
admin> read answer-defaults
admin> set ppp-options receive-auth-mode = any-ppp-auth
admin> write -f
```

With this setting, the system accepts session requests that provide any of the supported PPP authentication methods, but it drops requests that do not offer any authentication protocols during session negotiation.

The following commands enable bidirectional authentication for sessions that use CHAP and specify the proper settings in the connection or RADIUS profile:

```
admin> set ppp-answer bidirectional-auth = allowed
admin> write -f
```

With these settings, if a calling device accepts CHAP authentication, the system attempts to negotiate bidirectional CHAP, but does not reject the request if the negotiation fails. However, if bidirectional CHAP is negotiated, authentication must succeed in both directions. For related information, see "Sample PPPoA connection with bidirectional CHAP authentication" on page 4-66.

Terminating traffic on a LIM internal interface

For all installed LIMs that can terminate PPPoA or PPPoE calls, the system creates an atm-internal profile for the LIM's internal ATM segmentation assembly and reassembly (SAR) port. The internal interface number is one greater than the highest DSL interface number on the module. For example, the following command output indicates a 48-port LIM in slot 1, a 72-port LIM in slots 2 and 4, and a 24-port LIM in slot 6:

```
admin> dir atm-internal
42  05/09/2004 08:41:34  { shelf-1 slot-1 49 }  1:1:49
42  05/09/2004 08:41:34  { shelf-1 slot-2 73 }  1:2:73
42  05/09/2004 09:30:38  { shelf-1 slot-4 73 }  1:4:73
42  05/09/2004 08:41:34  { shelf-1 slot-6 25 }  1:6:25
38  05/09/2004 08:41:30  { shelf-1 first-control-module 1 } 1:8:1
```

For a DSL interface to handle PPPoA and PPPoE incoming calls, you must configure an ATM circuit between the external DSL interface and the LIM's internal ATM interface. The ATM circuit configuration must specify the nailed group of the internal interface as the second leg of the circuit (in the atm-connect-options subprofile).

Note The system prevents configuration of a circuit from one LIM to the ATM internal interface of another LIM, or configurations that attempt to use the internal interface of a LIM in other ways.

To determine the nailed group of a LIM's internal ATM interface, use the which command. Following is a sample command for the internal interface of a 72-port LIM in slot 4:

```
admin> which -n { 1 4 73 }
Nailed group corresponding to port { shelf-1 slot-4 73 } is 2271
```

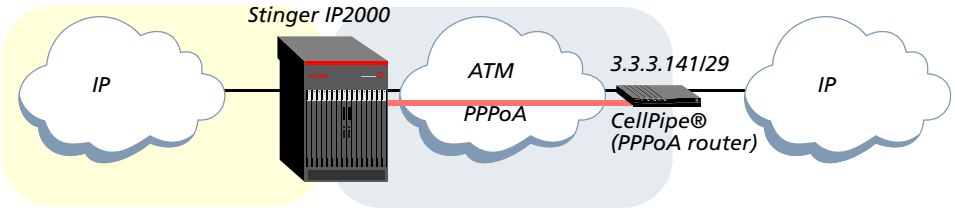
The following commands enable an SDSL interface in slot 4 and configure an ATM circuit from the DSL interface to the LIM's ATM internal interface:

```
admin> read sdsl { 1 4 1 }
admin> set enabled = yes
admin> write -f
admin> new connection cir-1
admin> set active = yes
admin> set encapsulation-protocol = atm-circuit
admin> set atm-options vci = 38
admin> set atm-options nailed-group = 151
admin> set atm-options atm1483type = aal5-vc
admin> set atm-connect-options vci = 36
admin> set atm-connect-options nailed-group = 2271
admin> write -f
```

Example of configuring a PPPoA connection

Figure 4-20 shows a Stinger IP2000. Across a DSL interface, a CellPipe® unit is operating as a PPPoA router.

Figure 4-20. Example of a PPPoA session on a DSL interface



Note You can use bidirectional CHAP authentication for PPPoA connection, but it is not required. The sample configuration provides an example of how to use bidirectional authentication.

Overview of PPPoA `connection` *settings*

For background information about IP routing configurations, see "Configuring IPoA subscriber connections" on page 4-34.In addition to those settings, following are relevant PPPoA parameters, shown with default settings, including bidirectional CHAP authentication:

```
[in CONNECTION/""]
station* = ""
encapsulation-protocol = atm-circuit
[in CONNECTION/"":ip-options]
ip-routing-enabled = yes
remote-address = 0.0.0.0/0
[in CONNECTION/"":ppp-options]
```

```
send-auth-mode = none
bidirectional-auth = none
substitute-recv-name = ""
send-password = ""
recv-password = ""
[in CONNECTION/"":telco-options]
call-type = ft1
```

Parameter	RADIUS attribute	Setting
`station`	User-Name (1)	Name of the PPPoA router. The value is case sensitive and must exactly match the name the router sends during authentication.
`encapsulation-protocol`	Framed-Protocol (7)	Encapsulation protocol. Set to `ppp` for PPPoA clients.
`ip-routing-enabled`	Ascend-Route-IP (228)	Enables or disables IP routing for the interface. IP routing is enabled by default.
`remote-address`	Framed-IP-Address (8) Framed-IP-Netmask (9)	IP address of the PPPoA router, which can include a subnet specification.
`bidirectional-auth`	Ascend-Bi-Directional-Auth (46)	Enables or disables bidirectional CHAP authentication. Used only for bidirectional CHAP.
`send-auth-mode`	Ascend-Send-Auth (231)	Set to `chap-ppp-auth` for bidirectional CHAP. Used only for bidirectional CHAP.
`substitute-recv-name`	Ascend-Recv-Name (45)	Name that must be received from the far end during bidirectional CHAP authentication, if different from the `station` setting. Used only for bidirectional CHAP.
`send-password`	Ascend-Send-Secret (214)	Password the Stinger must send to the far end during bidirectional CHAP, used only for bidirectional CHAP.
`recv-password`	Password (2)	Password the system must receive from the PPPoA router. This setting is used for all PPP authentication methods.
`call-type`	Ascend-Call-Type (177)	Set `call-type` to `off` in PPP profiles. For background information, see "Recommended call-type setting for PPP sessions" on page 4-61.

Sample PPPoA connection with bidirectional CHAP authentication

First, configure an ATM circuit between the DSL interface connecting to the PPPoA router and the LIM's internal interface. See "Terminating traffic on a LIM internal interface" on page 4-64.

Then, verify that the `answer-defaults` profile enables the PPP authentication methods to be used, and configure a `connection` profile to the PPPoA router.

For example, the following commands configure a profile in which bidirectional CHAP authentication is required with the CellPipe® router in Figure 4-20:

```
admin> new connection cellpipe1
admin> set active = yes
```

```
admin> set encapsulation-protocol = ppp
admin> set ip-options remote-address = 3.3.3.141/29
admin> set ppp-options send-auth-mode = chap-ppp-auth
admin> set ppp-options bi-directional-auth = required
admin> set ppp-options send-password = sendpw
admin> set ppp-options recv-password = recvpw
admin> set telco-options call-type = off
admin> write -f
```

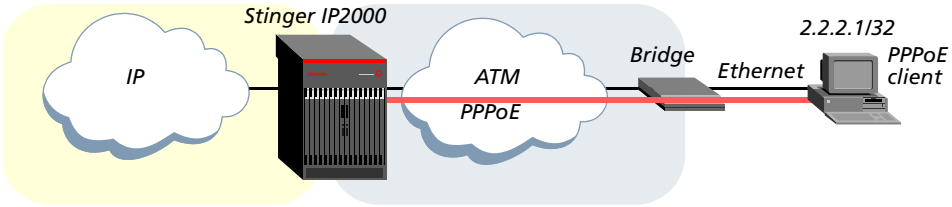
Following is a comparable RADIUS user profile:

```
cellpipe1 Password = "recvpw"
    Service-Type = Framed-User,
    Ascend-Require-Auth = Require-Auth,
    Ascend-Auth-Type = Auth-CHAP,
    Ascend-Send-Auth = Send-Auth-CHAP,
    Ascend-Bi-Directional-Auth=Bi-Directional-Auth-Required,
    Ascend-Send-Secret = "sendpw",
    Ascend-Call-Type = Switched,
    Framed-Protocol = PPP,
    Framed-IP-Address = 3.3.3.141,
    Framed-IP-Netmask = 255.255.255.248,
```

Example of configuring a PPPoE connection

Figure 4-20 shows a Stinger IP2000. Across a DSL interface, a CPE bridging device connects to an Ethernet segment with a PPPoE client.

Figure 4-21. Example of a PPPoE session on a DSL interface



Note You can use bidirectional CHAP authentication for PPPoE connections, but it is not required. For information about using bidirectional CHAP, see "Sample PPPoA connection with bidirectional CHAP authentication" on page 4-66.

Overview of PPPoE `connection` *settings*

For background information about IP routing configurations, see "Configuring IPoA subscriber connections" on page 4-34. In addition to those settings, following are relevant PPPoE parameters, shown with default setting, including PAP authentication:

```
[in CONNECTION/""]
station* = ""
encapsulation-protocol = atm-circuit
[in CONNECTION/"":ip-options]
```

```
ip-routing-enabled = yes
remote-address = 0.0.0.0/0

[in CONNECTION/"":ppp-options]
recv-password = ""

[in CONNECTION/"":pppoe-options]
pppoe = no
bridge-non-pppoe = no

[in CONNECTION/"":telco-options]
call-type = ft1
```

Parameter	RADIUS attribute	Setting
`station`	User-Name (1)	Name of the PPP client system. The value is case sensitive, and must exactly match the name the client presents during authentication.
`encapsulation-protocol`	Framed-Protocol (7)	Encapsulation protocol. Set to `ppp` for PPPoE clients.
`ip-routing-enabled`	Ascend-Route-IP (228)	Enables or disables IP routing for the interface. IP routing is enabled by default.
`remote-address`	Framed-IP-Address (8)	IP address of the remote device, which can include a subnet specification.
	Framed-IP-Netmask (9)	
`recv-password`	Password (2)	Password sent by the PPPoE client.
`pppoe`	Ascend-PPPoE-Enable (74)	Enables or disables processing of PPPoE packets. Must be set to `yes` for a PPPoE connection. If `encapsulation-protocol` is set to `ppp` and `pppoe` is not enabled, the connection is assumed to be PPPoA.
`bridge-non-pppoe`	Ascend-Bridge-Non-PPPoE (75)	*Not currently supported.*
`call-type`	Ascend-Call-Type (177)	Set `call-type` to `off` in PPP profiles. (For RADIUS profiles, set the attribute to `Switched`.) For background information, see "Recommended call-type setting for PPP sessions" on page 4-61.

Sample PPPoE connection using PAP authentication

First, configure an ATM circuit between the DSL interface connecting to the CPE bridging device and the LIM's internal interface. See "Terminating traffic on a LIM internal interface" on page 4-64.

Then, verify that the `answer-defaults` profile enables the PPP authentication methods to be used and configure a `connection` profile to the PPPoE client.

For example, the following commands configure a profile using PPP authentication for the PPPoE client in Figure 4-21:

```
admin> new connection pppoe-1

admin> set active = yes
```

```
admin> set encapsulation-protocol = ppp
admin> set ip-options remote-address = 2.2.2.1/29
admin> set ppp-options recv-password = pppoe1!pw
admin> set pppoe-options pppoe = yes
admin> set telco-options call-type = off
admin> write -f
```

Following is a comparable RADIUS user profile:

```
pppoe-1 Password = "pppoe1!pw"
    Service-Type = Framed-User,
    Framed-Protocol = PPP,
    Framed-IP-Address = 2.2.2.1,
    Framed-IP-Netmask = 255.255.255.248,
    Ascend-Call-Type = Switched,
    Ascend-PPPoE-Enable = PPPoE-Yes,
    Ascend-Call-Type = 0
```

Optional configuration of a LIM ATM internal interface

By default, the ATM internal interface of a LIM is enabled and configured with system-generated values for required settings, so no additional configuration is required. Following are the parameters, shown with default values, for configuring internal ATM interfaces:

```
[in ATM-INTERNAL/{ any-shelf any-slot 0 }]
name = ""
physical-address* = { any-shelf any-slot 0 }
enabled = yes
line-config = { 1 15 }
traffic-shapers = [ { no 1000 1000 2 no 1 } { no 1000 1000 2 no 2 } { no 100+
[in ATM-INTERNAL/{ any-shelf any-slot 0 }:line-config]
nailed-group = 1
vp-switching-vpi = 15
```

Parameter	Setting
name	Assigns a name to the interface, up to 15 characters. The name is used only for administrative purposes.
physical-address	Physical address of the internal SAR port within the system. This value is set by the system when it creates the atm-internal profile, and it is used to retrieve the ATM configuration for the interface.
enabled	Enables or disables the interface for use. The interface is enabled by default.

Parameter	Setting
`line-config:nailed-group`	A system-generated unique number that represents the interface in the system. You specify this number in a `connection` or RADIUS profile when creating the ATM circuit for terminating PPPoA or PPPoE connections. **Note** Lucent Technologies does not recommend modifying the system-generated `nailed-group` number assigned by default to the internal interface.
`line-config: vp-switching-vpi`	VPI to use for VP switching. For details, see the *Stinger ATM Configuration Guide*.
`traffic-shapers`	*This setting is not currently supported for terminating connections on DSL LIMs (PPPoE/PPPoA).*

Administrative tools for IP routing

The system supports several commands that are useful for locating the sources of problems on an IP network and for communicating with other hosts for management purposes. For examples, see the chapter on working with IP traffic in the *Stinger Adminstration Guide*, and entries in the *Stinger Reference* on commands such as the following:

- `arptable`
- `ipcache`
- `iproute`
- `netstat`
- `nslookup`
- `ping`
- `telnet`
- `traceroute`

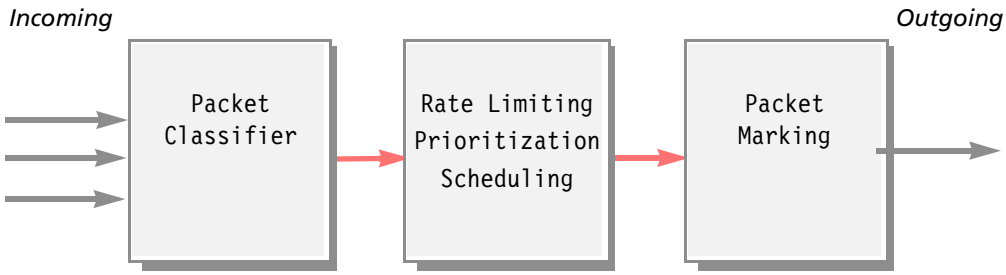
Ethernet and IP QoS

5

Overview of the QoS implementation

To provide quality of service (QoS), the system classifies an incoming packet flow according to configured rules, rate-limits the traffic, assigns packets a priority, and finally marks the outgoing traffic. Figure 5-1 shows the order in which these processes occur.

Figure 5-1. QoS subsystems



This scheme provides QoS in one direction of traffic flow only. Rate limiting, scheduling, and packet marking occur after the packets have been routed and classified, and are always applied at the output side of the outgoing interface.

IP QoS applies to all terminated routed connections, as well as to routed VLAN interfaces. IP QoS supports layer 3 packet classifiers, priority scheduling with rate limiting, and IP ToS marking. For routed VLAN interfaces, both IP ToS and Ethernet priority marking are supported.

Ethernet QoS applies to all bridged connections as well as to routed VLAN interfaces. Ethernet QoS supports both layer 2 and layer 3 packet classifiers, strict priority scheduling, and Ethernet priority marking.

Packet classification subsystem

The classification subsystem selects packets based on the content of packet headers according to defined rules. Rate limiting, priority scheduling, and marking can then be performed on selected packets.

A classification rule set can be created with only one rule or with a combination of rules. When a combination of rules is used, the subsystem performs a logical AND between the rules. For example, if the rule set is composed of a source IP address and its netmask and the IP protocol, only packets that match both the specified source address and the specified protocol are selected. In addition, when multiple rules are specified, the rules are applied in the order in which they appear in the profile. The first rule is applied first. If it does not match, then the next one is applied, and so on.

Caution Ethernet fields in a `layer2-classifier` subprofile and IP-specific fields in a `packet-classifier` subprofile *must not* be configured together in the same `packet-flows` profile. Configuring both classifications in the same profile yields unpredictable results. Note that the system does not prevent you from such a misconfiguration.

Interface grouping provides a simple way to classify packet flows by input interface. You configure input interfaces with an interface group number, and classify traffic based on that group number alone, or that number in combination with other rules. Interface grouping is configured in the `packet-classifier` subprofile but it is not IP-specific. It can be combined with layer 2 classification in the same `packet-flows` profile.

Rate-limiting, prioritization, and scheduling subsystem

Once the classification has been done, the classification result is passed to the scheduling subsystem, which is responsible for selecting a queue according to the priorities, the rate-limiting result, and scheduling at the connection level.

For information about connection-level scheduling, see "ATM QoS and IP QoS considerations" on page 5-21 and "Configuring Ethernet egress scheduling and shaping" on page 5-21.

Note Ethernet QoS supports priority scheduling. IP QoS supports both priority scheduling and rate limiting.

For individual packet flows, up to eight flow priorities (from 0 to 7) can be defined per connection in a `packet-flows` profile. Each priority level has a queue. The scheduler selects the proper output queue according to the priority configured for the packet flow and the rate limiting associated with the traffic class (for IP traffic), and the bandwidth of the connection.

The scheduler attached to the connection selects a priority queue as follows:

1 Select the highest priority queue.

2 If the selected queue has no traffic, select the next highest priority queue.

3 If the selected queue has no traffic, select the next highest priority queue, and so forth.

With rate-limiting, some packets are assigned lower priority queues than the configured priority when bursty conditions prevail. The rate limiting subsystem implements an RFC 2697 Single Rate Three Color Marker to meter an IP packet flow and color-mark its packets either green, yellow, or red. Marking is based on bandwidth allowed to the packet flow, and how far (or how long) the traffic stream exceeds the allowed bandwidth.

The policing algorithm (one rate, three colors) defined by the RFC 2697 is fully supported by the Stinger. Once a flow has been marked green, yellow or red, it is possible to either drop certain colors of packets (such as those marked red) or to reprioritize the flow with different, decreasing levels of priority for yellow or red packets. For example, the system might discard all red packets, forward yellow packets as best effort, and forward green packets with a high priority.

The rate-limiting subsystem always operates in color-blind mode. That is, the metering process assumes that the packet stream it receives has not already been color marked.

Packet marking engine

The packet marking engine marks outgoing packets to receive a particular service level along their path. Table 5-1 shows the types of packet marking supported on various types of interfaces. The *Profile* column in Table 5-1 indicates the profile in which the `packet-flows` profile is applied.

Table 5-1. Packet marking supported on egress interfaces

Profile	Type of egress interface	Marking action
`ip-interface`	Routed Ethernet	IP ToS
`vlan-ethernet`	Routed VLAN	IP ToS or Ethernet p-bit
`vlan-ethernet`	Bridged VLAN	Ethernet p-bit
`connection`	Routed WAN	IP ToS
`connection`	Pretagged stacked VLAN	Ethernet p-bit
`connection`	Bridged VLAN (other than pretagged)[a]	No marking support

a. Pretagged stacked VLAN connections are bridged, but because these connections contain a VLAN tag when received from the subscriber side, they can support p-bit marking in the downstream direction.

Note Only one type of marking can be performed for all packet flows defined in a given `packet-flows` profile. Both the fields cannot be marked simultaneously.

The system does not support classification of routed packets on the basis of layer 2 parameters, or marking of IP ToS values for bridged packets.

Configuration steps

The configuration of QoS for a particular output interface should be done in the following order:

1 Define the packet flows that must be discriminated for the output interface, and specify the scheduling priority for each flow. This step requires the creation of one `packet-flows` profile.

2 If needed, group input interfaces and specify the interface group number in the `packet-flows` profile.

3 Attach the `packet-flows` profile to the output interface. You do this in the `ip-interface`, `vlan-ethernet`, or `connection` profile for the interface.

What the system does at the output interface

Once the output interface is up, the system performs following steps:

1 Retrieve the `packet-flows` profile attached to the interface

2 Perform checks to ensure that the configuration is valid.

3 Create the number of queues needed according to the number of priorities defined by the `packet-flows` profile.

4 For ATM connections, select the correct scheduler according to the connection-level ATM service contract configured in the `atm-qos` profile. (See "ATM QoS and IP QoS considerations" on page 5-21.)

5 For Ethernet connections, select the correct scheduler according to the connection-level `ip-interface scheduling` priority configuration. (See "Configuring Ethernet egress scheduling and shaping" on page 5-21.)

6 Create the classification rules and apply them to the output interface.

Because classifiers must be communicated to the underlying system hardware, which can be time intensive when many rules have been defined, a method is provided for batching updates system wide. For details, see "Performance recommendations" on page 5-22.

Default IP QoS configuration

When no `packet-flows` profile has been attached to an interface, the following default IP QoS is applied:

- Control traffic goes in the highest priority queue.
- If multicast is enabled on the interface, multicast traffic goes in a high priority queue.
- Unicast traffic goes in a low priority queue.

When a `packet-flows` profile has been attached to the interface but does not contain an explicit default rule, the system creates one default rule for which the scheduling priority will be the lowest priority.

Introduction to `packet-flows` profile settings

A `packet-flows` profile describes a set of up to 32 packet flows and how each flow should be scheduled and marked. Following are the parameters and subprofiles, shown with default values, at the top level of the profile:

```
[in PACKET-FLOWS/""]
name* = ""
marking-type = none
flow = [ { { 0 0 00:00 } { 0 0.0.0.0/0 0.0.0.0/0 0 0 none 0 none 0 } { 0 } } +
global-packet-marking = [ { 0 0 0 } { 1 0 0 } { 2 0 0 } { 3 0 0 } { 4 0 0 } {+
exception-packet-marking = { 8 0 0 }
```

Parameter or subprofile	Purpose
`name`	Name of the profile, up to 31 characters, used to uniquely identify the packet flow set within the Stinger system.
`marking-type`	Specify whether to mark an IP ToS value (layer 3) or VLAN p-bit value (layer 2) in the packet headers.
`flow`	An array of 32 indexed subprofiles for configuring classification rules and a scheduling priority for up to 32 different flows. Each indexed subprofile contains the following subprofiles:
	`layer2-classifier` — Classify an Ethernet bridged traffic flow. A given `packet-flows` profile can configure either `layer2-classifier` settings or IP-specific `packet-classifier` settings, but not both.
	`packet-classifier` — Classify an IP packet flow. The packets can be routed through the system or bridged Ethernet traffic. A given `packet-flows` profile can configure either `layer2-classifier` settings or IP-specific `packet-classifier` settings, but not both.
	`scheduling` — Specify a priority for the traffic class. For VLAN, strict priority queueing is supported. IP traffic can be rate limited as well.
`global-packet-marking`	Specifies the IP ToS or VLAN p-bit values to be marked in the outgoing traffic.
`exception-packet-marking`	Specifies the IP ToS or VLAN p-bit values for outgoing packets generated by the Stinger system, such as ARP or ICMP packets.

Layer 2 classifiers

Layer 2 classification uses Ethernet header fields to select frames for strict priority scheduling, in both upstream and downstream directions.

Note The system does not support classification of routed packets on the basis of layer 2 parameters, or the classification of packet flows using a mix of layer 2 and IP-specific fields.

Following are the parameters, shown with default values, for classifying traffic at layer 2. For each parameter, the default zero or null setting matches *any* value in the corresponding fields of the packet.

```
[in PACKET-FLOWS/"":flow[1]:layer2-classifier]
vlan-id = 0
ethernet-priority = 0
ethernet-type = 00:00
```

Parameter	Description
`vlan-id`	The 802.1Q/P standard specifies a tag that appends to a MAC frame. The VLAN tag carries both the VLAN ID (12-bit) and Prioritization (3-bit). This parameter can specify a VLAN ID (a number from 0 to 4095). In the case of stacked VLANs, the user-level `vlan-id` is used instead of the NSP-level `vlan-id`. See "Overview of VLAN stacking settings for tagged frames" on page 3-16.
`ethernet-priority`	802.1Q/P priority value (p-bit value) of the frame's VLAN tag. A number from 0 to 7. In the case of stacked VLANs, the user-level `ethernet-priority` is used instead of the NSP-level `ethernet-priority`. See "Overview of VLAN stacking settings for tagged frames" on page 3-16.
`ethernet-type`	Ethernet type. A two-byte hexadecimal number representing the Ethernet protocol. For example, PPPoE values are `8863` (PPPoE Discovery Stage packets), and `8864` (PPP Session Stage packets).

For an example of layer 2 classification, see "Configuring bridging VLAN Ethernet QoS" on page 5-29.

Packet classifiers

Packet classification uses an interface group number and/or IP packet header fields to select packets for rate-limiting, prioritization, and scheduling, in both upstream and downstream directions. The IP packets can be routed or bridged. Following are the parameters, shown with default values, for classifying traffic at IP packet layers. For each parameter, the default zero or null setting matches *any* value in the corresponding fields of the packet.

```
[in PACKET-FLOWS/"":flow[1]:packet-classifier]
interface-group = 0
source-ip-address = 0.0.0.0/0
```

```
source-ip-netmask = 0.0.0.0
destination-ip-address = 0.0.0.0/0
destination-ip-netmask = 0.0.0.0
ip-protocol = 0
ip-tos = 0
source-port-comparison = none
source-port = 0
destination-port-comparison = none
destination-port = 0
```

Parameter	Description
interface-group	Interface group number (from 0 to 255) used for classification. Value 0 matches all interfaces, so it means no classification will be performed based on interface group. Up to 256 groups can be created. Interface grouping is configured in the packet-classifier subprofile but it is not IP-specific. It can be combined with layer 2 classification in the same packet-flows profile.
source-ip-address	This parameter and the optional source-ip-netmask are used for a classification based on the source IP address of the incoming packets. For example, 192.207.23.13/29.
source-ip-netmask	Network mask to apply before the source IP address comparison. For example, 255.255.255.248.
destination-ip-address	This parameter and the optional destination-ip-netmask are used for a classification based on the destination IP address of the incoming packets. For example, 192.207.23.13/29.
destintation-ip-netmask	Network mask to apply before the destination IP address comparison. For example, 255.255.255.248.
ip-protocol	Numeric value (from 0 to 255), used for a classification based on the IP protocol. For a list of assigned protocol numbers, see RFC 1700, *Assigned Numbers*.
ip-tos	Numeric value (from 0 to 255),used for a classification based on the IP TOS bits (8 bits) present in the incoming packets.
source-port-comparison	Type of comparison to be done on the source UDP/TCP port for a classification based on source port. Enumerated field, values: none — Packet address/port is not used in classifying this flow. less — Packet address/port less than rule's address/port. equal — Packet address/port same as rule's address/port.

Parameter	Description	
	`greater`	Packet address/port greater than rule's address/port.
`source-port`	Numeric value (from 0 to 65535) used for the source UDP/TCP port comparison defined by source-port-comparison.	
`destination-port-comparison`	Type of comparison to be done on the destination UDP/TCP port for a classification based on destination port. Enumerated field, values:	
	`none`	Packet address/port is not used in classifying this flow.
	`less`	Packet address/port less than rule's address/port.
	`equal`	Packet address/port same as rule's address/port.
	`greater`	Packet address/port greater than rule's address/port.
`destination-port`	Numeric value (from 0 to 65535) used for the destination UDP/TCP port comparison defined by destination-port-comparison.	

Caveat about fragmented IP packets

Keep in mind that for fragmented IP packets, only the IP packet header can be used for classification, which means that classification based on UDP or TCP source and destination ports cannot be used.

Details of packet classifier comparison passes

In evaluating incoming traffic destined for an particular output interface, the system begins by comparing the packet header to rule #1. If that comparison does not result in a match, it proceeds to compare the packet to rule #2, and so forth. The comparison process stops immediately if conditions in a packet match a classification rule. At that point, the scheduling and marking in the profile are applied to the packet flow. Within each rule, the comparisons proceed as shown in Table 5-2:

Table 5-2. Comparison passes performed on inbound packet flows

Comparison pass	Description
1. Interface group	If the group number assigned to the incoming interface does not match the `interface-group` of the rule, the comparison fails. See "Prioritizing IP packet flows based on DSL service contracts" on page 5-23 for an example.
2. Source address/netmask	If the source address of the packet does not match the `source-address/source-netmask` of the rule, the comparison fails.

Table 5-2. Comparison passes performed on inbound packet flows (Continued)

Comparison pass	Description
3. Destination address/ netmask	If the destination address of the packet does not match the `destination-address/destination-netmask` of the rule, the comparison fails.
4. IP protocol	If the protocol field of the packet does not match the `ip-protocol` of the rule, the rule comparison fails.
5. IP TOS bits	If the 8 TOS bits in the incoming packet header do not match the rule's `ip-tos`, the comparison fails.
6. UDP/TCP source port	If the source address of the packet does not compare as specified to the `source-port` of the rule, the comparison fails.
7. UDP/TCP destination port	If the destination address of the packet does not compare as specified to the `destination-port` of the rule, the comparison fails.

For each comparison, the default zero or null setting for an identifier matches all values in a packet. If all comparisons succeed, the packet matches the rule. If any one of the comparisons fail, the packet does not match the rule.

Comparisons of IP addresses

When you specify a source or destination address in a rule, the system assigns a priority to packets received from or sent to that address. If you also specify a subnet mask, the system applies the mask to the address value before comparing the resulting value to the source or destination address in a packet.

To apply the mask, the system translates both the mask and address values into binary format and then uses a logical AND to apply the mask to the address. The mask hides the bits whose positions match those of the binary zeros in the mask. A mask of all zeros (the default) masks all bits. If the address value itself is also all zeros (the default), the rule matches any source or destination address. A mask of all ones (255.255.255.255) masks no bits, so the full address for a single host is compared to the address value.

You can use the address mask to mask out the host portion of an address or the host and subnet portion, so the specification matches the address to or from any host on a given network. For example, the following sample commands specify that any packet received from subnet 192.168.2.0 must be processed at priority level 3:

```
admin> read packet-flows src-netmask-demo
admin> set flow 1 packet-classifier source-ip-netmask = 255.255.255.0
admin> set flow 1 packet-classifier source-ip-address = 192.168.2.2
admin> set flow 1 scheduling queue-priority = 3
admin> write -f
```

Comparisons of IP TOS values

A rule specifying an `ip-tos` value allows for prioritization of traffic based on the value of the IP TOS byte in the IPv4 header. The default zero value matches all values in the TOS byte. A nonzero value must be matched exactly. For example, the following sample commands specify that packets with a TOS value of 255 must be processed at priority level 5:

```
admin> read packet-flows tos-demo
admin> set flow 1 packet-classifier ip-tos = 255
admin> set flow 1 scheduling queue-priority = 5
admin> write -f
```

Comparisons of port numbers

Rules can specify a port number to be compared to the source or destination port (or both) in a packet.

Note When a nonzero `source-port` or `destination-port` is specified in a `packet-flows` profile, the `ip-protocol` value must be set to either 17 (UDP) or 6 (TCP) and the `source-port-comparison` or `destination-port-comparison` parameter must specify the type of comparison to be performed.

TCP and UDP port numbers are typically assigned to services. For a list of well-known port assignments, see RFC 1700, *Assigned Numbers*.

The `source-port-comparison` or `destination-port-comparison` setting determines when a match occurs. For all traffic, the following settings can be used to compare source port or destination port values:

- `none` (no comparison is made)
- `equal` (equal to)
- `less` (less than)
- `greater` (greater than)

The following commands specify that packets received on any port other than port 69 (which is used for TFTP) should be assigned priority level 2:

```
admin> read packet-flows src-port-demo
admin> set flow 1 packet-classifier ip-protocol = 17
admin> set flow 1 packet-classifier source-port-comparison = less
admin> set flow 1 packet-classifier source-port = 69
admin> set flow 2 packet-classifier ip-protocol = 17
admin> set flow 2 packet-classifier source-port-comparison = greater
admin> set flow 2 packet-classifier source-port = 69
admin> set flow 1 scheduling queue-priority = 2
admin> set flow 2 scheduling queue-priority = 2
admin> write -f
```

How nonmatching packets are prioritized (the default rule)

The default rule, which defines how non-matching packets are prioritized, has following characteristics:

- The default rule must be the last `flow` configuration in the `packet-flows` profile.
- All `packet-classifier` fields and `layer2-classifier` of the default rule must be `0` or `none` (the default values).
- The `scheduling queue-priority` for the default rule must be set to the desired priority setting for nonmatching packets.

If you do not define a scheduling priority for the default rule, the system uses priority zero for nonmatching packets. If a rule specifies more than one default rule, only the first one is taken into account. If a `packet-flows` profile specifies 32 classifications and does not include a default rule, the system creates an implicit default rule.

For example, the following commands specify priority 1 for packets that do not match the configured rule for flow 1:

```
admin> read packet-flows src-netmask-demo
admin> set flow 1 packet-classifier source-ip-netmask = 255.255.255.0
admin> set flow 1 packet-classifier source-ip-address = 192.168.2.2
admin> set flow 1 scheduling queue-priority = 3
admin> set flow 2 scheduling queue-priority = 1
admin> write -f
```

Scheduling and rate limiting

In this section, the following terms are used:

Table 5-3. Rate limiting terminology

Term	Definition
CIR	Committed Information Rate
CBS	Committed Burst Size
EBS	Excess Burst Size
Green	Traffic color returned by the policer when the traffic does not exceed CBS.
Yellow	Traffic color returned by the policer when the traffic exceeds CBS but does not exceed EBS.
Red	Traffic color returned by the policer when the traffic exceeds CBS and EBS.

Following are the parameters, shown with default settings, for configuring scheduling and rate limiting:

```
[in PACKET-FLOWS/"":flow[1]:scheduling]
queue-priority = 0
yellow-queue-priority = -1
red-queue-priority = -1
committed-information-rate = 0
```

```
committed-burst-size = 0
excess-burst-size = 0
```

Parameter	Setting
queue-priority	Queue priority of the green traffic (from 0 to 7). The default value of 0 means the lowest priority queue. For scheduling Ethernet traffic, this setting represents a strict priority for the bridged traffic flow.
yellow-queue-priority	Priority (from -1 to 7) of the flow when the policer is active and detects traffic above CBS but below EBS. The default value of -1 indicates that yellow traffic must be discarded. In this condition, the red-priority queue is not used and must be also configured with the value -1.
red-queue-priority	Priority (from -1 to 7) of the flow when the policer is active and detects traffic above EBS. The default value of -1 indicates that red traffic must be discarded.
committed-information-rate	Average bandwidth of the flow in Kbps, from 0 to 1,000,000. With the default value of 0, the policer is disabled and the flow is not rate limited. This is the maximum possible setting for CIR. The flow rate in this case is the rate defined by the atm-qos profile for ATM connections, or by the scheduling subprofile for Gigabit Ethernet interfaces. A nonzero value automatically enables the policer in color-blind mode. The minimum policing rate (minimum CIR) 16.276 Kbps rounded to 17 Kbps. The precision of the policing algorithm is a function of the CIR, the configured peak rate of the connection or interface, and the actual line rate. The worst precision occurs when the desired peak rate is close to the line rate.
committed-burst-size	CBS expressed in bytes. This value (from 0 to 65,535) is used to size the CBS token bucket. The minimum recommended value when the policer is enabled is two times the MTU of the connection.
excess-burst-size	EBS expressed in bytes. This value (from 0 to 65,535) is used to size the EBS token bucket. The minimum recommended value when the policer is enabled is three times the MTU of the connection.

Token buckets in the single-rate three color policing algorithm

A "token bucket" is a metaphor for a calculated transfer rate used internally by the policer to regulate a packet flow. In the metaphor, the system places tokens into a bucket at a certain rate (the CIR). The size of a token is the IP packet size, as defined by RFC 2697. If the transfer rate is always at the CIR level, tokens pass through the buckets without accumulating.

When a source sends bursty traffic, the system discards packets if tokens are not available in either of the commit and excess buckets. The largest burst a source can send into the network is generally proportional to the size of the bucket.

It is strongly recommended to configure the size of the buckets to be able to store at least two packets. That is the reason why the minimum recommended value for CBS is two times the MTU of the connection. Because the EBS bucket must be larger than the CBS bucket, the minimum recommended value for EBS is three times the MTU (to store three packets).

Using a single rate two-color algorithm

To configure the rate-limiting subsystem to use only two colors (green and red), set the `excess-burst-size` parameter to 0 and the `yellow-queue-priority` parameter to -1 (not used). You can configure the other parameters in the `scheduling` subprofile to any desired values.

Notes on the policing implementation

- The policer acts as a real policer, so it has no associated queues. This means that rate-limiting a bursty flow implies a correct setting of the burst-size parameters.
- Since a policer is in use, multicast flows cannot be rate-limited. If the multicast group was limited, in fact all member of the group will be rate-limited.
- The minimum policing rate is 17Kbps regardless of the type of interface.
- The policing precision is about 1% in the worst case (when the peak rate is close to the line rate).
- The Stinger supports up to 32K policing flows.
- Some reserves have to be done regarding the throughput of the network processor due to the policer use.
- For fragmented packets, the policer might not be applicable if the flow to be policed includes the IP protocol (TCP/UDP) and associated port number.

Example of rate limiting on a BIR connection

This sample configuration creates a `packet-flows` profile that rate limits and prioritizes a mix of unicast and multicast traffic flows, and applies it to a bridged IP routing (BIR) connection.

1 Configure the `packet-flows` profile to assign a high priority (7) to multicast traffic, middle priority (5) to ICMP packets, and low priority (3) to IP unicast traffic. Unicast traffic and ICMP traffic is rate limited.

```
admin> new packet-flows bir-flow
admin> set flow 1 packet-classifier destination-ip-address = 224.1.1.0
admin> set flow 1 packet-classifier destination-ip-net = 255.255.255.0
admin> set flow 1 scheduling queue-priority = 7
admin> set flow 2 packet-classifier ip-protocol = 1
admin> set flow 2 scheduling queue-priority = 5
admin> set flow 2 scheduling yellow-queue-priority = 4
admin> set flow 2 scheduling committed-information-rate = 100
admin> set flow 2 scheduling committed-burst-size = 3000
```

```
admin> set flow 2 scheduling excess-burst-size = 30000
admin> set flow 3 packet-classifier ip-protocol = 4
admin> set flow 3 scheduling queue-priority = 3
admin> set flow 3 scheduling committed-information-rate = 200
admin> set flow 3 scheduling committed-burst-size = 3000
admin> write -f
```

2 Configure the BIR connection profile and apply the packet-flows profile.

```
admin> new connection bir-user
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 192.168.190.2/24
admin> set ip-options local-address = 192.168.190.1/24
admin> set ip-options multicast-allowed = yes
admin> set ip-options multicast-max-groups = 3
admin> set ip-options packet-flows = bir-flow
admin> set bir-options enable = yes
admin> write -f
```

Packet marking

For routed traffic, the IPv4 packet header ToS field can be set to pass priority information on to the next hop. This is useful for IP traffic from a subscriber that is routed to an upstream router.

For bridged and routed VLAN traffic, the 802.1Q VLAN tag can carry both the VLAN ID and a priority value (called a *p-bit value*). The system can set a p-bit value to carry priority information to other p-bit aware devices along the traffic's path.

QoS packet marking for routed traffic

For traffic that is routed through the system, IP ToS marking is supported in both upstream and downstream directions. Values can range from 0 to 255 decimal. The Stinger supports both classification of packets based on this value in incoming packets, and marking of this field in outgoing packets. It does not interpret the value contained in the ToS field, it simply uses it in the configured way.Other network elements, located upstream or downstream, are expected to interpret the values as required, as defined for ToS or DS (for example, as defined in RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*),

Ethernet p-bit marking for bridged or routed VLAN traffic

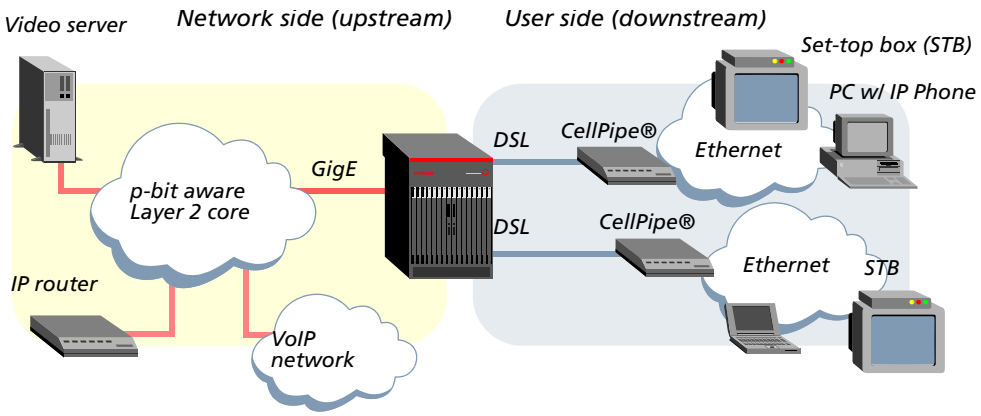
For traffic that is bridged or routed through the system onto a VLAN in the upstream direction, the Stinger snoops the incoming frames and applies the classification rules to determine the p-bit value to be marked in the VLAN tag.

For routed VLANs, p-bit marking is supported in the upstream direction.

For bridged traffic in the downstream direction, only pretagged stacked VLAN configurations receive VLAN-tagged traffic, so p-bit marking in the downstream direction is restricted to pretagged stacked VLANs.

In Figure 5-2, all CellPipes are operating in LLC-bridged mode. QoS differentiation in the Layer 2 core network is based on priority-bits in the Ethernet header.

Figure 5-2. Sample deployment with VLAN p-bit marking



When transparent bridging is in use, all frames that are flooded are marked with the p-bit value corresponding to the least priority configured (or applicable) for the VLAN interface, and the frames are scheduled on the least priority queue. The same applies to frames with the broadcast MAC destination address. However, if IGMP snooping is not enabled, frames with the multicast MAC destination address are scheduled using configured queues.

If IGMP snooping is enabled, scheduling priority and p-bit marking for multicast frames can be configured at the IP multicast address level in the packet-classifier subprofile. Note that a finer granularity of classification (using other fields of the IP header) is not supported for multicast packets.

Overview of packet marking settings

Following are the relevant parameters, shown with default settings:

```
[in PACKET-FLOWS/""]
marking-type = none
global-packet-marking = [ { 0 0 0 } { 1 0 0 } { 2 0 0 } { 3 0 0 } { 4 0 0 } {+
exception-packet-marking = { 8 0 0 }
```

Parameter or subprofile	Purpose
`marking-type`	Specify whether to mark an IP ToS value (layer 3) or VLAN p-bit value (layer 2) in the packet headers. This setting affects all packet marking for this `packet-flows` profile. Only one type of packet marking (IP ToS or VLAN p-bit) can be configured per profile. With the default `none` setting, no packet marking is performed. With any other value, you must configure the related field in the `global-packet-marking` and `exception-packet-marking` subprofiles. Otherwise, the default marking value of zero is applied. `none` — No packet marking is performed.

Parameter or subprofile	Purpose	
	`ip-tos`	Perform IP ToS marking at the output interface.
	`ethernet-priority`	Perform Ethernet priority marking at the output interface.
`global-packet-marking`	An array of 8 indexed subprofiles, one for each scheduling priority in the system. Each of the subprofiles can be used to specify the IP ToS or VLAN p-bit values to be marked in outgoing traffic of the specified priority. Each subprofile contains the following settings:	
	`scheduling-priority`	A read-only value indicating the scheduler priority queue. Packets in this priority queue will be marked with the specified ToS or p-bit value.
	`ip-tos`	IP ToS value (from 0 to 255) to be set in the ToS field of the IPv4 header. The `marking-type` parameter must be `ip-tos` for this value to be marked in outbound packets.
	`ethernet-priority`	802.1P priority value (from 0 to 7) to be set in the VLAN tag of the Ethernet frame. The `marking-type` parameter must be `ethernet-priority` for this value to be marked in outbound packets.
`exception-packet-marking`	A subprofile for specifying the IP ToS or VLAN p-bit values for outgoing packets generated by the Stinger system, such as ARP or ICMP packets.	
	`scheduling-priority`	A read-only value indicating the priority queue. Exception packets are always sent on the highest priority queue (8).
	`ip-tos`	IP ToS value (from 0 to 255) to be set in the ToS field of the IPv4 header. The `marking-type` parameter must be `ip-tos` for this value to be marked in outbound packets.
	`ethernet-priority`	Currently, for `vlan-circuit` and `vlan-stacked` bridged connections, the Stinger does not generate any traffic.

Example of IP ToS marking on a routed VLAN interface

A routed VLAN interface is the router interface or subinterface to which the router's IP address on the VLAN is attached. In a Stinger system, a routed VLAN interface must be mapped to a virtual IP interface. Packets received on a routed VLAN interface are routed based on the IP address, and packets are sent through the routed VLAN interface based on an IP routing decision.

The following steps configure IP ToS marking and priority scheduling on a routed VLAN interface. In this example, the destination IP address is used for packet classification. Each priority class is marked with a specific value in IP ToS field.

1 Create a vlan-ethernet profile. This example uses VLAN ID 572.

```
admin> new vlan-ethernet { { 1 8 2 } 572}
admin> set enabled = yes
admin> set bridging-options bridge-type = no-bridging
admin> write -f
```

2 Create an ip-interface profile for the routed VLAN.

```
admin> new ip-interface { { 1 8 2 } 1 }
admin> set ip-address = 10.10.10.10/24
admin> set vlan-enabled = yes
admin> set vlan-id = 572
admin> set packet-flows = routed-flow
admin> write -f
```

3 Create a packet-flows profile for the routed VLAN.

```
admin> new packet-flows routed-flow
admin> set marking-type = ip-tos
admin> set flow 1 packet-classifier destination-ip-address = 20.0.0.0
admin> set flow 1 packet-classifier destination-ip-netmask = 255.0.0.0
admin> set flow 1 scheduling queue-priority = 3
admin> set flow 2 packet-classifier destination-ip-address = 50.0.0.0
admin> set flow 2 packet-classifier destination-ip-netmask = 255.0.0.0
admin> set flow 2 scheduling queue-priority = 2
admin> set flow 3 packet-classifier destination-ip-address = 70.0.0.0
admin> set flow 3 packet-classifier destination-ip-netmask = 255.0.0.0
admin> set flow 3 scheduling queue-priority = 1
admin> set global-packet-marking 2 ip-tos = 1
admin> set global-packet-marking 3 ip-tos = 2
admin> set global-packet-marking 4 ip-tos = 3
admin> write -f
```

Example of Ethernet p-bit marking

The following steps configure Ethernet p-bit marking and priority scheduling on a bridged VLAN output interface. In this example, the destination IP address is used for packet classification. Each priority class is marked with a specific value in IP ToS field.

1 Create a vlan-ethernet profile. This example uses VLAN ID 624.

```
admin> new vlan-ethernet { { 1 8 2 } 624 }
admin> set enabled = yes
admin> set bridging-group = 7
admin> set bridge = yes
admin> set bridge-type = transparent-bridging
admin> set packet-flows = bridged-flow
admin> write -f
```

2 Create a packet-flows profile for the bridged VLAN.

```
admin> new packet-flows bridged-flow
admin> set marking-type = ethernet-priority
admin> set flow 1 packet-classifier destination-ip-address = 20.0.0.0
admin> set flow 1 packet-classifier destination-ip-netmask = 255.0.0.0
admin> set flow 1 scheduling queue-priority = 3
admin> set flow 2 packet-classifier destination-ip-address = 50.0.0.0
admin> set flow 2 packet-classifier destination-ip-netmask = 255.0.0.0
admin> set flow 2 scheduling queue-priority = 2
admin> set flow 3 packet-classifier destination-ip-address = 70.0.0.0
admin> set flow 3 packet-classifier destination-ip-netmask = 255.0.0.0
admin> set flow 3 scheduling queue-priority = 1
admin> set global-packet-marking 2 ethernet-priority = 1
admin> set global-packet-marking 3 ethernet-priority = 2
admin> set global-packet-marking 4 ethernet-priority = 3
admin> write -f
```

Example of mapping ATM QoS to a packet marking value

The following steps show how to use interface grouping to mark packets based on connections' atm-qos service category. The intent is to assign the same interface group number to connections with the same atm-qos service category. For example, CBR connections are placed in interface group 70, VBR connections in interface group 50, and UBR connections in interface group 10. You can then use the interface group to assign marking values. For example, all CBR traffic to be marked with ip-tos = 7, VBR traffic with ip-tos = 5 and UBR with ip-tos = 0.

1 Modify connection profiles to assign interface group numbers based on the connections' service category. For example:

```
admin> read connection cbr-user1
admin> set ip-options qos-interface-group = 70
```

```
admin> write -f
admin> read connection vbr-user1
admin> set ip-options qos-interface-group = 50
admin> write -f
admin> read connection ubr-user1
admin> set ip-options qos-interface-group = 10
admin> write -f
```

2 Create a `packet-flows` profile that maps the interface groups to appropriate `ip-tos` marking values.

```
admin> new packet-flows qos-tos-marking
admin> set marking-type = ip-tos
admin> set flow 1 packet-classifier interface-group = 70
admin> set flow 1 scheduling priority = 7
admin> set flow 2 packet-classifier interface-group = 50
admin> set flow 2 scheduling priority = 5
admin> set flow 3 packet-classifier interface-group = 10
admin> set flow 3 scheduling priority = 1
admin> set global-packet-marking 8 ip-tos = 7
admin> set global-packet-marking 6 ip-tos = 5
admin> set global-packet-marking 2 ip-tos = 0
admin> write -f
```

QoS-related connection and interface settings

The following `connection` and `ip-interface` parameters, shown with default values, are related to QoS handling:

```
[in CONNECTION/"":ip-options]
qos-interface-group = 0
packet-flows = ""
[in CONNECTION/"":bridging-options]
qos-interface-group = 0
packet-flows = ""
[in CONNECTION/"":atm-qos-options]
usr-up-stream-contract = default
usr-dn-stream-contract = default
[in VLAN-ETHERNET/{ { shelf-1 first-control-module 2 } n }:bridging-options ]
qos-interface-group = 0
packet-flows = ""
[in IP-INTERFACE/{ { shelf-1 first-control-module 2 } 0 }]
qos-interface-group = 0
packet-flows = ""
[in IP-INTERFACE/{ { shelf-1 first-control-module 2 } 0 }:scheduling]
priority = 0
bandwidth = 0
```

Parameter	Setting
`qos-interface-group`	Numeric field, from 0 to 255, used to group input interfaces for QoS handling. The assigned value can be used as a classifier. For an example configuration using interface grouping, see "Prioritizing IP packet flows based on DSL service contracts" on page 5-23. For routed subscriber interfaces, set the group number in the `connection ip-options` subprofile. For bridged subscriber interfaces, set the number in the `connection bridging-options` subprofile. For routed VLANs, the `qos-interface-group` is set in the `ip-interface` profile and in case of bridged VLANs, it is set in the `vlan-ethernet` profile.
`packet-flows`	Name of the `packet-flows` profile to be attached to the output interface. For routed subscriber output interfaces, attach the `packet-flows` profile in the `connection ip-options` subprofile. For bridged subscriber output interfaces, attach the profile in the `connection bridging-options` subprofile. For routed VLANs, the `packet-flows` profile can be set in the `ip-interface` profile. For bridged VLANs, it is set in the `vlan-ethernet` profile.
`usr-up-stream-contract` `usr-dn-stream-contract`	Names of the `atm-qos` profiles to be applied to upstream traffic and downstream traffic, respectively. See "ATM QoS and IP QoS considerations" on page 5-21.
`priority`	Scheduler priority of the virtual IP interface, from 0 to 3. The default value of 0 indicates the lowest priority.
`bandwidth`	Scheduling rate of the virtual IP interface, expressed in Kbps, from 0 to 1,000,000. The default value of 0 indicates full bandwidth. To configure a nonzero number, the `priority` setting must also be nonzero. For IP2000 versions A and B with a revision level of 25 or less, the minimum scheduling rate is 801Kbps. For IP2000 version B with a revision level of 26 and higher, the minimum scheduling rate is 401Kbps. The system ensures that these minimums are respected.

Applying a `packet-flows` profile to an output interface

The following commands apply a configured `packet-flows` profile named `demo-priority-flow` on the Gigabit Ethernet interface:

```
admin> read ip-interface { { 1 8 2 } 0 }
admin> set packet-flows = demo-priority-flow
admin> write -f
```

Inheritance of `packet-flows` configurations on virtual IP interfaces

Both WAN and LAN interfaces can have virtual IP interfaces, although only LAN interfaces support static configuration of a virtual interface.

Dynamic virtual interfaces, such as those created on a WAN interface when relaying DHCP requests, always inherit the same `packet-flow` configuration and interface group as the main IP interface on the port.

Packet flow processing occurs at the output side of an interface, and is applicable to all statically configured virtual interfaces. A virtual interface and the main interface on the same port do not have to use the same `packet-flows` profile.

Virtual IP interfaces and interface grouping

When a port has been assigned a `qos-interface-group` number, the system must be able to determine which virtual IP interface on the port is receiving a packet flow, because interface grouping occurs at the input side of the interface. Because it is not ensured that the system can determine the correct virtual interface carrying a packet flow, a virtual IP interface must use the same interface group as the main IP interface on the port. If a different interface group is specified in the virtual interface `ip-interface` profile, the system logs a warning message applies the group number of the main IP interface.

ATM QoS and IP QoS considerations

The `atm-qos` and `packet-flows` profiles can be used in conjunction to configure connection bandwidth and the per-flow bandwidth. The `atm-qos` profile definitions are used in the downstream direction to specify the ATM QoS contract (CBR, nrtVBR, rtVBR, or UBR), which sets the priority level of the connection, with CBR at the highest priority and UBR at the lowest. This connection-level prioritization is particularly useful when several connections share the same physical link. The `atm-qos` profile traffic parameters (PCR for CBR connections, and SCR for VBR connections) are used to shape the downstream traffic at the specified rates. In the upstream direction, ATM policing performed on the traffic received on the connection.

Using an unlimited rate contract for an ATM connection can have an impact regarding priorities defined by the `packet-flows` profile. For example, suppose a `connection` profile uses the default ATM QoS, which applies a UBR contract, and applies a `packet-flows` profile that defines three flows mapped to three different priorities, with flow 1 at the highest priority and a throughput of 2Mbps, flow 2 at medium priority and a throughput of 4Mbps, and flow 3 at the lowest priority and a throughput of 5Mbps. Suppose the connection has a line rate of 6Mbps. Because there is no traffic shaper associated with the connection, the network processor sends the full 11Mbps to the DSL line, causing discards at the LIM side regardless of the traffic priority.

Configuring Ethernet egress scheduling and shaping

The `scheduling` subprofile of the `ip-interface` profile enables you to apply a traffic shaper and a priority for a logical interface mapped on the Gigabit Ethernet interface. The `scheduling` configuration is applicable to all kinds of routed connections, including routed VLANs.

Note The minimum scheduling rate for the Gigabit interface is 401Kbps.

The following commands specify middle priority (rather than the default low priority) for traffic on the IP interface of the Gigabit Ethernet port:

```
admin> read ip-interface { { 1 8 2 } 0 }
admin> set scheduling priority = 2
admin> write -f
```

QoS-related settings in the system profile

Settings in the system profile affect call management systemwide, and are documented in the *Stinger Adminstration Guide*. The next sections provide overviews of settings that can affect performance when QoS is applied.

Performance recommendations

When packet-flows or filter profiles have been applied to output interfaces, the Stinger system software must communicate resulting changes in traffic processing requirements to the underlying system hardware. These updates to the hardware are typically time intensive, so creating a batch of these requests is recommended to improve the overall performance of the system (system performance) and to control the rate at which connections or service is restored to end-users (call performance).

The following parameters, shown with default values, provide performance enhancements for interactions with the network processor:

```
[in SYSTEM]
np-update-time = 0
np-default-filtering-policy = drop
np-fpp-compact-timer = 3600
```

Setting np-update-time to between 10-20 seconds is strongly recommended, particularly when a large number of filter and/or classification rules are configured on the system. For details about np-update-time and the other network processor performance options, see the *Stinger Adminstration Guide*.

Configurable queue size for DSL links

For a system heavily loaded with subscriber uplinks to the Gigabit Ethernet interface, the system can begin to discard traffic sent to the LIM, based on scheduler memory and queue size. For DSL links, the default queue is 200 blocks (with 256 bytes per block). You can increase the queue size for all LIM interfaces system wide by setting the following parameter, shown with its default value:

```
[in SYSTEM]
scheduler-queue-size = 200
```

For trunk connections and Gigabit Ethernet links, the queue size is not configurable.

Examples of configuring QoS

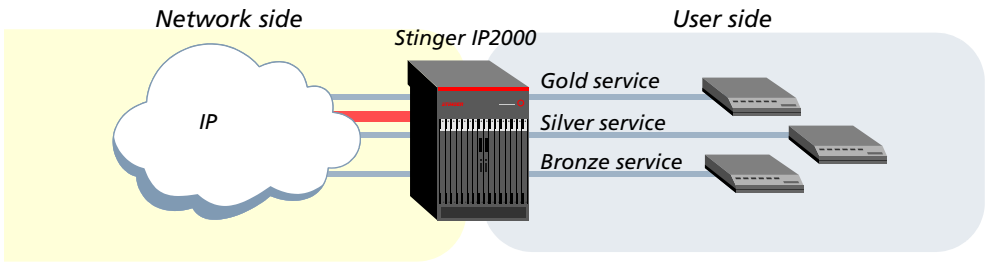
This section contains the following configuration examples:

- Prioritizing traffic based on the level of end-user service contract
- Classifying several types of IP traffic for prioritization on one ATM PVC
- Using packet flow prioritization together with ATM QoS to prioritize bandwidth use for different types of IP traffic

Prioritizing IP packet flows based on DSL service contracts

In this example, the provider configures processing priorities according to the price of the end-user DSL contract. For a low-price service contract (bronze), packet flows are assigned a low processing priority, for a medium-price contract (silver), packet flows will be processed at medium priority, and for a high-price contract (gold), packet flows will have high-priority processing. The `packet-flows` profile is applied to the Ethernet output interface.

Figure 5-3. Using interface grouping to prioritize traffic by service level



The result of this configuration is that the traffic coming from a user with a gold service contract is output on the Ethernet interface from the high-priority queue. Once the high-priority queue is empty, traffic from a user with a silver service contract is output from the medium-priority queue. When the high-priority and medium-priority queues are empty, traffic from a user with a bronze contract is transmitted on the output interface.

This sample configuration requires the following steps:

1 Configure subscriber profiles with the appropriate `qos-interface-group` value. In this example, bronze customers are assigned to group 1, silver customers are assigned to group 2 and gold customers are assigned to interface group 3.

```
admin> read connection user-gold-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 192.168.100.1/32
admin> set ip-options local-address = 192.168.100.200/32
admin> set ip-options qos-interface-group = 3
admin> set atm-options nailed-group = 21
admin> write -f
admin> new connection user-silver-1
admin> set active = yes
```

```
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 192.168.100.10/32
admin> set ip-options local-address = 192.168.100.201/32
admin> set ip-options qos-interface-group = 2
admin> set atm-options nailed-group = 2
admin> write -f
admin> new connection user-bronze-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 192.168.100.20/32
admin> set ip-options local-address = 192.168.100.202/32
admin> set ip-options qos-interface-group = 1
admin> set atm-options nailed-group = 11
admin> write -f
```

2 Configure the `packet-flows` profile to classify and prioritize traffic by interface group. Traffic from bronze customers will have priority 1, from silver customers will have priority 3, and from gold customers will have priority 5.

```
admin> new packet-flows service-contracts
admin> set flow 1 packet-classifier interface-group = 1
admin> set flow 1 scheduling queue-priority = 1
admin> set flow 2 packet-classifier interface-group = 2
admin> set flow 2 scheduling queue-priority = 3
admin> set flow 3 packet-classifier interface-group = 3
admin> set flow 3 scheduling queue-priority = 5
admin> write -f
```

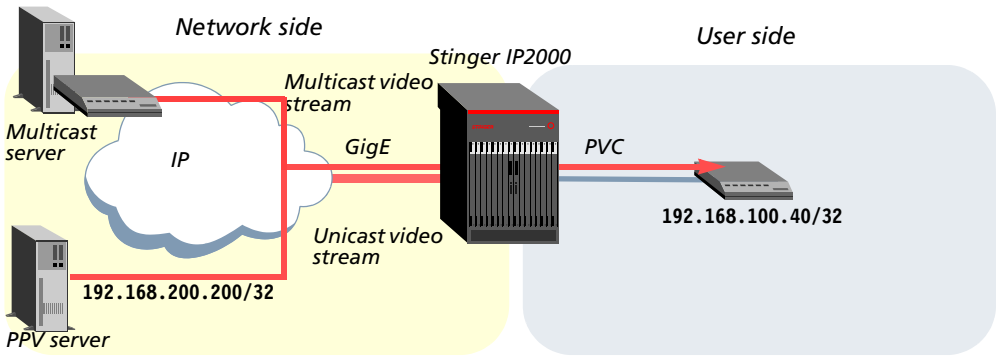
3 Apply the `packet-flows` profile to the output interface (Gigabit Ethernet).

```
admin> read ip-interface { { 1 8 2 } 0 }
admin> set ip-address = 10.60.1.1/8
admin> set netmask = 255.0.0.0
admin> set packet-flows = service-contracts
admin> write -f
```

Prioritizing different kinds of IP traffic on an ATM PVC

In this example, the provider needs to classify different kinds of IP traffic across a single ATM PVC. The `packet-flows` profile will be applied to the DSL connection as the output interface.

Figure 5-4. Unicast and multicast video share the same priority



The provider classifies IP traffic types as shown in Table 5-4. All control traffic such as IGMP, RIP, and so forth, is assigned the highest priority and has precedence over the scheduled QoS priorities.

Table 5-4. Sample IP traffic types and priorities across an ATM PVC

Kind of IP traffic	Priority	Classification
Multicast video	Medium (6) video	Multicast destination address
Unicast video pay per view (PPV)	Medium (6)	PPV server source IP address
Data traffic	Low (5)	Default rule for all other traffic

Note Because multicast video and the unicast PPV have the same queue-priority value, they will share the same queue.

This sample configuration requires the following basic steps:

1 Configure the `packet-flows` profile to classify and prioritize traffic by type. Multicast video and PPV are assigned medium priority, and data traffic is assigned low priority (by defining a default rule). PPV traffic is classified by the source IP address of the PPV server (192.168.200.200).

```
admin> new packet-flows
admin> set name = mixed-traffic
admin> set flow 1 packet-classifier destination-ip-address = 224.0.0.0
admin> set flow 1 packet-classifier destination-ip-netmask = 240.0.0.0
admin> set flow 1 scheduling queue-priority = 6
admin> set flow 2 packet-classifier source-ip-address = 192.168.200.200
admin> set flow 2 packet-classifier source-ip-netmask = 255.255.255.255
admin> set flow 2 scheduling queue-priority = 6
admin> set flow 3 scheduling queue-priority = 5
admin> write -f
```

2 Apply the `packet-flows` profile to the output interface (the DSL side).

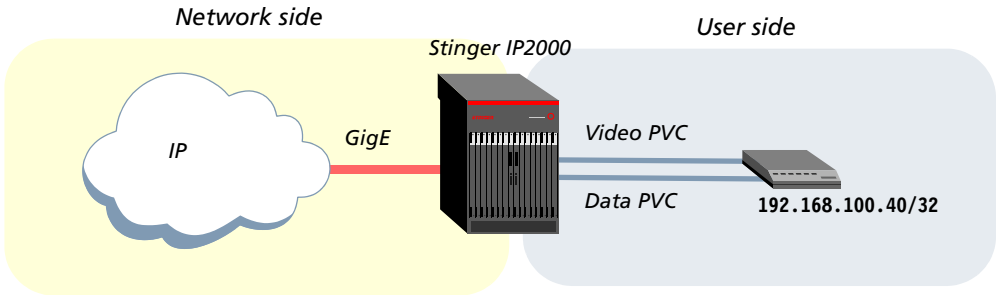
```
admin> new connection user-1
admin> set active = yes
```

```
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 192.168.100.40/32
admin> set ip-options local-address = 192.168.100.204/32
admin> set ip-options packet-flows = mixed-traffic
admin> set atm-options nailed-group = 4
admin> write -f
```

Prioritizing traffic using both IP and ATM QoS

In this example, the provider needs to classify different kind of IP traffic across two ATM PVCs on the same DSL interface. The `packet-flows` profile is applied to the DSL connection as the output interface, and ATM QoS is applied to each ATM PVC separately.

Figure 5-5. Prioritizing traffic at the connection level and flow level



In this example, two `packet-flows` profiles are configured, because the two ATM PVCs have very different classification requirements, as shown in Table 5-5:

Table 5-5. Sample IP traffic types and priorities across two PVCs

ATM PVC	Kind of IP traffic	Priority	Classification
`video-pvc`	Signaling	High (7)	All other traffic
	Multicast video	Medium (6)	Multicast destination address
	Unicast video/PPV	Medium (6)	PPV server source IP address
	Miniguide	Low (5)	TCP protocol (for HTTP)
`data-pvc`	Web TV	High (7)	UDP protocol (for RTP)
	Internet traffic	Low (6)	Default rule for all other traffic

Caution The setup shown in Table 5-5 is just an example. Because it grants priority 7 to a classification for "all other traffic" (the default rule), it could cause the system to grant high priority to unauthorized traffic such as packets received during a denial-of-service attack or a Blaster worm attack. This type of security issue must be considered before configuring a packet flow with a high-priority default rule.

This sample configuration requires the following basic steps:

1 Configure a `packet-flows` profile for `video-pvc`. The classification for packet flow 1 defines multicast video at medium priority. Packet flow 2 specifies unicast PPV at medium priority, on the basis of the source IP address of the PPV server (192.168.10.1).

Packet flow 3 classifies HTTP traffic on the basis of the TCP protocol. A null classification (a default rule) for flow 4, specifies that all other IP traffic (such as signaling traffic) carried on the PVC should be assigned the highest priority.

```
admin> new packet-flows
admin> set name = video-traffic
admin> set flow 1 packet-classifier destination-ip-address = 224.0.0.0/8
admin> set flow 1 packet-classifier destination-ip-netmask = 255.0.0.0
admin> set flow 1 scheduling queue-priority = 6
admin> set flow 2 packet-classifier source-ip-address = 192.168.10.1/32
admin> set flow 2 packet-classifier source-ip-netmask = 255.255.255.255
admin> set flow 2 scheduling queue-priority = 6
admin> set flow 3 packet-classifier ip-protocol = 6
admin> set flow 3 scheduling queue-priority = 5
admin> set flow 4 scheduling queue-priority = 7
admin> write -f
```

2 Configure a `packet-flows` profile for `data-pvc`, by classifying packet flow 1 for WebTV at high priority, and then defining a default rule for flow 2 with a priority of 6 for all other IP traffic carried on the PVC:

```
admin> new packet-flows
admin> set name = internet-traffic
admin> set flow 1 packet-classifier ip-protocol = 17
admin> set flow 1 scheduling queue-priority = 7
admin> set flow 2 scheduling queue-priority = 6
admin> write -f
```

3 Configure three `atm-qos` profiles, one for downstream traffic for each PVC, and one to be used by both PVCs for upstream traffic.

In this example, the DSL line supporting the two PVCs provides 6Mbps downstream and 256Kbps upstream bandwidth. The provider has to limit the bandwidth accessible to the two interfaces, as shown in the following table:

ATM PVC	Bandwidth	Service classification
`video-pvc`	5.5Mbps downstream	rtVBR at 5500Kbps
	128k upstream	nrtVBR at 128Kbps
`data-pvc`	320Kbps downstream	rtVBR at 320Kbps
	128k upstream	nrtVBR at 128Kbps

The next set of commands defines an `atm-qos` profile for `data-pvc` downstream traffic:

```
admin> new atm-qos
admin> set contract-name = internet-downstream
admin> set traffic-descriptor-type = noclp-scr-cdvt
admin> set atm-service-category = real-time-vbr
admin> set peak-rate-kbits-per-sec = 320
admin> set peak-cell-rate-cells-per-sec = 754
admin> set sustainable-rate-kbits-per-sec = 320
admin> set sustainable-cell-rate-cells-per-sec = 754
admin> set ignore-max-burst-size = no
admin> set max-burst-size = 1
admin> set aal-type = aal-5
admin> set early-packet-discard = yes
admin> write -f
```

The next set of commands defines an `atm-qos` profile for `video-pvc` downstream traffic:

```
admin> new atm-qos
admin> set contract-name = video-downstream
admin> set traffic-descriptor-type = noclp-scr-cdvt
admin> set atm-service-category = real-time-vbr
admin> set peak-rate-kbits-per-sec = 5500
admin> set peak-cell-rate-cells-per-sec = 12971
admin> set sustainable-rate-kbits-per-sec = 5500
admin> set sustainable-cell-rate-cells-per-sec = 12971
admin> set ignore-max-burst-size = no
admin> set max-burst-size = 1
admin> set aal-type = aal-5
admin> set early-packet-discard = yes
admin> write -f
```

Because both PVCs support the same amount of upstream bandwidth, they can both use the same `atm-qos` profile. The next set of commands defines an `atm-qos` profile for `video-pvc` and `data-pvc` upstream traffic:

```
admin> new atm-qos
admin> set contract-name = upstream
admin> set traffic-descriptor-type = noclp-scr-cdvt
admin> set atm-service-category = real-time-vbr
admin> set peak-rate-kbits-per-sec = 128
admin> set peak-cell-rate-cells-per-sec = 301
admin> set sustainable-rate-kbits-per-sec = 128
```

```
admin> set sustainable-cell-rate-cells-per-sec = 301
admin> set ignore-max-burst-size = no
admin> set max-burst-size = 1
admin> set aal-type = aal-5
admin> set early-packet-discard = yes
admin> write -f
```

4 Apply the `packet-flows` and `atm-qos` profiles to the output interfaces (the subscriber ATM PVCs in this example).

The following commands configure `video-pvc`:

```
admin> new connection
admin> set station = user-1-video-pvc
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 192.168.100.100/32
admin> set ip-options local-address = 192.168.100.210/32
admin> set ip-options packet-flows = video-traffic
admin> set atm-options nailed-group = 6
admin> set atm-qos-options usr-up-stream-contract = upstream
admin> set atm-qos-options usr-dn-stream-contract = video-downstream
admin> write -f
```

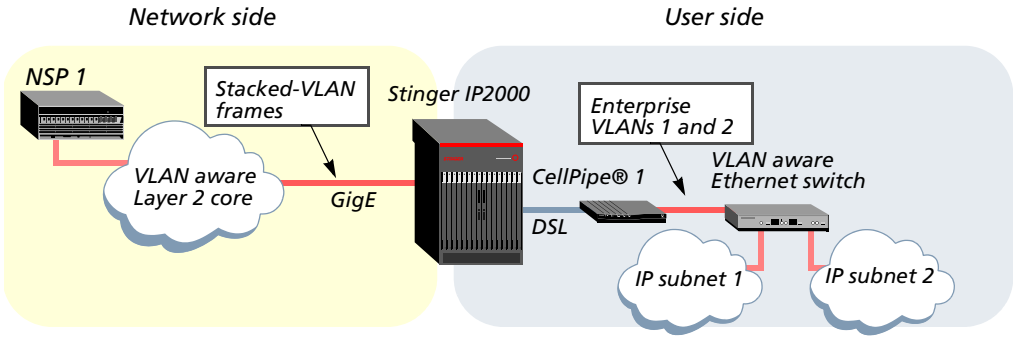
The next commands configure `data-pvc`:

```
admin> new connection
admin> set station = user-1-data-pvc
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 192.168.100.101/32
admin> set ip-options local-address = 192.168.100.211/32
admin> set ip-options packet-flows = internet-traffic
admin> set atm-options vci = 36
admin> set atm-options nailed-group = 6
admin> set atm-qos-options usr-up-stream-contract = upstream
admin> set atm-qos-options usr-dn-stream-contract = internet-downstream
admin> write -f
```

Configuring bridging VLAN Ethernet QoS

In this example, the Stinger system receives pre-tagged VLAN traffic from the CellPipe®. The incoming VLAN traffic is p-bit marked, and the system must mark the p-bit again before sending the stacked frames out to NSP 1. This sample setup is shown in Figure 5-6.

Figure 5-6. Stacked VLAN requiring p-bit remarking on the output interface



The incoming p-bit values from the CellPipe® and the p-bit values in the output frames to NSP VLAN 1 are shown in Table 5-6.

Table 5-6. Ethernet p-bit remarking table

p-bit value in input frame	p-bit value in output frame
1	1
2	3
3	5

To configure the p-bit priority mappings, follow these steps:

1 Make sure bridging is enabled in the `ethernet` profile. (If the layer-2 core network is using a specific EtherType value for stacked VLAN frames, you must also set the `vlan-stack-tag-type` parameter to that value.)

```
admin> read ethernet { 1 8 2 }
admin> set bridging-enabled = yes
admin> write -f
```

2 Create a `packet-flows` profile that classifies the traffic by incoming p-bit value and reassigns a new p-bit value, as shown in Table 5-6.

```
admin> new packet-flows
admin> set name = pbitremark
admin> set marking-type = ethernet-priority
admin> set flow 1 layer2-classifier ethernet-priority = 1
admin> set flow 1 scheduling queue-priority = 1
admin> set flow 2 layer2-classifier ethernet-priority = 2
admin> set flow 2 scheduling queue-priority = 2
admin> set flow 3 layer2-classifier ethernet-priority = 3
admin> set flow 3 scheduling queue-priority = 3
admin> set global-packet-marking 2 ethernet-priority = 1
admin> set global-packet-marking 3 ethernet-priority = 3
admin> set global-packet-marking 4 ethernet-priority = 5
admin> write -f
```

3 Create the NSP VLAN and apply the `packet-flows` profile. In this example, the VLAN is assigned VLAN ID 572 and `bridging-group` 300.

```
admin> new vlan-ethernet
admin> set interface-address physical-address shelf = 1
admin> set interface-address physical-address slot = 8
admin> set interface-address physical-address item-number = 2
admin> set interface-address logical-item = 572
admin> set enabled = yes
admin> set bridging-options bridging-group = 300
admin> set bridging-options bridge = yes
admin> set bridging-options bridge-type = stacked-vlan
admin> set bridging-options packet-flows = pbitremark
admin> write -f
```

4 Configure the `flow-services` mapping between the incoming (enterprise) VLAN tags on the DSL interface and user ID values that are unique within the NSP VLAN. Then create a `connection` profile for the inbound CellPipe® connection with the same `bridging-group` number as the NSP VLAN, and apply the `flow-services` profile. For details about configuring stacked VLAN, see "Configuring stacked VLANs" on page 3-13.

Administrative tools for monitoring IP QoS

The `stats` command, described in detail in the *Stinger Reference*, provides options for gathering and displaying traffic statistics related to QoS. This section describes only the commans related to monitoring IP QoS.

Following is the syntax of the `stats` command in the context of IP QoS monitoring:

`stats` *`cmd`* **`np ipqos`** *`ifnum`* [**`up`** | **`down`**]

The *`cmd`* argument specifies one of the following actions to be performed on the specified interface (*`ifnum`*):

- **`enable`** (Begin monitoring QoS on the specified interface.)
- **`disable`** (Stop monitoring QoS on the specified interface.)
- **`traffic`** (Display traffic statistics on the specified interface.)
- **`rate`** (Display traffic statistics each second. Entering the same command again causes the frequent updates to stop.)
- **`error`** (Display error statistics on the specified interface.)
- **`clear`**(Clear statistics on the specified interface.)

The *`ifnum`* argument must be obtained via the `ifmgr -d` command, which is available only in the debug environment. For details about `ifmgr -d`, see "ifmgr" on page A-15.

The [**`up`** | **`down`**] argument applies only to `vlan-circuit` or `stacked-vlan` bridged connections. For these connections the*`ifnum`* argument is always that of the DSL WAN interface number, and the `up` or `down` argument is used to enable QoS monitoring upstream (on the VLAN interface), or downstream (on the WAN interface).

For example, the following command starts QoS statistics monitoring on interface 15:

```
admin> stats enable np ipqos 15
```

and the following command stops monitoring on that interface:

```
admin> stats disable np ipqos 15
```

Example of monitoring routed traffic onto Gigabit Ethernet

This example shows IP QoS monitoring on the main Gigabit Ethernet interface as routed traffic is forwarded onto Ethernet.

Creating a terminating routed connection

The following commands configure a routed connection:

```
admin> new connection routed-test
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 4.4.4.2/8
admin> set ip-options local-address = 4.4.4.1/8
admin> set atm-options vci = 70
admin> set atm-options nailed-group = 101
admin> write -f
```

Applying a **packet-flows** profile to the Ethernet IP interface

The following commands configure a `packet-flows` profile and apply it to the `ip-interface` profile:

```
admin> new packet-flows
admin> set name = ipflows
admin> set flow 1 packet-classifier source-ip-address = 9.0.0.0/8
admin> set flow 1 packet-classifier source-ip-netmask = 255.0.0.0
admin> set flow 1 scheduling queue-priority = 6
admin> set flow 2 packet-classifier source-ip-address = 8.0.0.0/8
admin> set flow 2 packet-classifier source-ip-netmask = 255.0.0.0
admin> set flow 2 scheduling queue-priority = 5
admin> set flow 3 packet-classifier source-ip-address = 7.0.0.0/8
admin> set flow 3 packet-classifier source-ip-netmask = 255.0.0.0
admin> set flow 3 scheduling queue-priority = 4
admin> write -f
admin> read ip-interface { { 1 8 2 } 0 }
admin> set ip-address = 7.7.7.1/8
admin> set packet-flows = ipflows
admin> write -f
```

Obtaining the Ethernet interface number

The `ifmgr -d` command, available in the debug environment, obtains the Gigabit Ethernet interface number (interface 1 in the output below). For details, see Appendix A, "IP2000 Diagnostics."

```
admin> ifmgr -d
bif slot sif u m p ifname     host-name  remote-addr       local-addr
-------------------------------------------------------------------------
000 1:08 000 *     ie0        -          0.0.0.0/32        110.110.110.72/32
001 1:08 001 *     ie1        -          0.0.0.0/32        7.7.7.1/32
006 1:08 006 *     lo0        -          0.0.0.0/32        127.0.0.1/32
007 0:00 000 *     rj0        -          0.0.0.0/32        127.0.0.2/32
008 0:00 000 *     bh0        -          0.0.0.0/32        127.0.0.3/32
009 1:08 000 *     wanabe     -          0.0.0.0/32        127.0.0.3/32
010 0:00 000 *     local      -          0.0.0.0/32        127.0.0.1/32
011 0:00 000 *     mcast      -          0.0.0.0/32        224.0.0.0/32
012 0:00 000 -     tunnel0    -          0.0.0.0/32        127.0.0.5/32
013 0:00 000 *     vr0_main   -          0.0.0.0/32        127.0.0.4/32
014 0:00 000 -     sip0       -          0.0.0.0/32        0.0.0.0/32
017 1:03 002 *   p wan17      x          4.4.4.2/32        4.4.4.1/32
<end>
```

Eanbling monitoring on the Ethernet IP interface

The following command shows that the CoS-specific QoS counters are incremented on interface 1 (the Gigabit Ethernet port):

```
admin> stats enable np ipqos 1
IP QOS monitoring start for IF 1
IP QOS profile: pbitmark
CLASSIFICATION & FLOW
  Default classification      :provided
  Number of classified DID    :0
  Number of class rules       :5
  Number of multicast IP flow :0
  Number of unicast IP flow   :0
PRIORITIES
  Number of priority          :4
  Lowest priority             :0
  Mapping priority/COS        :P[0]=5 P[1]=x P[2]=x P[3]=x P[4]=4 P[5]=3
P[6]=2 P[7]=x
SCHEDULING
  Line Rate                   :0
  Data queue Id               :256
  Scheduler ID                :3
  PDU type                    :packet
  Number of COS queues        :5
  Scheduling priority         :0
  Peak Rate (Kbps)            :0
  TS parameters              :00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  TM parameters              :06 0f 7d 00 05 dc 00 00 00 00 00 00 00 00 00 00
admin> stats traffic np ipqos 1
FPP counters for IF 1
```

```
Priority     COS     Counter
  drp        -         0
  exp        1         0
  6          2         179
  5          3         0
  4          4         0
  0          5         0
TM counters for IF 1
Priority     COS      MTU drop     Queue drop    Sched drop    Port drop
exp          1        0            0             0             0
6            2        0            0             0             0
5            3        0            0             0             0
4            4        0            0             0             0
0            5        0            0             0             0
```

Example of monitoring bridged VLAN traffic (transparent bridging)

This example shows IP QoS monitoring on both a DSL WAN interface and a VLAN interface, for an N:1 bridged VLAN configuration.

Creating a bridged VLAN interface and **packet-flows** profile

The following commands create a bridged VLAN interface and a `packet-flows` profile to monitor its traffic:

```
admin> new bridge-group
admin> set enable = yes
admin> set bridging-group = 572
admin> set port-block-enabled = no
admin> write -f
admin> new vlan-ethernet { { 1 8 2 } 572 }
admin> set enabled = yes
admin> set bridging-options bridging-group = 572
admin> set bridging-options bridge = yes
admin> set bridging-options bridge-type = transparent-bridging
admin> set bridging-options packet-flows = pbitmark
admin> write -f
admin> new packet-flows pbitmark
admin> set marking-type = ethernet-priority
admin> set flow 1 packet-classifier source-ip-address = 9.0.0.0/8
admin> set flow 1 packet-classifier source-ip-netmask = 255.0.0.0
admin> set flow 1 scheduling queue-priority = 6
admin> set flow 2 packet-classifier source-ip-address = 8.0.0.0/8
admin> set flow 2 packet-classifier source-ip-netmask = 255.0.0.0
admin> set flow 2 scheduling queue-priority = 5
admin> set flow 3 packet-classifier source-ip-address = 7.0.0.0/8
admin> set flow 3 packet-classifier source-ip-netmask = 255.0.0.0
```

```
admin> set flow 3 scheduling queue-priority = 4
admin> set global-packet-marking 5 ethernet-priority = 5
admin> set global-packet-marking 6 ethernet-priority = 6
admin> set global-packet-marking 7 ethernet-priority = 7
admin> write -f
```

Creating a bridged subscriber interface and **packet-flows** profile

The following commands configure a bridged VLAN subscriber interface and a `packet-flows` profile for its traffic flow:

```
admin> new connection bridged-vlan-test
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options ip-routing-enabled = no
admin> set bridging-options bridging-group = 572
admin> set bridging-options bridge = yes
admin> set bridging-options bridge-type = transparent-bridging
admin> set bridging-options packet-flows = bridgesched
admin> set telco-options nailed-groups = 101
admin> set atm-options vci = 70
admin> set atm-options nailed-group = 101
admin> write -f
admin> new packet-flows bridgesched
admin> set flow 1 packet-classifier source-ip-address = 9.0.0.0/8
admin> set flow 1 packet-classifier source-ip-netmask = 255.0.0.0
admin> set flow 1 scheduling queue-priority = 6
admin> set flow 2 packet-classifier source-ip-address = 8.0.0.0/8
admin> set flow 2 packet-classifier source-ip-netmask = 255.0.0.0
admin> set flow 2 scheduling queue-priority = 5
admin> set flow 3 packet-classifier source-ip-address = 7.0.0.0/8
admin> set flow 3 packet-classifier source-ip-netmask = 255.0.0.0
admin> set flow 3 scheduling queue-priority = 4
admin> write -f
```

Obtaining the interface numbers

The `ifmgr -d` command, available in the debug environment, obtains the VLAN and subscriber interface numbers (interfaces 16 and 15, respectively, in the output below). For details, see Appendix A, "IP2000 Diagnostics."

```
admin> ifmgr -d
bif slot sif u m p ifname     host-name  remote-addr       local-addr
-------------------------------------------------------------------------
000 1:08 000 *     ie0        -          0.0.0.0/32        110.110.110.72/32
001 1:08 001 *     ie1        -          0.0.0.0/32        1.1.1.1/32
```

```
006 1:08 006 *     lo0        -          0.0.0.0/32        127.0.0.1/32
007 0:00 000 *     rj0        -          0.0.0.0/32        127.0.0.2/32
008 0:00 000 *     bh0        -          0.0.0.0/32        127.0.0.3/32
009 1:08 000 *     wanabe     -          0.0.0.0/32        127.0.0.3/32
010 0:00 000 *     local      -          0.0.0.0/32        127.0.0.1/32
011 0:00 000 *     mcast      -          0.0.0.0/32        224.0.0.0/32
012 0:00 000 -     tunnel0    -          0.0.0.0/32        127.0.0.5/32
013 0:00 000 *     vr0_main   -          0.0.0.0/32        127.0.0.4/32
014 0:00 000 -     sip0       -          0.0.0.0/32        0.0.0.0/32
015 1:03 001 *   p wan15      x          0.0.0.0/32        110.110.110.72/32
016 0:00 000 *     vlan16     -          0.0.0.0/32        0.0.0.0/32
<end>
```

Enabling monitoring for the bridged VLAN interface

The following commands enable IP QoS monitoring on the bridged VLAN interface (interface 16):

```
admin> stats enable np ipqos 16
IP QOS monitoring start for IF 16
IP QOS profile: pbitmark
CLASSIFICATION & FLOW
  Default classification      :provided
  Number of classified DID    :3
  Number of class rules       :5
  Number of multicast IP flow :0
  Number of unicast IP flow   :0
PRIORITIES
  Number of priority          :4
  Lowest priority             :0
  Mapping priority/COS        :P[0]=5 P[1]=x P[2]=x P[3]=x P[4]=4 P[5]=3
P[6]=2 P[7]=x
SCHEDULING
  Line Rate                   :0
  Data queue Id               :274
  Scheduler ID                :3
  PDU type                    :packet
  Number of COS queues        :5
  Scheduling priority         :0
  Peak Rate (Kbps)            :0
  TS parameters              :00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  TM parameters              :06 0f 7d 00 05 dc 00 01 00 00 00 00 00 00 00 00
admin> stats traffic np ipqos 16
FPP counters for IF 16
Priority     COS     Counter
  drp        -         0
  exp        1         0
  6          2         143
  5          3         0
  4          4         0
  0          5         0
TM counters for IF 16
Priority     COS      MTU drop     Queue drop    Sched drop    Port drop
```

```
exp          1        0            0            0            0
6            2        0            0            0            0
5            3        0            0            0            0
4            4        0            0            0            0
0            5        0            0            0            0
```

Enabling monitoring for the bridged subscriber interface

The following commands disable monitoring on the VLAN interface (16) and then enable monitoring on the bridged VLAN subscriber interface (interface 15) with a `packet-flows` profile attached.

```
admin> stats disable np ipqos 16
```

```
admin> stats enable np ipqos 15
IP QOS monitoring start for IF 15
IP QOS profile: bridgesched
CLASSIFICATION & FLOW
  Default classification      :provided
  Number of classified DID    :3
  Number of class rules       :5
  Number of multicast IP flow :0
  Number of unicast IP flow   :0
PRIORITIES
  Number of priority          :4
  Lowest priority             :0
  Mapping priority/COS        :P[0]=5 P[1]=x P[2]=x P[3]=x P[4]=4 P[5]=3
P[6]=2 P[7]=x
SCHEDULING
  Line Rate                   :0
  Data queue Id               :262
  Scheduler ID                :3
  PDU type                    :cell
  Number of COS queues        :5
  ATM service category        :UBR
  PCR (cell per sec)          :0
  TS parameters              :00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  TM parameters              :06 4b 00 c8 00 c8 00 02 00 00 00 00 00 00 00 00
```

```
admin> stats traffic np ipqos 15
FPP counters for IF 15
Priority     COS     Counter
  drp        -         0
  exp        1         0
  6          2         173
  5          3         0
  4          4         0
  0          5         0
TM counters for IF 15
Priority     COS      MTU drop     Queue drop    Sched drop    Port drop
exp          1        0            0             0             0
6            2        0            0             0             0
5            3        0            0             0             0
4            4        0            0             0             0
0            5        0            0             0             0
```

Example of monitoring `vlan-circuit` or `stacked-vlan` bridged traffic

For `vlan-circuit` and `stacked-vlan` bridged connections only, you monitor QoS on the WAN interface only, and specify the direction of traffic (up or down).

This example shows IP QoS monitoring on the DSL WAN interface in the upstream direction for a 1:1 (`vlan-circuit`) bridged VLAN. The same procedure for monitoring QoS applies when the `connection` profile is configured for `stacked-vlan` bridging.

Creating the subscriber-side profiles

The following commands configure the subscriber-side `connection` profile and a `packet-flows` profle to monitor its traffic:

```
admin> new connection vlancir1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options ip-routing-enabled = no
admin> set bridging-options bridging-group = 572
admin> set bridging-options bridge = yes
admin> set bridging-options packet-flows = bridgesched
admin> set atm-options vci = 70
admin> set atm-options nailed-group = 101
admin> write -f
admin> new packet-flows bridgesched
admin> set flow 1 packet-classifier source-ip-address = 9.0.0.0/8
admin> set flow 1 packet-classifier source-ip-netmask = 255.0.0.0
admin> set flow 1 scheduling queue-priority = 6
admin> set flow 2 packet-classifier source-ip-address = 8.0.0.0/8
admin> set flow 2 packet-classifier source-ip-netmask = 255.0.0.0
admin> set flow 2 scheduling queue-priority = 5
admin> set flow 3 packet-classifier source-ip-address = 7.0.0.0/8
admin> set flow 3 packet-classifier source-ip-netmask = 255.0.0.0
admin> set flow 3 scheduling queue-priority = 4
admin> write -f
```

Creating the VLAN-side profiles

The next set of commands configure the `vlan-ethernet` profile for the 1:1 bridged VLAN, and a `packet-flows` profile to monitor its traffic:

```
admin> new vlan-ethernet { { 1 8 2 } 572 }
admin> set enabled = yes
admin> set bridging-options bridge = yes
admin> set bridging-options packet-flows = pbitmark
admin> write -f
admin> new packet-flows pbitmark
```

```
admin> set marking-type = ethernet-priority
admin> set flow 1 packet-classifier source-ip-address = 9.0.0.0/8
admin> set flow 1 packet-classifier source-ip-netmask = 255.0.0.0
admin> set flow 1 scheduling queue-priority = 6
admin> set flow 2 packet-classifier source-ip-address = 8.0.0.0/8
admin> set flow 2 packet-classifier source-ip-netmask = 255.0.0.0
admin> set flow 2 scheduling queue-priority = 5
admin> set flow 3 packet-classifier source-ip-address = 7.0.0.0/8
admin> set flow 3 packet-classifier source-ip-netmask = 255.0.0.0
admin> set flow 3 scheduling queue-priority = 4
admin> set global-packet-marking 5 ethernet-priority = 5
admin> set global-packet-marking 6 ethernet-priority = 6
admin> set global-packet-marking 7 ethernet-priority = 7
admin> write -f
```

Obtaining the interface numbers

The `ifmgr -d` command, available in the debug environment, obtains the subscriber interface number (interface 15 in the output below). For details, see Appendix A, "IP2000 Diagnostics."

```
admin> ifmgr -d
bif slot sif u m p ifname     host-name  remote-addr       local-addr
-------------------------------------------------------------------------
-
000 1:08 000 *     ie0        -          0.0.0.0/32        110.110.110.72/32
001 1:08 001 *     ie1        -          0.0.0.0/32        1.1.1.1/32
006 1:08 006 *     lo0        -          0.0.0.0/32        127.0.0.1/32
007 0:00 000 *     rj0        -          0.0.0.0/32        127.0.0.2/32
008 0:00 000 *     bh0        -          0.0.0.0/32        127.0.0.3/32
009 1:08 000 *     wanabe     -          0.0.0.0/32        127.0.0.3/32
010 0:00 000 *     local      -          0.0.0.0/32        127.0.0.1/32
011 0:00 000 *     mcast      -          0.0.0.0/32        224.0.0.0/32
012 0:00 000 -     tunnel0    -          0.0.0.0/32        127.0.0.5/32
013 0:00 000 *     vr0_main   -          0.0.0.0/32        127.0.0.4/32
014 0:00 000 -     sip0       -          0.0.0.0/32        0.0.0.0/32
015 1:03 001 *   p wan15      vlancir1   0.0.0.0/32        110.110.110.72/32
016 0:00 000 *     vlan16     -          0.0.0.0/32        0.0.0.0/32
<end>
```

Enabling monitoring in the upstream direction

The following commands enable IP QoS monitoring in upstream direction and show that the QoS counters are incremented.

```
admin> stats enable np ipqos 15 up
IP QOS monitoring start for IF 15 flow direction - upstream
IP QOS profile: pbitmark
QoS monitoring direction: upstream
CLASSIFICATION & FLOW
```

```
  Default classification      :provided
  Number of classified DID    :3
  Number of class rules       :5
  Number of multicast IP flow :0
  Number of unicast IP flow   :0
PRIORITIES
  Number of priority          :4
  Lowest priority             :0
  Mapping priority/COS        :P[0]=5 P[1]=x P[2]=x P[3]=x P[4]=4 P[5]=3
P[6]=2 P[7]=x
SCHEDULING
  Line Rate                   :0
  Data queue Id               :274
  Scheduler ID                :3
  PDU type                    :packet
  Number of COS queues        :5
  Scheduling priority         :0
  Peak Rate (Kbps)            :0
  TS parameters              :00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  TM parameters              :06 0f 7d 00 05 dc 00 01 00 00 00 00 00 00 00 00
```

```
admin> stats traffic np ipqos 15
QoS monitoring direction: upstream
FPP counters for IF 15
Priority     COS     Counter
  drp        -         0
  exp        1         0
  6          2         150
  5          3         0
  4          4         0
  0          5         0
TM counters for IF 15
Priority     COS      MTU drop     Queue drop    Sched drop    Port drop
exp          1        0            0             0             0
6            2        0            0             0             0
5            3        0            0             0             0
4            4        0            0             0             0
0            5        0            0             0             0
```

Enabling monitoring in the downstream direction

The following commands enable IP QoS monitoring in downstream direction and show that the QoS counters are incremented.

```
admin> stats enable np ipqos 15 down
IP QOS monitoring start for IF 15 flow direction - downstream
IP QOS profile: bridgesched
QoS monitoring direction: downstream
CLASSIFICATION & FLOW
  Default classification      :provided
  Number of classified DID    :3
  Number of class rules       :5
  Number of multicast IP flow :0
  Number of unicast IP flow   :0
```

```
PRIORITIES
  Number of priority          :4
  Lowest priority             :0
  Mapping priority/COS        :P[0]=5 P[1]=x P[2]=x P[3]=x P[4]=4 P[5]=3
P[6]=2 P[7]=x
SCHEDULING
  Line Rate                   :0
  Data queue Id               :262
  Scheduler ID                :3
  PDU type                    :cell
  Number of COS queues        :5
  ATM service category        :UBR
  PCR (cell per sec)          :0
  TS parameters              :00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  TM parameters              :06 4b 00 c8 00 c8 00 02 00 00 00 00 00 00 00 00
```

```
admin> stats traffic np ipqos 15
QoS monitoring direction: downstream
FPP counters for IF 15
Priority     COS     Counter
  drp        -         0
  exp        1         0
  6          2         139
  5          3         0
  4          4         0
  0          5         0
TM counters for IF 15
Priority     COS      MTU drop     Queue drop    Sched drop    Port drop
exp          1        0            0             0             0
6            2        0            0             0             0
5            3        0            0             0             0
4            4        0            0             0             0
0            5        0            0             0             0
```

Limitations with the current software version

This section lists limitations that apply to the IP QoS implementation with this software version.

- The Stinger IP2000 can support classification among different multicast streams that are going to the same output interface. Thus, for example, two streams, one to the destination address 224.1.1.1 and the second to the destination address 224.1.1.2 but that go on the same DSL user connection can be classified to have 2 different priorities.

  Prioritization between unicast and multicast flows on the same connection is also supported.

  However, subclassification within a multicast flow cannot be done. Thus, for example, UDP packets on 224.1.1.1 and TCP packets going to 224.1.1.1 cannot be distinguished.

- Only 32 packet flows are supported per interface.

  If 32 packet flows have been defined in the `packet-classifier` subprofile, and there is no default rule, the Stinger system creates an implicit default rule default.

- For fragmented IP packets, only the IP header is available for classification, which means that classification based on UDP/TCP source or destination ports cannot be used.
- Packets that are fragmented by the Stinger system always take the highest priority. Classifications rules for such packets are not applicable.
- IP QoS does not apply to L2TP connections.
- When a nonzero `source-port` or `destination-port` is specified in a `packet-flows` profile, the `ip-protocol` value must be set to either `17` (UDP) or `6` (TCP).
- SNMP support is not provided with the current software version.
- The precision of the rate-limiting policer is about 1%.
- The minimum policing rate (the minimum CIR) is 17Kbps.
- The minimum scheduling rate for a logical interface on the Gigabit Ethernet port is 401Kbps.
- Rate limiting is not supported for bridged connections and interfaces.
- A given `packet-flows` profile can have either `layer2-classifier` or IP-specific `packet-classifier` settings, but not both.

Virtual Router Configuration

6

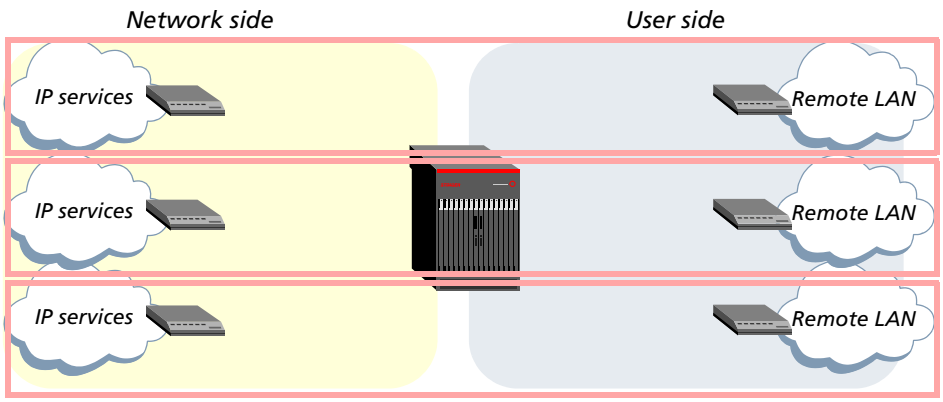
Virtual routing enables you to partition virtual domains within the global IP router. Each virtual domain is defined by a named virtual router. Currently, the Stinger IP2000 controller supports up to 1022 virtual routers in addition to the global router.

If you do not configure virtual routers, the global IP router operates as documented in Chapter 4, "IP Routing Configuration." All interfaces that are not explicitly grouped with a defined virtual router are grouped with the global router.

Overview of virtual routing

Virtual routing enables high-density circuit termination with secure logical partitioning and multiple route tables. Virtual routing is particularly useful for remote access server (RAS) functionality. For example, it can securely partition traffic from many CPE devices to different ISPs, with each ISP mapped to a separate virtual domain. Figure 6-1 shows a simplified diagram with three virtual routers configured in the global router. Because each virtual router has its own routing table, traffic within a virtual domain is typically not shared with other domains.

Figure 6-1. Simple diagram of three virtual domains (virtual routers)



How virtual routers affect the routing table

When virtual routers are not defined, the global router maintains a single IP routing table that enables the router to reach any of its many interfaces. In that context, each interface known to the system requires a unique address.

With virtual routers, addresses must be unique within the virtual domain, but not necessarily within the Stinger IP2000. Because each virtual router maintains its own routing table, and because it knows about only those interfaces that explicitly specify the same virtual router, there is no requirement that the private networks maintain unique address spaces, as long as the virtual domains are not interconnected.

Interconnecting virtual domains

Each virtual router has its own associated routing table, ARP table, route cache, and address pools, which cannot be shared with another virtual router. However, it is possible to interconnect two virtual domains by defining an inter-virtual-router route. For details, see "Specifying an inter-virtual-router route" on page 6-9.

Note Since routing traffic between virtual domains is not fast routed, it is strongly recommended to minimize such traffic.

Applicability and limitations

When configuring virtual routing on a Stinger IP2000, consider the following issues related to virtual router applicability and limitations:

- Virtual routing does not apply to switched or bridged connections.
- Only terminated virtual circuits (IP, PPPoA, PPPoE) can be integrated into a virtual domain.
- When the virtual LAN (VLAN) feature is used in its usual bridging capacity, to form a bridge between a VLAN ID on Gigabit Ethernet and a DSL interface, virtual routing does not apply. If the VLAN is not used in a bridge capacity but acts as an IP interface, virtual routing applies.
- Bridged IP routing (BIR) profiles have no special considerations for virtual routing. The `connection` profile terminating the PVC must define the virtual router.
- With the current implementation, multicasting applications must use the global router.
- If multiple virtual IP interfaces are configured on a physical interface, particularly an Ethernet interface, all virtual interfaces must be attached to the same virtual router.
- SNMP management utilities do not currently display information on a per-virtual-router basis.
- Errors and events are not logged on a per-virtual-router basis.
- The `syslog` host defined in the system's `log` profile must be accessible to the global router.
- Servers defined in the `debug`, `trap`, `external-auth`, `ip-global` (for SNTP and multicast), `call-logging`, and `snmp` profiles must be accessible to the global router.

Creating a virtual router

When at least one `vrouter` profile is configured, the `system-ip-address` parameter and the `global-vrouter` parameter in the `ip-global` profile apply to the global router. All interfaces that are not explicitly assigned to another virtual router are grouped with the global router.

For each virtual router in the system, an instance of RIP is created to process routes. The new instance of RIP sends and receives update packets only on the interfaces associated with its particular virtual router and manipulates only that virtual router's routing table. A default instance of RIP is always created for the global router.

When you create a virtual router, the new instance of RIP sends and receives packets only on the interfaces associated with that virtual router and manipulates only that virtual router's routing table. All RIP-related parameters in a `vrouter` profile use default settings that are recommended for most sites.

Overview of `vrouter` profile settings

A `vrouter` profile contains the following parameters, shown with default values:

```
[in VROUTER/"" ]
name* = ""
active = yes
vrouter-ip-addr = 0.0.0.0
pool-base-address = [ 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0 0.0.0.0+
assign-count = [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 +
pool-name = [ "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "+
pool-summary = no
share-global-pool = yes
rip-policy = Poison-Rvrs
summarize-rip-routes = no
rip-trigger = yes
domain-name = ""
sec-domain-name = ""
dns-primary-server = 0.0.0.0
dns-secondary-server = 0.0.0.0
client-primary-dns-server = 0.0.0.0
client-secondary-dns-server = 0.0.0.0
allow-as-client-dns-info = True
```

Parameter	Setting
`name`	Unique name for the virtual router, up to 15 characters. All interfaces belonging to a virtual router specify the same virtual router name in the `ip-interface` or `connection` profile.
`active`	Activate the virtual router.
`vrouter-ip-address`	System IP address for the virtual router.

Parameter	Setting
`pool-base-address`	Base address of a pool of contiguous addresses on a local network or subnet. The pool will be exclusively for use by the virtual router. For details about defining address pools, see "Configuring and using address pools" on page 4-13.
`assign-count`	Number of addresses in the pool. The pool will be exclusively for use by the virtual router. For details about defining address pools, see "Configuring and using address pools" on page 4-13.
`pool-name`	A pool name, required only when TACACS+ authentication is in use. The pool will be exclusively for use by the virtual router. For details about defining address pools, see "Configuring and using address pools" on page 4-13.
`pool-summary`	Set/clear the pool summary flag to specify that the address pools will be summarized. For details about defining address pools that can be summarized, see "Configuring and using address pools" on page 4-13.
`share-global-pool`	Enable/disable the virtual router to share the address pools defined in the `ip-global` profile.
`rip-policy`	Policy for the virtual router to use when sending update packets that include routes received on the same interface. For details, see "Setting RIP options" on page 4-10.
`summarize-rip-routes`	Whether the virtual router summarizes subnet information in RIP-v1 advertisements. For details about this feature, see "Setting RIP options" on page 4-10.
`rip-trigger`	Enable/disable RIP triggering for the virtual router. For details about RIP triggering, see "Setting RIP options" on page 4-10.

Note For details about `domain-name` and other DNS parameters, see "Configuring virtual router DNS servers" on page 6-10.

Example of defining a virtual router

The following commands create a virtual router named `vr1` with a system address of 130.200.200.100:

```
admin> new vrouter vr1
admin> set vrouter-ip-addr = 130.200.200.100
admin> write -f
```

The vr1 virtual router maintains minimal routing and interface tables at this point, as shown in the following sample output:

```
admin> netstat vr1 -rn
```

Destination	Gateway	IF	Flg	Pref	Met	Use	Age
127.0.0.0/8	-	bh0_vr1	CP	0	0	1	6815
127.0.0.1/32	-	local	CP	0	0	1	6815
127.0.0.2/32	-	rj0_vr1	CP	0	0	1	6815
224.0.0.9/32	-	local	CP	0	0	1	6815

```
Total Routes = 4    Hidden Routes = 0

admin> netstat vr1 -in
```

Name	MTU	Net/Dest	Address	Ipkts	Ierr	Opkts	Oerr
vr0_vr1	1500	127.0.0.4/32	127.0.0.4	0	0	0	0
lo0_vr1	1500	127.0.0.1/32	127.0.0.1	0	0	0	0
local	65535	127.0.0.1/32	127.0.0.1	0	0	0	0
rj0_vr1	1500	127.0.0.2/32	127.0.0.2	0	0	0	0
bh0_vr1	1500	127.0.0.3/32	127.0.0.3	0	0	0	0

The virtual router also maintains its own IP, TCP, UDP, and ICMP statistics. For example:

```
admin> netstat vr1 -s
udp:
       1442 packets received
       0 packets received with no ports
       0 packets received with errors
       0 packets dropped
       32 packets transmitted
tcp:
       0 active opens
       1 passive opens
       0 connect attempts failed
       0 connections were reset
       1 connections currently established
       858 segments received
       0 segments received out of order
       548 segments transmitted
       0 segments retransmitted
       0 active closes
       0 passive closes
       0 disconnects while awaiting retransmission
icmp:
       31 packets received
       0 packets received with errors
       Input histogram:
               30 echo requests
               1 netmask requests

       31 packets transmitted
```

```
        0 packets not transmitted due to lack of resources
        Output histogram:
                30 echo replies
                1 netmask replies

ip:
        0 packets received
        0 packets received with header errors
        0 packets received with address errors
        0 packets received forwarded
        0 packets received with unknown protocols
        0 inbound packets discarded
        0 packets delivered to upper layers
        0 transmit requests
        0 discarded transmit packets
        0 outbound packets with no route
        0 reassemblies timeout
        0 reassemblies required
        0 reassemblies succeeded
        0 reassemblies failed
        0 fragmentation succeeded
        0 fragmentation failed
        0 fragmented packets created
        0 route discards due to lack of memory
        64 default ttl
igmp:
        0 packets received
        0 bad checksum packets received
        0 bad version packets received
        0 query packets received
        0 leave packets received
        0 packets transmitted
        0 query packets sent
        0 resonse packets sent
        0 leave packets sent
mcast:
        0 packets received
        0 packets forwarded
        0 packets in error
        0 packets dropped
        0 packets transmitted
pim:
        0 packets received
        559 packets transmitted
        559 hello packets sent
```

Note Multicast is not currently supported on a per-virtual-router basis, so the IGMP, multicast, and PIM statistics relate only to the global router.

Defining address pools for a virtual router

The following commands define an address pool for the vr1 virtual router defined in "Example of defining a virtual router" on page 6-4:

```
admin> read vrouter vr1

admin> set pool-base 1 = 130.100.100.128

admin> set assign-count 1 = 127

admin> write -f
```

Following is a comparable RADIUS pool definition:

```
pools-taos01 Password = "ascend", Service-Type = Outbound-User
    Ascend-IP-Pool-Definition = "1 130.100.100.128 127 vr1"
```

The vr1 virtual router is now maintaining the following pool of addresses:

```
admin> ip-pools vr1
Pool#                Base         Count      InUse
   1          130.100.100.128      127          0
Number of remaining allocated addresses:    0
```

Note The Ascend-IP-Pool-Definition attribute supports a virtual router name as the last syntax element in a pool definition. The value of Ascend-IP-Pool-Definition uses the following syntax:

"*pool-num base-addr assign-count* [*vrouter-name*]"

For background information about address pools, see "Configuring and using address pools" on page 4-13. The process of defining address pools for a virtual router is the same as described in that section.

Assigning interfaces to a virtual router

To assign virtual router membership to an interface, you specify a virtual router name in the interface profile. For a virtual router to be active, at least one IP interface (LAN or WAN) must specify its name.

Overview of interface **vrouter** settings

To assign virtual router membership to an interface in local profiles, set the vrouter parameter. For example:

```
[in IP-INTERFACE/{ { shelf-1 slot-8 2 } 0 } ]
vrouter = vr1

[in CONNECTION/vr1-client]
vrouter = vr1
```

Parameter	RADIUS attribute	Setting
vrouter	Ascend-VRouter-Name (102)	Name of a defined virtual router. Specifying the virtual router name groups the interface with the virtual router. The default null value specifies the global router.

Examples of assigning virtual router membership to interfaces

The following commands assigns a WAN interface to the `vr1` virtual router:

```
admin> read connection router-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set vrouter = vr1
admin> set ip-options remote-address = 10.7.8.200/30
admin> set atm-options vci = 100
admin> set atm-options nailed-group = 201
admin> write -f
```

Following is a comparable RADIUS profile:

```
permconn-st-1 Password = "ascend"
    Service-Type = Outbound-User,
    Framed-Protocol = ATM-1483,
    User-Name = "router-1",
    Framed-IP-Address = 10.7.8.200,
    Framed-IP-Netmask = 255.255.255.252,
    Ascend-ATM-Group = 201,
    Ascend-Route-IP = Route-IP-Yes,
    Ascend-ATM-Vpi = 8,
    Ascend-ATM-Vci = 100,
    Ascend-Vrouter-Name = "vr1"
```

Defining virtual router static routes

You specify a static route associated with a virtual router for one of the following reasons:

- To define a route on a per-virtual-router basis
- To specify an inter-virtual-router route

Overview of static route settings

Following are the virtual router static route parameters (shown here with default values) in `ip-route` profiles:

```
[in IP-ROUTE/""]
vrouter = ""
inter-vrouter = ""
```

Parameter	Setting
`vrouter`	Name of the virtual router that will own this route. The route will be part of the specified virtual router's routing table. If no name is specified (the default), the global router is assumed.

Parameter	Setting
`inter-vrouter`	Name of a virtual router to use as the route's next hop. All packets to the static route's destination network are sent to the specified virtual router for a routing decision. The `gateway-address` parameter must be set to the zero address for this parameter to apply.

In a RADIUS profile, the value of the Framed-Route (22) attribute can specify a virtual router name in the following syntax:

"*dest-addr* [*/prefix*] *gateway-addr metric* [private] [*profile*] [*preference*] [*vrouter-name*]"

Note The fields within the value of the Framed-Route attribute are positional. With the exception of the optional prefix-length specification, if any of the optional fields are specified, the optional fields to the left of that setting must also be specified.

Examples of defining a route on a per-virtual-router basis

When you define a route on a per-virtual-router basis, it appears only in the specified virtual router's routing tables. That virtual router "owns" the route.

Following is an example of defining a static route within the `vr1` virtual router domain. This route will appear only in the routing table for `vr1`.

```
admin> new ip-route rtr1
admin> set dest = 10.5.6.7/28
admin> set gateway = 10.1.1.1
admin> set vrouter = vr1
admin> write -f
```

Following is a comparable RADIUS profile:

```
route-taos-1 Password = "ascend", Service-Type = Outbound-User
    Framed-Route = "10.5.6.7/28 10.1.1.1 7 n rtr1 60 vr1"
```

The following sample output shows the new static route that was added to the `vr1` virtual router's routing table:

```
admin> netstat vr1 -rn
Destination     Gateway      IF            Flg   Pref Met    Use        Age
10.1.1.0/24     10.1.1.1     wan30         SG     120   7      0          9
10.1.1.1/32     10.1.1.1     wan30         S      120   7      2          9
10.5.6.0/28     10.1.1.1     wan30         SG      60   8      0          9
11.1.1.0/24     11.1.1.1     wan31         SG     120   7      0          9
11.1.1.1/32     11.1.1.1     wan31         S      120   7      1          9
12.1.1.0/24     12.1.1.1     wan32         SG     120   7      0          9
12.1.1.1/32     12.1.1.1     wan32         S      120   7      1          9
127.0.0.0/8     -            bh0_vr1       CP       0   0      0       2274
127.0.0.1/32    -            local         CP       0   0      0       2274
127.0.0.2/32    -            rj0_vr1       CP       0   0      0       2274
```

Specifying an inter-virtual-router route

You can cause one virtual router or the global router to forward traffic to another virtual router for a routing decision by specifying an inter-virtual-router static route.

Note Because routing traffic between virtual domains is not fast routed, it is strongly recommended to minimize such traffic.

In the following example, the static route specifies the `vr1` virtual router as the route's next hop. This route is not defined on a per-virtual-router basis, so it is owned by the global router.

```
admin> new ip-route rtr2
admin> set dest-address = 11.0.0.0/24
admin> set inter-vrouter = vr1
admin> write -f
```

Following is a comparable RADIUS route profile:

```
route-taos-1 Password = "ascend", Service-Type = Outbound-User
    Framed-Route = "11.0.0.0/24 0.0.0.0 vr1"
```

The following output shows that the route has been added to the global router's routing table:

```
admin> netstat -rn
Destination      Gateway     IF          Flg  Pref Met    Use     Age
0.0.0.0/0        10.1.6.1    ie0         SGP   60   1      59       4
11.0.0.0/24       -          vr0_vr1      S    60   8       0       4
20.0.0.0/8        -          ie1-12-1     C     0   0      12     234
20.1.1.2/32       -          local       CP     0   0       0    2347
127.0.0.0/8       -          bh0         CP     0   0       0    2378
127.0.0.1/32      -          local       CP     0   0       0    2378
127.0.0.2/32      -          rj0         CP     0   0       0    2378
130.1.1.1/32      -          sip0         C     0   0       0    2378
130.1.1.252/30    -          rj0          C     0   0       0    2378
100.1.6.0/24     100.1.6.221 wanabe      SG    60   1       0       4
101.1.6.0/24      -          ie0          C     0   0    2531    2378
101.1.6.234/32    -          local       CP     0   0    4152    2378
224.0.0.0/4       -          mcast       CP     0   0       0    2378
224.0.0.1/32      -          local       CP     0   0       0    2378
224.0.0.2/32      -          local       CP     0   0       0    2378
224.0.0.5/32      -          local       CP     0   0     732    2378
224.0.0.6/32      -          local       CP     0   0       0    2378
255.255.255.255/32 -         ie0          P     0   0     422    2378
```

Configuring virtual router DNS servers

Virtual router DNS configuration includes settings for primary and secondary DNS servers, domain names, and client DNS servers. The settings direct connections that belong to the virtual router to a particular DNS service. To completely segment the virtual router's DNS information from any other hosts, you can configure and manage DNS information separately for each virtual router. The addresses configured for client DNS servers are presented to dial-in users during IP Control Protocol (IPCP) negotiation.

If DNS information is not found in the `vrouter` profile, the system uses the DNS information in the `ip-global` profile. The DNS list and the local DNS table maintained in RAM are systemwide DNS configurations that are not supported separately for each virtual router.

Overview of virtual router DNS settings

Following are the virtual router-specific DNS parameters (shown with their default settings):

```
[in VROUTER/""]
domain-name = ""
sec-domain-name = ""
dns-primary-server = 0.0.0.0
dns-secondary-server = 0.0.0.0
client-primary-dns-server = 0.0.0.0
client-secondary-dns-server = 0.0.0.0
allow-as-client-dns-info = True
```

Parameter	Setting
`domain-name`	Primary domain name (up to 63 characters) to use for DNS lookups for this virtual router. The system appends this domain name to hostnames when performing lookups.
`sec-domain-name`	Secondary domain name to use for DNS lookups for this virtual router if the hostname is not found in the primary domain.
`dns-primary-server`	Address of the primary local DNS server to use for lookups for this virtual router.
`dns-secondary-server`	Address of the secondary local DNS server to use for lookups for this virtual router. Used only if the primary server is not found.
`client-dns-primary-server`	Address of a client DNS server for dial-in clients of this virtual router.
`client-dns-secondary-server`	Address of a secondary DNS server for dial-in clients of this virtual router.
`allow-as-client-dns-info`	Enable/disable use of local DNS information if the client DNS servers are not found. To isolate local network information for this virtual router, set to `false`.

Example of a typical virtual router DNS configuration

The following commands specify a primary and secondary domain name for DNS lookups for a virtual router named `xyz`:

```
admin> read vrouter xyz
admin> set domain-name = xyz.com
admin> set sec-domain-name = eng.xyz.com
admin> write -f
```

If a lookup fails in the first domain, the router tries again with the secondary domain name. To enable the system to use DNS to look up addresses, specify DNS server addresses, as shown in the following example:

```
admin> read vrouter xyz
admin> set dns-primary-server = 1.2.2.2
admin> set dns-secondary-server = 1.3.3.3
admin> write -f
```

If the primary server is unavailable, the system attempts a lookup on the secondary server. The following commands configure a client DNS server for this virtual router:

```
admin> read vrouter xyz
admin> set client-dns-primary-server = 1.2.2.2
admin> set client-dns-secondary-server = 1.2.2.96
admin> set allow-as-client-dns-info = false
admin> write -f
```

The secondary server is accessed only if the primary one is inaccessible. If both of these client DNS servers are not accessible, the system does not allow the client to access local DNS servers.

Deleting a virtual router

You can delete a virtual router only if no more interfaces are attached to it. If one interface is attached to a virtual router, the system prevents its deletion. To delete a virtual router that has no attached interfaces, delete the `vrouter` profile. For example:

```
admin> delete vrouter vr1
```

Lucent Technologies recommends that you reset the system after deleting a virtual router with active connections. If a system reset is not possible, the recommended course of action before deleting the virtual router is to manually tear down its active connections, and then modify the local `connection`, `ip-interface`, and `ip-route` profiles that point to the virtual router to point instead to the global router or another existing virtual router.

Administrative tools for virtual routers

You can specify a virtual router name on the command line of the network administration commands listed in Table 6-1 to obtain information specific to a particular virtual domain.

Table 6-1. Administrative commands showing optional vrouter arguments

Command	Permissions	Usage with optional vrouter argument
`arptable`	`system`	`arptable [`*`vrouter`*`] [[-a `*`hostname MAC_address`*`] | [-d `*`hostname`*`] | [-f]]`
`ipcache`	`system`	`ipcache [-r `*`vrouter`*`] [cache] [stats]`
`iproute`	`system`	`iproute add [-r `*`vrouter`*`] `*`dest_IPaddress/subnet_mask gateway_IPaddress`*` [`*`preference`*`] [metric]` `iproute delete [-r `*`vrouter`*`] `*`dest_IPaddress/subnet_mask`*` [gateway]`

Table 6-1. Administrative commands showing optional vrouter arguments (Continued)

Command	Permissions	Usage with optional vrouter argument
netstat	system	netstat [*vrouter*] [-i] [-r] [?] [-n] [-d] [-s *identifiers*] [-z]
vrouter	system	vrouter [dump [full]] [callback]
nslookup	diagnostic	nslookup [-v] [-s *dnssrvr_IPaddress*] [-r *vrouter*] *hostname*
ping	diagnostic	ping [-q | -v] [-i *delay_sec* | -I *delay_msec*] [-s *packetsize*] [-r *vrouter*] [-x *source_IPaddress*] *hostname*
telnet	diagnostic	telnet [-a | -b | -t] [-v *vrouter*] [-l[e] | -r[e]] *hostname* [*portnumber*]
traceroute	diagnostic	traceroute [-n] [-v] [-m *max_ttl*] [-p *port*] [-q *nqueries*] [-w *waittime*] [-r *vrouter*] [-s src_IPaddr] *hostname* [*datasize*]
info np	debug	info np arp [*vrouter*] info np route [*vrouter*]
ifmgr	debug	ifmgr [-r *vrouter*] -d [*ifNum*] -t ifmgr [up|down] [ifNum|ifName]

For details about the system or diagnostic commands in Table 6-1, see the *Stinger Reference*. For information about the debug-level commands, see Appendix A, "IP2000 Diagnostics."

OSPF Configuration

7

Open Shortest Path First (OSPF) is an Internet routing protocol, developed by the OSPF working group of the Internet Engineering Task Force, and defined in RFC 2328, *OSPF Version 2*.

OSPF was designed for the TCP/IP Internet environment, including explicit support for IP subnets, tagging of externally derived routing information, and use of IP multicast for sending or receiving link-state updates.

Overview of OSPF features supported by the IP2000

This section provides a brief overview of OSPF routing to help you configure the system properly. For details about how OSPF works, see RFC 2328, *OSPF Version 2*. Following are the OSPF functions discussed in this section:

- Multiple authentication schemes, per RFC 1583
- Variable length subnet masks (VLSMs)
- Link state advertisement (LSA) types 1, 2, 3, 4, 5, and 7
- Backup and designated backup router capability on broadcast networks
- Nonbroadcast multiaccess (NBMA) network support over point-to-point links
- Configurable cost metrics
- Hierarchical routing via normal areas, stub areas, and not-so-stubby-areas (NSSAs)
- Shortest path first link-state routing algorithm
- Diagnostics and traps

Limited border router capability

A Stinger IP2000 system acts as an OSPF internal router with limited border router capability. It does not currently function as an interior gateway protocol (IGP) gateway, although it performs autonomous system border router (ASBR) calculations for external routes (such as WAN links that do not support OSPF). The Stinger unit imports external routes into its OSPF database and flags them as autonomous system external (ASE). It redistributes these routes via OSPF ASE advertisements, and propagates its OSPF routes to remote WAN routers running RIP.

One active IP interface per port

If more than one IP address is assigned to the same physical port, only one of the logical interfaces can have OSPF enabled. For example, in the following listing the Gigabit Ethernet port has one virtual interface:

```
admin> dir ip-interface
    27  05/06/2004 09:27:35  { { any-shelf any-slot 0 } 0 }
    46  05/13/2004 10:53:46  { { shelf-1 first-control-module 1 } 0 }
    39  08/12/2004 08:37:48  { { shelf-1 first-control-module 2 } 0 }
    54  08/12/2004 08:37:27  { { shelf-1 first-control-module 2 } 1 }
    30  08/10/2004 10:14:31  { { shelf-1 slot-14 1 } 0 }
    30  08/10/2004 10:14:31  { { shelf-1 slot-14 2 } 0 }
```

OSPF can be enabled on one of the port's IP interfaces, but not on more than one interface for the same port.

Authentication

All OSPF protocol exchanges are authenticated by simple authentication by default. Only trusted routers can participate in the autonomous system's routing. A variety of authentication schemes can be used. In fact, different authentication types can be configured for each area. For a discussion of areas, see "Hierarchical routing (areas)" on page 7-5.

Authentication provides added security for the routers that are on the network. Routers that do not have the password are unable to gain access to the routing information, because authentication failure prevents a router from forming adjacencies. (For a discussion of adjacencies, see "Exchange of routing information" on page 7-3.) If both sides of a connection do not support the same authentication method, packet error messages can result.

In addition to null and simple password authentication, Stinger units support the MD5 cryptographic authentication method for OSPF as described RFC 2328. For details about MD5 encryption, see RFC 2328.

Support for variable-length subnet masks

OSPF routers handle variable-length subnet masks (VLSMs). Each route distributed by OSPF has a destination address and subnet mask, and two different subnets of the same IP network can use different size subnet masks. A packet is routed to the best (longest or most specific) match. Host routes are considered to be subnets whose masks are all ones (0xFFFFFFFF).

Note OSPF is useful for networks that use VLSMs. However, to prevent excessive link-state calculations by all OSPF routers on the network, make every effort to assign subnets that are as continuous as possible.

Exchange of routing information

An OSPF router stores its information about the network in a topological database and propagates only changes to the database. Selected neighboring routers form relationships, referred to as *adjacencies*, for the purpose of exchanging routing information. Not every pair of neighboring routers become adjacent. Routers connected by point-to-point networks and virtual links always become adjacent. On multiaccess networks, all routers become adjacent to routers identified as the designated router (DR) and the backup designated router (BDR).

As the adjacency is established, the neighbors exchange databases and build a consistent, synchronized database between them. When an OSPF router detects a change on one of its interfaces, it modifies its topological database and multicasts the change to its adjacent neighbors, which in turn propagate the change to their adjacent neighbors, until all routers within an area have synchronized topological databases. This process provides quick convergence among routers.

A link state advertisement (LSA) is a packet that describes various aspects of an OSPF route. Each LSA is flooded throughout a routing domain. The collected LSAs of all routers and networks forms the OSPF topological database. Table 7-1 shows the types of LSAs.

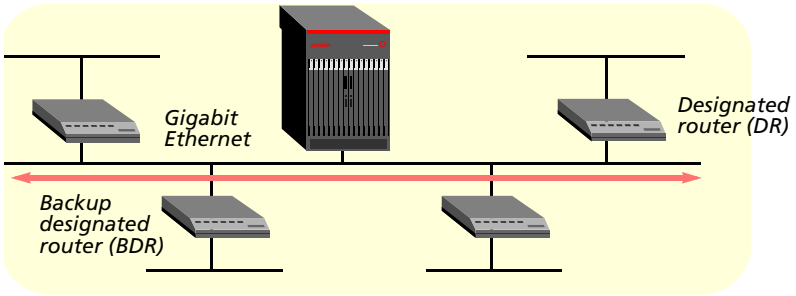
Table 7-1. Description of LSA types

LSA type	Description
Type 1 (RTR) router	Type 1 LSAs describe the collected states of the router's interfaces.
Type 2 (NET) network	Type 2 LSAs describe the set of routers attached to the network.
Types 3 and 4 (Summary LSAs)	Summary LSAs are flooded throughout a single area. Type 3 summary LSAs describe routes to networks. Type 4 summary LSAs describe routes to autonomous system boundary routers.
Type 5 (ASE) AS-external	Type 5 LSAs describe routes to destinations external to the autonomous system (AS). An AS-external-LSA can also describe a default route for the autonomous system. For example, other routers send LSAs to only the designated router by using the All-Designated-Routers multicast address of 224.0.0.6.
Type 7 (ASE) NSSA	NSSAs are like stub areas in that they do not receive or originate type 5 LSAs. However, NSSAs rely solely on default routing for external routes. They employ type 7 LSAs for carrying ASE route information within the area. Type 7 LSAs use a propagate (P) bit to flag the NSSA border router to translate the type 7 LSA into a type 5 LSA, which can then be propagated into other areas.

Designated and backup designated routers on broadcast networks

In OSPF terminology, a broadcast network is any network that has more than two OSPF routers attached and supports the capability to address a single physical message to all the attached routers (broadcast). Neighboring routers are discovered dynamically on these networks using the OSPF Hello protocol, which uses the broadcast capability. Ethernet is an example of a broadcast network. Figure 7-1 shows such a network.

Figure 7-1. OSPF broadcast network on Gigabit Ethernet



To reduce the number of adjacencies each router must form, OSPF calls one of the routers the designated router. As routers begin to form adjacencies, they elect a designated router and then all other routers on the network establish adjacencies, primarily with the designated router. This process simplifies the routing table update procedure and reduces the number of link-state records in the database. The designated router plays other important roles as well to reduce the overhead of OSPF link-state procedures.

To prevent the designated router from becoming a serious liability to the network if it fails, OSPF routers also elect a backup designated router at the same time. Other routers maintain adjacencies with both the designated router and its backup, but the backup router leaves as many of the processing tasks as possible to the designated router. If the designated router fails, the backup immediately becomes the designated router and a new backup is elected.

You choose the designated router on the basis of the processing power, speed, and memory of the system, then assign priorities to other routers on the network, in case both the designated and backup designated routers fail.

Note The Stinger unit can function as a designated router or backup designated router. However, many sites choose to assign a LAN-based router for these roles to dedicate the Stinger unit to WAN processing.

Routing across NBMA interfaces

An OSPF nonbroadcast multiaccess (NBMA) network is any network that has multiple points of access (more than two routers) and does not support broadcast capability. OSPF routers operate on an NBMA network much as they do on a broadcast network, by using the Hello protocol to form adjacencies and identify the designated router. However, because the routers cannot discover their neighboring routers dynamically by means of broadcasts, you must specify some additional parameters. For Stinger units, a WAN link is always point-to-point. However, you can configure the system to interact with NBMA routers via a virtual circuit across an ATM trunk interface.

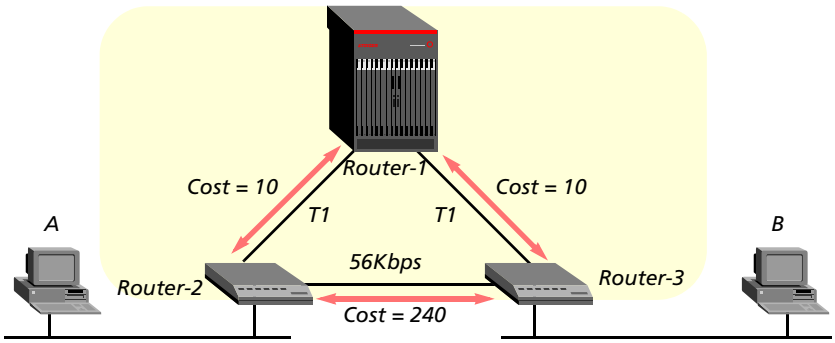
Configurable cost metrics

You assign a cost to the output side of each router interface. The lower the cost, the more likely the interface is to be used to forward data traffic. Costs can also be associated with the externally derived routing data.

The OSPF cost can also be used for preferred-path selection. If two paths to a destination have equal costs, you can assign a higher cost to one of the paths so it can be a backup to be used only when the primary path is not available.

Figure 7-2 shows how costs are used to direct traffic over high-speed links. For example, if Router-2 in Figure 7-2 receives packets destined for Host B, it routes them through Router-1 across two T1 links (`cost=20`) rather than across one 56Kbps B channel to Router-3 (`cost=240`).

Figure 7-2. OSPF costs for different types of links



The Stinger unit has a default cost of 1 for a connected route (Ethernet) and 10 for a WAN link. If two paths have the same destination, the Stinger unit uses the path with the lower cost unless route preferences change the equation.

When assigning costs, remember to account for the bandwidth of a connection. For example, for a single B-channel connection, the cost is 24 times greater than for a T1 link.

Note Be careful when assigning costs. Incorrect cost metrics can cause delays and congestion on the network.

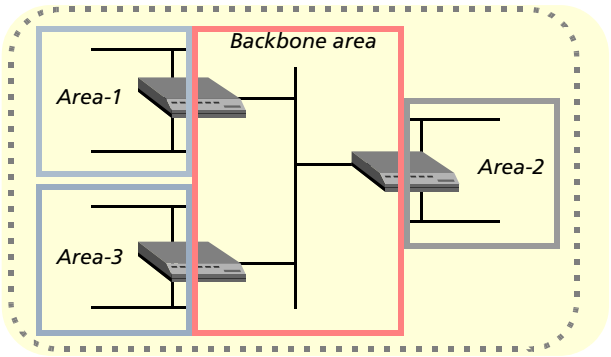
Hierarchical routing (areas)

If a network becomes too large, the size of the database, time required for route computation, and related network traffic become excessive. You can partition an autonomous system into areas to provide hierarchical routing, with a backbone area connecting the other areas. The backbone area is special and always has the area number 0.0.0.0. The backbone consists of networks not contained in any area, their attached routers, and routers that belong to multiple areas.

The backbone must be contiguous. You can use virtual links to connect two backbone routers that have an interface to a common nonbackbone area. OSPF treats two routers joined by a virtual link as if they were connected by an unnumbered point-to-point network. The backbone distributes routing information between areas and has all the properties of an area. The topology of the backbone is invisible to each of the areas, while the backbone itself has no information about area topology.

Each area acts as its own network: All area-specific routing information stays within the area, and all routers within an area must have a synchronized topological database. To tie the areas together, some routers belong to the backbone area and also to one of the other areas. These routers are area border routers (ABRs). In Figure 7-3, all the routers are ABRs.

Figure 7-3. Dividing an OSPF autonomous system into areas



With the ABRs and area boundaries set up correctly, link-state databases are unique to an area. You can configure the Stinger unit to route in normal areas, stub areas, and NSSAs. These different kinds of areas handle the autonomous system external (ASE) routes originated by ASBRs in the following ways.

- Normal areas

  An OSPF normal area allows type 5 LSAs to be flooded throughout the area.

- Stub areas

  Areas that are connected only to the backbone area by one ABR have one exit point and need not maintain information about external routes. To reduce the cost of routing, OSPF supports stub areas in which a default route summarizes all external routes. A stub area allows no type 5 LSAs to be propagated into or throughout the area, and instead depends on default routing to external destinations.

- NSSAs

  For details about the NSSA specification, see RFC 1587, *The OSPF NSSA Option.* NSSAs are like stub areas in that they do not receive or originate type 5 LSAs. However, NSSAs rely solely on default routing for external routes. They employ type 7 LSAs for carrying ASE route information within the area. Type 7 LSAs use a P-bit to flag the NSSA border router to translate the type 7 LSA into a type 5 LSA, which can then be propagated into other areas. When the Stinger unit is routing OSPF in an NSSA, it imports ASE routes defined in local or RADIUS profiles as type 7 LSAs. These imported ASE LSAs always have the P-bit enabled, which flags border routers to translate them into type 5 LSAs.

  You can list the router IDs of NSSA border routers that are translating type 7 LSAs to type 5 LSAs, by entering the `ospf translators` command. For example:

```
admin> ospf translators
Area ID     Router ID
0.0.0.1     10.105.0.13
0.0.0.2     12.1.1.1
```

Link-state routing algorithms

The link-state routing algorithms require that all routers within a domain maintain synchronized (identical) topological databases, and that the databases describe the complete topology of the domain. An OSPF router's domain can be an autonomous system or an area within an autonomous system.

OSPF routers create and update a link-state database from information exchanged with other routers. Link-state databases are synchronized between pairs of adjacent routers (as described in "Exchange of routing information" on page 7-3). In addition, each OSPF router uses its link-state database to calculate a self-rooted tree of shortest paths to all destinations. The routing table is built from these calculated shortest-path trees. For example, consider the network topology in Figure 7-4.

Figure 7-4. Sample OSPF topology

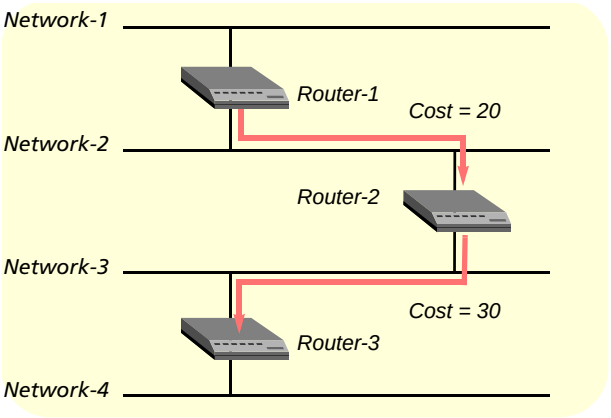


Table 7-2 shows the relevant information in the routers' link-state databases.

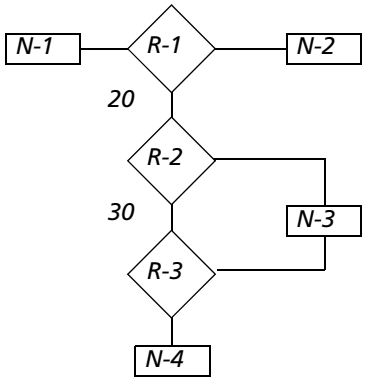
Table 7-2. Link-state databases for OSPF topology in Figure 7-4

Router-1	Router-2	Router-3
Network-1/Cost 0	Network-2/Cost 0	Network-3/Cost 0
Network-2/Cost 0	Network-3/Cost 0	Network-4/Cost 0
Router-2/Cost 20	Router-1/Cost 20	Router-2/Cost 30
	Router-3/Cost 30	

From the link-state database, each router builds a self-rooted shortest-path tree, and then calculates a routing table stating the shortest path to each destination in the autonomous system. (See Table 7-3, Table 7-4, and Table 7-5.) The table also includes externally derived routing information.

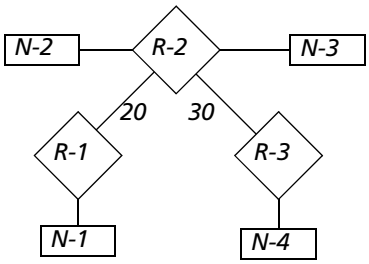
All the routers calculate a routing table of shortest paths, based on the link-state database. Externally derived routing data is advertised throughout the autonomous system but is kept separate from the link-state data. Each external route can also be tagged by the advertising router, enabling the passing of additional information between routers on the boundary of the autonomous system.

Table 7-3. Shortest-path tree and resulting routing table for Router-1



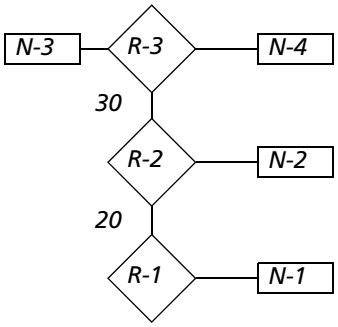
Destination	*Next hop*	*Metric*
Network-1	Direct	0
Network-2	Direct	0
Network-3	Router-2	20
Network-4	Router-2	50

Table 7-4. Shortest-path tree and resulting routing table for Router-2



Destination	*Next hop*	*Metric*
Network-1	Router-1	20
Network-2	Direct	0
Network-3	Direct	0
Network-4	Router-3	30

Table 7-5. Shortest-path tree and resulting routing table for Router-3



Destination	*Next hop*	*Metric*
Network-1	Router-2	50
Network-2	Router-2	30
Network-3	Direct	0
Network-4	Direct	0

Enabling OSPF systemwide

Before the Stinger IP2000 can route OSPF, it must be configured for IP routing. For details about configuring IP routing, see Chapter 4, "IP Routing Configuration."

To configure the system to use OSPF routing, you must configure each LAN or WAN interface that will support OSPF routing, and enable the protocol systemwide. The following parameters, shown with default values, enable the protocol and specify a few global settings:

```
[in IP-GLOBAL:ospf-global]
enable = no
```

```
as-boundary-router = yes
ospf-max-lsa = 0
```

Parameter	Setting
`enable`	Enables or disables the OSPF protocol systemwide. Set to `yes` to enable the protocol. If set to `no` (the default), the protocol is disabled systemwide. If you are modifying several OSPF-related profiles, you can use the `no` setting to prevent the OSPF subsystem from reinitializing whenever you write a modified profile. Then set the parameter to `yes` when the modifications are complete. A change to the setting takes effect immediately after you write the profile.
`as-boundary-router`	Enables or disables ASBR calculations related to external routes.
`ospf-max-lsa`	Maximum number of LSAs allowed in the link-state database. Specify a number from 0 through 4,294,967,295. The default setting is 0.

For example, the following commands enable the OSPF protocol:

```
admin> read ip-global
admin> set ospf-global enable = yes
admin> write -f
```

Configuring OSPF on Gigabit Ethernet

Before the Stinger IP2000 can route OSPF, it must be configured for IP routing. For details about configuring `ip-interface` profiles, see "Configuring ip-interface profiles for Ethernet ports" on page 4-5.

Overview of `ip-interface ospf` settings

Following are the OSPF parameters, shown with default values, for configuring OSPF routing on the controller's GigE interface:

```
[in IP-INTERFACE/{ { any-shelf any-slot 0 } 0 }:ospf]
active = no
area = 0.0.0.0
area-type = normal
hello-interval = 10
dead-interval = 40
priority = 5
authen-type = simple
auth-key = *******
key-id = 0
cost = 1
down-cost = 16777215
ase-type = type-1
ase-tag = c0:00:00:00
transit-delay = 1
```

```
retransmit-interval = 5
non-multicast = no
network-type = Broadcast
poll-interval = 10
profile-type = lan
md5-auth-key = *******
```

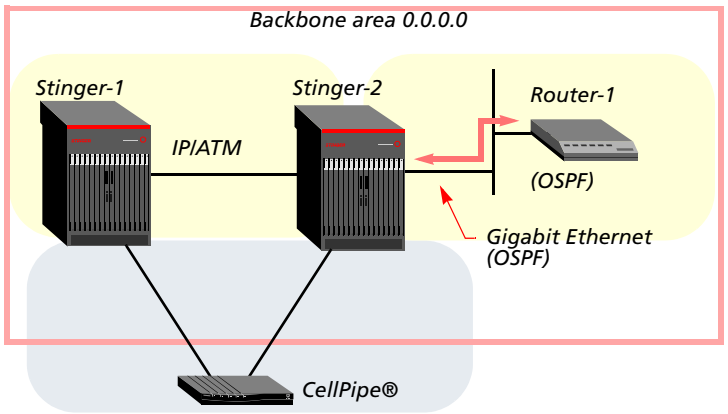
Parameter	Setting
active	Enables or disables OSPF on an interface.
area	OSPF area number in dotted decimal notation. The default 0.0.0.0 represents the backbone area.
area-type	Type of area. The default is the normal area type.
hello-interval	Number of seconds between Hello packets.
dead-interval	Number of seconds without receiving a Hello packet before instituting a link-state change.
priority	Priority value, from 0 to 255, used to elect a designated router and backup designated router. A setting of 0 excludes the Stinger from becoming a designated router or backup designated router. The higher the priority value of the Stinger IP2000 relative to other OSPF routers on the network, the better the chances that it will become one of these routers. For details, see "Designated and backup designated routers on broadcast networks" on page 7-4.
authen-type	Type of authentication to use. **none** — No authentication is required. **simple** — The router uses the password supplied in the auth-key parameter to validate OSPF packet exchanges. This is the default value. **md5** — The router uses MD5 encryption and the authentication key ID supplied by the key-id parameter to validate OSPF packet exchanges. For related information, see "Authentication" on page 7-2.
auth-key	Secret key for authenticating traffic in the router's area. Enter a text string of up to 8 characters. When authen-type is set to md5, you must set the md5-auth-key parameter to specify a key.
key-id	Number from 0 to 255, used to encrypt the secret key when authen-type is set to md5.
cost	Cost of routing to the interface. The lower the cost assigned to a route, the more likely it is to be used to forward traffic. For details, see "Configurable cost metrics" on page 7-5.

Parameter	Setting
`down-cost`	Cost applied to the interface when it is unavailable. The output cost when the link is physically unavailable but virtually active.
`ase-type`	Type of metric to apply to routes learned from RIP. This parameter applies in a `connection` profile only when OSPF is *not* active. `type-1` — Expresses the metric in the same units as the interface cost. This is the default. `type-2` — Metric is larger than any link-state path.
`ase-tag`	A hexadecimal number attached to an external route, set by default in Stinger systems to `c0:00:00:00`. This is not used by the OSPF protocol itself. It may be used to communicate information between AS boundary routers, and may appear in management utilities to indicate a route is external.
`transit-delay`	Estimated number of seconds required to transmit a Link State Update packet over this interface, taking into account transmission and propagation delays. On a connected route, you can leave the default of `1`.
`retransmit-interval`	Number of seconds between LSA retransmissions for adjacencies belonging to this interface. Its value is also used when retransmitting database description and link-state request packets. On a typical connected route, accept the default setting of `5`.
`non-multicast`	Supports NBMA configuration to a GRF® multigigabit router. See "Sample configuration of NBMA across point-to-point" on page 7-14.
`network-type`	Type of OSPF interface. `broadcast` — A broadcast-capable network, such as Ethernet. `nonbroadcast` — An NBMA network, such as a trunk interface. `point-to-point` — A point-to-point network, consisting of two routers only.
`poll-interval`	Not used on a broadcast network.
`profile-type`	A read-only parameter used internally to verify settings in the profile.
`md5-auth-key`	Secret key to be used for MD5 authentication, up to 16 characters. The default value is `ascend0`. When `authen-type` is set to `md5`, you must supply a value for this parameter.

Sample Gigabit Ethernet interface configuration

Figure 7-5 shows three OSPF routers in the backbone area of an autonomous system. Because all OSPF routers are in the same area, the routers form adjacencies and synchronize their databases. This example shows how to configure the Gigabit Ethernet interface of the unit labeled Stinger-2.

Figure 7-5. OSPF on a LAN interface



All OSPF routers in Figure 7-5 have RIP turned off. Running both RIP and OSPF is unnecessary, and turning RIP off reduces processor overhead. OSPF can learn routes from RIP interfaces, incorporate them in the routing table, assign them an external metric, and tag them as external routes.

Although RFC 2328 does not specify a limitation for the number of routers in the backbone area, keep the number of routers relatively small, because changes that occur in area zero are propagated throughout the autonomous system. Another way to configure the same units is to create a second area (such as 0.0.0.1) in one of the existing OSPF routers, and add the Stinger IP2000 to that area. You can then assign the same area number (0.0.0.1) to all OSPF routers reached through the Stinger IP2000 across a WAN link.

The following sample commands show how to configure `Stinger-2` in Figure 7-5. The commands assign the IP address 10.168.8.17/24 to the local interface and configure the OSPF router in the backbone area:

```
admin> read ip-interface { { 1 8 2 } 0 }
admin> set ip-address = 10.168.8.17/24
admin> set rip-mode = routing-off
admin> set ospf active = yes
admin> write -f
```

The following sample commands show how to configure the IP interface for MD5 authentication:

```
admin> read ip-interface { { 1 8 2 } 0 }
admin> set ospf authen-type = md5
admin> set ospf md5-auth-key = 12!secret*34key
admin> write -f
```

Configuring OSPF on an ATM trunk interface

Before the Stinger IP2000 can route OSPF, it must be configured for IP routing. For details about configuring `connection` profiles for IP routing, see "Configuring IPoA subscriber connections" on page 4-34. In Stinger IP2000 systems, OSPF routing across an ATM trunk interface uses a point-to-point link.

Overview of `connection ospf-options` settings

Following are the parameters, shown with default settings, for configuring OSPF on an ATM trunk interface.

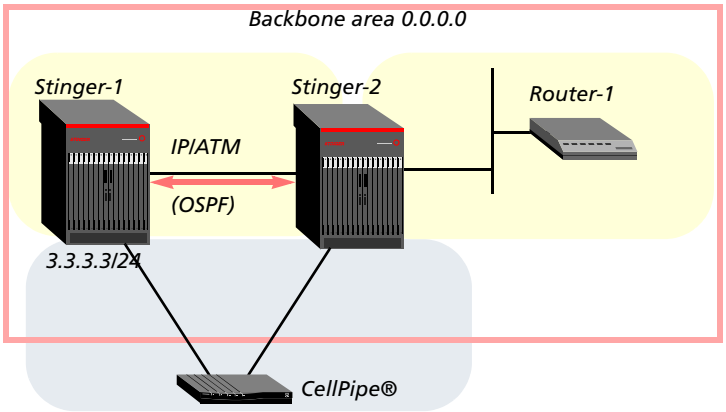
```
[in CONNECTION/"":ip-options:ospf-options]
active = no
area = 0.0.0.0
area-type = normal
hello-interval = 30
dead-interval = 120
priority = 5
authen-type = simple
auth-key = *******
key-id = 0
cost = 10
down-cost = 1000
ase-type = type-1
ase-tag = c0:00:00:00
transit-delay = 1
retransmit-interval = 5
non-multicast = no
network-type = Point-to-Point
poll-interval = 10
profile-type = wan
md5-auth-key = *******
```

These are the same parameters described for enabling OSPF on the Gigabit Ethernet interface. For definitions, see "Overview of ip-interface ospf settings" on page 7-9, or the *Stinger Reference*.

Sample OSPF point-to-point configuration

This example shows how to configure a `connection` profile in the system labeled Stinger-2 in Figure 3-6, to enable it to route OSPF across the ATM cloud to Stinger-1. In this example, the unit labeled Stinger-1 uses the IP address 3.3.3.3/24.

Figure 7-6. OSPF over ATM point to point



The following commands configure OSPF in the unit labeled Stinger-2 in Figure 7-6:

```
admin> read conn stinger1-atmvc
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 3.3.3.3/24
admin> set ip-options rip = routing-off
admin> set ip-options ospf active = yes
admin> set atm-options atm1483type = aal5-llc
admin> set atm-options vci = 101
admin> set atm-options nailed-group = 802
admin> write -f
```

Sample configuration of NBMA across point-to-point

With the current software version, NBMA is supported only on a point-to-point WAN link. Full support for multiaccess is not provided.

Overview of additional NBMA settings

In addition to the standard settings for OSPF point-to-point, the following parameters, shown with default settings, must be configured to support NBMA:

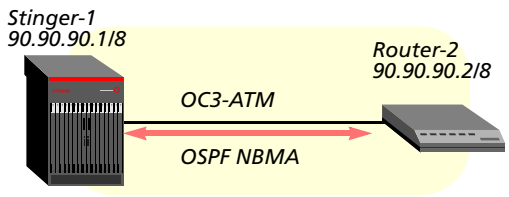
```
[in CONNECTION/"":ip-options:ospf-options]
non-multicast = no
network-type = Point-to-Point

[in CONNECTION/"":ip-options]
local-address = 0.0.0.0/0

[in OSPF-NBMA-NEIGHBOR/""]
name* = ""
host-name = ""
ip-address = 0.0.0.0
dr-capable = no
```

Parameter	Setting
non-multicast	For an NBMA connection to a GRF® multigigabit router, the non-multicast parameter must be set to yes. This causes the translation of the multicast traffic to directed traffic. This setting is required only when connecting to a GRF® multigigabit router.
network-type	For NBMA, the network-type parameter must be set to NonBroadcast.
local-ip-address	When network-type is set to NonBroadcast, a local-ip-address value must be provided or the system reports a configuration error such as the following: OSPF CONFIG ERROR: NBMA profile router-2 has illegal 0.0.0.0 address
ospf-nbma-neighbor: name	Name of the ospf-nbma-neighbor profile.
ospf-nbma-neighbor: host-name	Name of the local connection profile that defines the connection to the neighboring router.
ospf-nbma-neighbor: ip-address	IP address of the neighboring router.
ospf-nbma-neighbor: dr-capable	Whether the neighboring router can be the designated router (yes or no).

Example of an NBMA configuration

Figure 7-7 shows a Stinger IP2000 connecting point-to-point with another router that operates in an OSPF NBMA network.

Figure 7-7. OSPF NBMA over ATM point to point



Following is a connection profile on the system labeled Stinger-1 to enable NBMA on the link to Router-2:

```
admin> new connection router-2
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 90.90.90.2/8
admin> set ip-options local-address = 90.90.90.1/8
admin> set ip-options ospf-options active = yes
admin> set ip-options ospf-options authen-type = simple
admin> set ip-options ospf-options auth-key = mykey
```

```
admin> set ip-options ospf-options network-type = NonBroadcast
admin> set telco-options nailed-groups = 851
admin> set mp-options enabled = no
admin> set atm-options atm1483type = aal5-llc
admin> set atm-options vci = 101
admin> set atm-options nailed-group = 851
admin> write -f
```

The following profile enables the unit to form an adjacency with Router-2:

```
admin> new ospf-nbma-neighbor router-2
admin> set host-name = router-2
admin> set ip-address = 90.90.90.2/8
admin> write -f
```

Configuring global route options that apply to OSPF

The `ip-global`profile contains several settings that apply only when OSPF routing is in use. Following are the relevant parameters, shown here with their default settings:

```
[in IP-GLOBAL]
pool-ospf-adv-type = type-1
ospf-pref = 10
ospf-ase-pref = 150
ospf-global = { no yes 0 }
rip-tag = c8:00:00:00
rip-ase-type = 1
```

Parameter	Setting
`pool-ospf-adv-type`	Type of ASE metric applied to summarized pools imported into OSPF as external routes.
`ospf-pref`	Preference value for routes learned from OSPF. Valid values are 0 to 255. The default value is 10.
`ospf-ase-pref`	Preference value for routes learned from RIP, ICMP, or another non-OSPF protocol. Specify a value from 0 through 255. By default, routes learned dynamically from another routing protocol are assigned a preference value of 150.
`rip-tag`	Hexadecimal number associated with routes learned from RIP. OSPF border routers can use the tag to filter a record.
`rip-ase-type`	Type of ASE metric applied to routes learned from RIP.

Example of importing a summarized pool as an ASE

The following commands configure a summarized pool and import it to OSPF with a type 1 OSPF metric:

```
admin> read ip-global
```

```
admin> set pool-summary = yes
admin> set pool-base-address 1 = 10.12.253.1
admin> set assign-count 1 = 62
admin> set pool-ospf-adv-type = type-1
admin> write -f
```

When pool-summary is set to yes and OSPF is enabled, the OSPF subsystem uses the pool-ospf-adv-type parameter to determine how to import summarized routes into OSPF. If this parameter is set to type-1, the metric for the route to a summarized pool is expressed in the same units as the link-state metric (interface cost).

If pool-ospf-adv-type is set to type-2, the unit considers the routing between autonomous systems as the major cost of routing a packet, and conversion of external costs to internal link-state metrics is unnecessary. If the parameter is set to internal, the summarized pool addresses are imported into OSPF as intra-area routes, which enables them to work properly with stub areas.

Example of setting ASE preferences

The ospf-pref and ospf-ase-pref parameters determine the preference values assigned to routes learned from other OSPF routers and those imported from other dynamic routing protocols. The default settings place a much lower preference on OSPF routes, which means that the routes learned from other protocols (ASE routes) are more likely to be used. The following commands decrease to 100 the preference assigned to ASE routes (the default is 150):

```
admin> read ip-global
admin> set ospf-ase-pref = 100
admin> write -f
```

Configuring ip-route OSPF options

For details about configuring static routes, see "Configuring ip-route profiles" on page 4-25. The following parameters in the ip-route profile (shown with sample settings), apply only when OSPF is enabled:

```
in IP-ROUTE/""]
cost = 1
ase-type = type-1
ase-tag = c0:00:00:00
ase7-adv = N/A
```

Parameter	Setting
cost	Cost of routing to the interface. The lower the cost, the more likely the interface is to be used to forward traffic. See "Configurable cost metrics" on page 7-5.
ase-type	Type of metric to apply to routes learned from RIP. The default value of type-1 expresses the metric in the same units as the interface cost. With the value of type-2, the metric is larger than any link-state path.

Parameter	Setting
ase-tag	Hexadecimal number that appears in management utilities and flags this route as external. It can also be used by border routers to filter this record.
ase7-adv	*Currently not used.*

Example of configuring a type 7 LSA in an NSSA

For background information about NSSAs, see "Hierarchical routing (areas)" on page 7-5. To configure the Stinger unit to route OSPF in an NSSA, *all* OSPF interfaces in the Stinger IP2000 must specify the NSSA area-type.

To configure a type 7 LSA, you must specify a static route in an ip-route profile. Following are the related parameters (shown with sample settings):

```
[in IP-ROUTE/external]
name* = external
dest-address = 10.4.5.0/22
gateway-address = 10.4.5.7
metric = 0
cost = 1
preference = 100
third-party = no
ase-type = type-1
ase-tag = c0:00:00:00
private-route = no
active-route = yes
ase7-adv = n/a
```

The following procedure configures the Stinger unit to route in an NSSA and import a type 7 LSA that specifies an external route across the WAN link:

1 Assign an NSSA area type to the IP interface that is running OSPF. For example:

```
admin> read ip-interface { { 1 8 2 } 0 }
admin> set ospf area-type = nssa
admin> write -f
```

2 Configure the WAN link that represents an ASE route. For example:

```
admin> read connection ase-link
admin> set ip-options remote = 10.4.5.7/22
admin> set ip-options rip = routing-off
admin> set ip-options ospf active = yes
admin> write -f
```

3 Configure a static route to the remote site. For example:

```
admin> new ip-route type7
admin> set dest = 10.4.5.0/22
admin> set gateway = 10.4.5.7
admin> write -f
```

Example of assigning a cost to a static route

The lower the cost assigned to a route, the more likely the router is to choose the route to forward traffic. Typically, you account for the bandwidth of a connection when assigning costs. The Stinger unit has a default cost of 1 for a connected route (Ethernet) and 10 for an ATM VC. If two paths have the same destination, the Stinger unit uses the path with the lower cost.

Note Be careful when assigning costs. Incorrect cost metrics can cause delays and congestion on the network.

In the following example, an administrator assigns a cost of 25 to a static route:

```
admin> new ip-route mylink
admin> set dest = 10.1.2.0/24
admin> set gateway = 10.9.8.10
admin> set cost = 25
admin> write -f
```

Administrative tools for OSPF routing

The `ospf` diagnostic-level commands display information related to OSPF routing, including the link state advertisements (LSAs); the routing table for border routers; and the OSPF areas, interfaces, statistics, and routing table. For details, see the chapter on monitoring OSPF in the *Stinger Adminstration Guide*.

IP Multicast Configuration

8

IP multicast forwarding is supported with an optional software license. Enter the following command to determine whether the multicast license is enabled:

```
admin> get base igmp
[in BASE]
igmp-np-enabled = yes
```

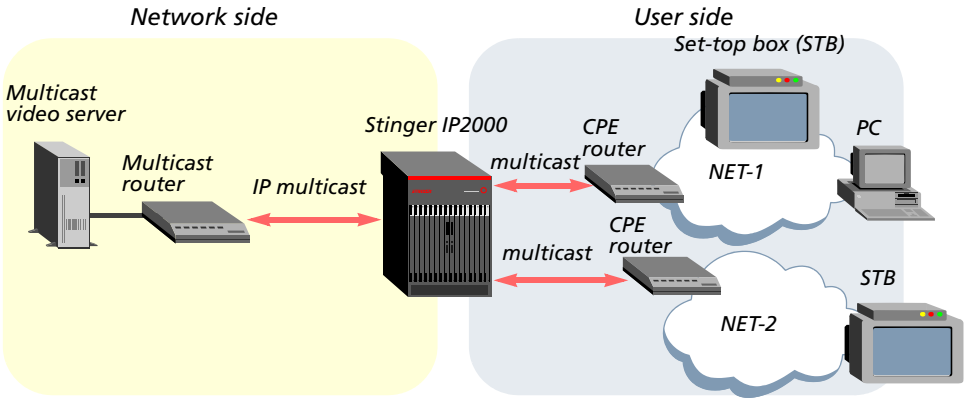
The system sets this parameter to `yes` when the IGMP license is enabled for the IP2000 network processor. If the license is not enabled, the system displays an error message if you configure multicast forwarding on the IP2000. For information about obtaining and enabling Lucent Technologies software licenses, contact your Lucent sales representative.

IP multicast forwarding

IP multicast forwarding enables the Stinger to receive multicast transmissions from multicast backbone (MBONE) routers and forward the transmissions to multiple client interfaces.

A common use for IP multicast is to transmit streaming video across the Internet to applications running on multiple PCs or set-top boxes (STB) for television sets, as shown in Figure 8-1.

Figure 8-1. Multicast video sample setup



To the multicast clients, the Stinger appears to be a multicast router originating the video stream. To the multicast routers, the Stinger appears to be a multicast client, initiating and responding to group management messages via Internet Group Management Protocol (IGMP) version-1 or version-2.

To receive a transmission, the client interfaces must join a specific multicast group. A *multicast group* is a Class D IP address (from 224.0.0.0 to 239.255.255.255). When data is sent to an address in that range, it is multicast to all hosts that have joined that group. The Stinger forwards IGMP messages between clients (hosts) and the multicast router to enable these transactions.

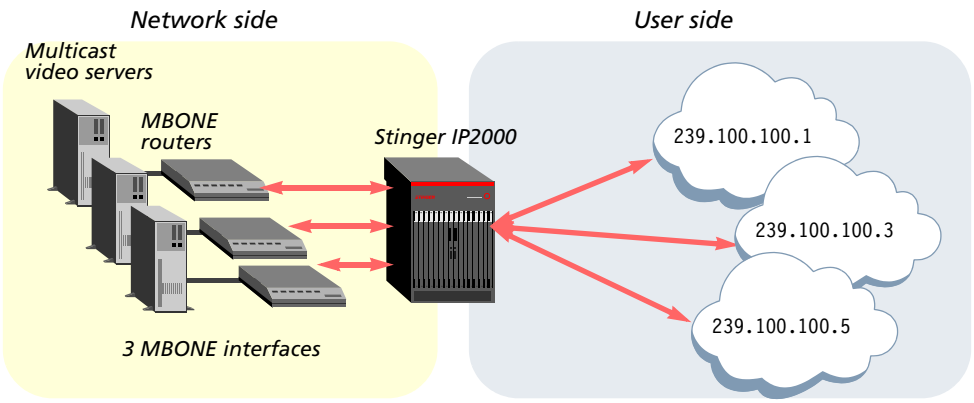
Network-side MBONE interfaces

An MBONE interface, used to receive transmissions from a multicast router, can be either a WAN MBONE on an ATM trunk interface or a LAN MBONE on the IP2000 Gigabit Ethernet interface. On an MBONE interface, the Stinger responds to multicast router queries, and returns responses from clients to the router.

The global IP router can receive multicast data from up to four MBONE interfaces. Multiple MBONE capability is not currently supported for virtual routers.

The multiple MBONE interfaces can be WAN (ATM trunks) or LAN, or any combination of the two interface types, but the total number of MBONE interfaces cannot exceed four. Figure 8-2 shows a Stinger unit with three MBONE interfaces:

Figure 8-2. Multiple MBONE interfaces on trunk or LAN interfaces



Notice about Gigabit Ethernet redundancy for a LAN MBONE

For Stinger systems with redundant controllers, you can configure Gigabit Ethernet redundancy for a LAN MBONE interface to enable the system to maintain MBONE operations across a controller switchover. For details, see "Configuring a redundant LAN MBONE" on page 2-8.

LIM-side multicast client interfaces

The multicast clients must be on LIM interfaces, accessing the Stinger unit through an ATM virtual circuit.

Transmission of multicast data to multicast clients on a local Ethernet interface is not currently supported. For that reason, the following parameters not currently used in the `ip-interface` profile and its `igmp-option` subprofile for the IP2000:

Table 8-1. Unused multicast client settings for LAN interfaces

Unused `ip-interface` settings

`[in IP-INTERFACE/{ { any-shelf any-slot 0 } 0 }]`
`multicast-rate-limit = 100`
`multicast-group-leave-delay = 0`
`multicast-group-leave-delay-msec = 0`
`multicast-service-profile = ""`
`multicast-max-groups = 0`

`[in IP-INTERFACE/{ { any-shelf any-slot 0 } 0 }:igmp-options]`
`robust-count = 2`
`query-interval = 125`
`query-response-interval = 100`
`last-member-query-interval = 10`
`last-member-query-count = 2`

Configuring MBONE interfaces

To configure an MBONE interface, you must complete the following steps:

1. Enable multicast forwarding in the `ip-global` profile.
2. Specify a profile index for each MBONE interface in the `ip-global` profile.
3. Configure the `ip-interface` or `connection` profile for each LAN or WAN MBONE interface. Be sure to set `multicast-allowed` to `yes`.

A Stinger IP2000 does not support multicast heartbeat monitoring, so the following `ip-global` settings are not used:

Table 8-2. Unused multicast heartbeat monitoring settings

Unused `ip-global` settings

`[in IP-GLOBAL]`
`multicast-hbeat-addr = 0.0.0.0`
`multicast-hbeat-port = 0`
`multicast-hbeat-slot-time = 0`
`multicast-hbeat-Number-Slot = 0`
`multicast-hbeat-Alarm-threshold = 0`
`multicast-hbeat-src-addr = 0.0.0.0`
`multicast-hbeat-src-addr-mask = 0.0.0.0`

If you need more information about these settings, see the parameter descriptions in the *Stinger Reference*.

Overview of multiple MBONE configuration

The following parameters, shown with their default settings, are used to specify from MBONE interfaces:

```
[in IP-GLOBAL]
multicast-forwarding = no

[in IP-GLOBAL:multiple-mbone:mbone-profile]
mbone-profile[1] = ""
mbone-profile[2] = ""
mbone-profile[3] = ""
mbone-profile[4] = ""

[in IP-GLOBAL:multiple-mbone:mbone-lan-interface]
mbone-lan-interface[1] = { { any-shelf any-slot 0 } 0 }
mbone-lan-interface[2] = { { any-shelf any-slot 0 } 0 }
mbone-lan-interface[3] = { { any-shelf any-slot 0 } 0 }
mbone-lan-interface[4] = { { any-shelf any-slot 0 } 0 }

[in IP-INTERFACE/{ { shelf-1 first-control-module 2 } 0 } ]
multicast-allowed = no

[in CONNECTION/"":ip-options]
multicast-allowed = no
```

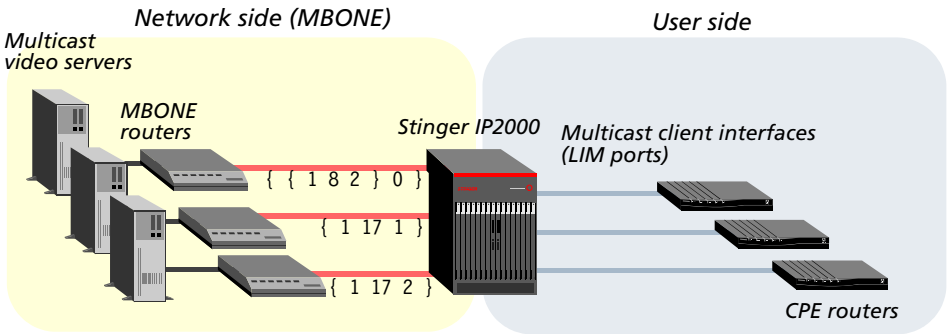
Parameter	Setting
`multicast-forwarding`	Enables or disables multicast forwarding. When you change the value to `yes` and write the profile, the multicast subsystem reads the values in the `ip-global` profile and initiates the forwarding function.

Parameter	Setting
`mbone-profile[N]`	Array of four indexed parameters for specifying the name of a local `connection` profile that provides access to an MBONE router across a trunk interface. This configures up to four WAN MBONE interfaces across ATM trunk ports. The total number of MBONE interfaces specified in either these parameters or the `mbone-lan-interface` parameters, or both, cannot exceed four.
`mbone-lan-interface[N]`	Array of four indexed parameters for specifying the index of an `ip-interface` profile that provides access to an MBONE router across an Ethernet interface. This configures up to four LAN MBONE interfaces across the Gigabit Ethernet port of the IP2000 controller or Ethernet interfaces of T1000 modules. For Stinger systems with redundant controllers, you can configure Gigabit Ethernet redundancy for a LAN MBONE interface to enable the system to maintain MBONE operations across a controller switchover. For details, see "Configuring a redundant LAN MBONE" on page 2-8. The total number of MBONE interfaces specified in either these parameters or the `mbone-profile` parameters, or both, cannot exceed four.
`multicast-allowed`	Enables or disables handling of IGMP requests and responses on the LAN (`ip-interface`) or trunk (`connection`) MBONE interface.

Sample configuration with multiple MBONE interfaces

The sample setup in Figure 8-3 shows three MBONE interfaces, one on the Gigabit Ethernet interface ({ { 1 8 2 } 0 }) and two additional MBONE interfaces on the two trunk interfaces in slot 17.

Figure 8-3. Sample configuration with multiple MBONE interfaces



The following commands configure a LAN MBONE interface on the Gigabit Ethernet port of the IP2000 controller:

```
admin> read ip-interface { { 1 8 2 } 0 }
admin> set ip-address = 1.1.1.2/28
admin> set multicast-allowed = yes
admin> write -f
```

The next commands configure two WAN MBONE interfaces on ATM trunk interfaces:

```
admin> new connection mcast1-17-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 3.3.3.3/29
admin> set ip-options multicast-allowed = yes
admin> set atm-options vci = 101
admin> set atm-options nailed-group = 801
admin write -f
admin> new connection mcast1-17-2
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 4.4.4.4/29
admin> set ip-options multicast-allowed = yes
admin> set atm-options vci = 101
admin> set atm-options nailed-group = 802
admin write -f
```

The following commands enable the multicast forwarding function and specify the MBONE interfaces:

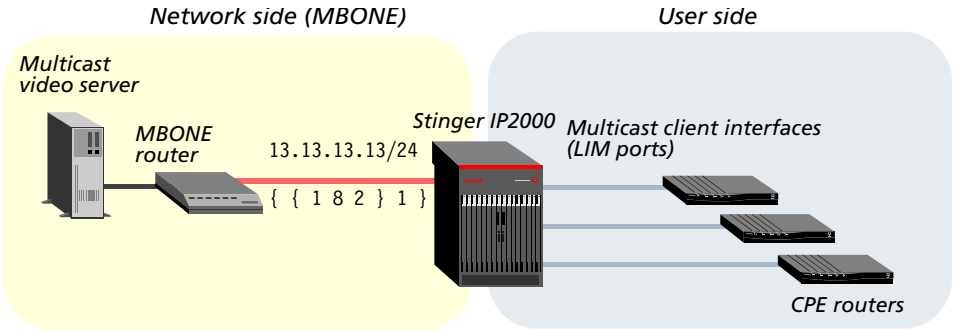
```
admin> read ip-global
admin> set multicast-forwarding = yes
admin> set multiple-mbone mbone-profile 1 = mcast1-17-1
admin> set multiple-mbone mbone-profile 2 = mcast1-17-2
admin> set multiple-mbone mbone-lan-interface 1 = { { 1 8 2 } 0 }
admin> write -f
```

Sample MBONE configuration on Gigabit Ethernet VLANs

You can set up a virtual IP interface on the Gigabit Ethernet port as a virtual LAN (VLAN) to be used for routing purposes as well as for handling multicast traffic. For information about the VLAN configuration, see "Configuring routed VLANs" on page 3-20.

Figure 8-4 shows a basic representation of this type of configuration, which enables the system to handle a multicast data stream with a VLAN tag, and forward the multicast data stream to multicast clients.

Figure 8-4. Sample configuration of VLAN MBONE interface



Following is a sample Gigabit Ethernet VLAN MBONE interface configuration.

1 Enable bridging on the physical interface.

```
admin> read ethernet { 1 8 2 }
admin> set bridging-enabled = yes
admin> write -f
```

2 Enable multicast forwarding in the `ip-global` profile.

```
admin> read ip-global
admin> set multicast-forwarding = yes
admin> write -f
```

3 Create a `vlan-ethernet` profile. This example uses VLAN ID 101.

```
admin> new vlan-ethernet { { 1 8 2 } 101}
admin> set enabled = yes
admin> write -f
```

4 Create an `ip-interface` profile with both multicast and VLAN enabled. Specify the same VLAN ID used in the `vlan-ethernet` profile (101 in this example).

```
admin> new ip-interface { { 1 8 2 } 1 }
admin> set ip-address = 13.13.13.13/24
admin> set multicast-allowed = yes
admin> set vlan-enabled = yes
admin> set vlan-id = 101
admin> write -f
```

Note If you enable multicast without enabling VLAN, the system refuses to write the profile and displays the following error message:

```
Multicast can be enabled on a Logical IP interface only when VLAN is
enabled
```

5 Configure the new multicast and VLAN enabled interface as an MBONE interface in the `ip-global` profile.

```
admin> read ip-global
admin> set multiple-mbone mbone-lan 1 physical-address shelf = 1
admin> set multiple-mbone mbone-lan 1 physical-address slot = 8
admin> set multiple-mbone mbone-lan 1 physical-address item = 2
```

```
admin> set multiple-mbone mbone-lan 1 logical-item = 1
admin> write -f
```

Managing multicast group memberships

To receive a multicast transmission, a client interface must join a specific multicast group. A multicast group is a class D IP address range from 224.0.0.0 to 239.255.255.255. Multicast groups from 224.0.0.100 through 224.0.0.160 are reserved for internal use. When data is sent to a valid multicast group, it is multicast to all hosts that have joined that group.

The `mcast-service` profile provides a way to manage which multicast groups can be accessed by a client interface. Each profile specifies a number of multicast groups (up to 256), and a filter type. The filter type determines how the list of multicast groups is used: to allow access only to those groups, or allow access to all groups *except* those listed.

You can configure multiple `mcast-service` profiles, one for each level of multicast services you provide. You can define the profiles locally or via RADIUS. The number of active `mcast-service` profiles that can be provisioned is limited only by memory considerations in the system.

For examples of applying `mcast-service` profiles to a client interface, see "Configuring multicast client interfaces" on page 8-11.

Number of multicast clients per group

A Stinger IP2000 allows up to 1017 multicast clients per group. If the number of multicast clients in a group exceeds 1017, a message such as the following is displayed:

```
LOG info, Shelf 1, Controller-1, Time: 17:07:07--
  We hit maximum number of client per multicast group for 239.100.100.1
```

Overview of `mcast-service` settings

Following are the `mcast-service` settings, shown with default values:

```
[in MCAST-SERVICE/""]
service-name* = ""
active = no
filter-type = none
filter-list = [ { no 0.0.0.0 1 } { no 0.0.0.0 1 } { no 0.0.0.0 1} { no 0.0.0+
[in MCAST-SERVICE/"":filter-list[1]]
active = no
mcast-ip-address = 0.0.0.0
group-range-count = 1
```

Parameter	Setting
`service-name`	Name assigned to the service profile, up to 31 characters.

Parameter	Setting
`active`	Enables or disables the profile. If the profile is disabled, none of the clients on the interfaces whose profiles point to this service profile are allowed to join any multicast groups. If you want the opposite effect, to allow clients on those interfaces access to all multicast groups, use the `filter-type` setting instead.
`filter-type`	Specifies whether access to the multicast groups defined in the filter list will be filtered inclusively or exclusively. With inclusive filtering, client interfaces have access only to those groups specified in the filter list. With exclusive filtering, clients have access to all multicast groups *except* those in the list. If you set this to `none`, access to all multicast groups is allowed.
`filter-list[`*n*`]: active`	The filter list contains 256 indexed subprofiles, each of which specifies a multicast group address filter. The `active` parameter enables or disables the filter. When the filter is enabled, access to the address or address range specified in the filter is controlled on the basis of the `filter-type` setting. If the filter is disabled, it is not used.
`filter-list[`*n*`]: mcast-ip-address`	Class D IP address from 224.0.0.0 to 239.255.255.254. When data is sent to this address, it is multicast to all hosts that have joined that group. WThe address 239.255.255.255 cannot currently be used as a valid multicast address in a filter rule.
`filter-list[`*n*`]: group-range-count`	The number of group addresses in a range of contiguous addresses that begins with the value of the `mcast-ip-address` parameter. Valid values are from 1 to 1024. The default value is 1, which indicates the single address specified as the `mcast-ip-address`. For example, if `mcast-ip-address` is set to `234.1.1.1` and `group-range-count` is set to `5`, the filter affects addresses `234.1.1.1` through `234.1.1.5`. Wraparound is permitted; for example, if `mcast-ip-address` is set to `234.1.1.230` and `group-range-count` is set to `37`, the filter affects groups `234.1.1.230` through `234.1.2.10`.

Sample multicast address filters

In this example, a multicast video server supports two multicast group addresses. Multicast clients can subscribe to either the “bronze” or the “gold” multicast service. Bronze service permits access to the group at 239.255.129.119. Gold service permits access to both 239.255.129.119 and a premium group at 239.255.129.120.

The following commands configure the `mcast-service` profiles:

```
admin> new mcast-service bronze-service

admin> set active = yes

admin> set filter-type = inclusive
```

```
admin> set filter-list 1 active = yes
admin> set filter-list 1 mcast-ip-address = 239.255.129.119
admin> write -f
admin> new mcast-service gold-service
admin> set active = yes
admin> set filter-type = inclusive
admin> set filter-list 1 active = yes
admin> set filter-list 1 mcast-ip-address = 239.225.129.119
admin> set filter-list 2 active = yes
admin> set filter-list 2 mcast-ip-address = 239.255.129.120
admin> write -f
admin> dir mcast-service
   802  07/24/2004 20:12:09  bronze-service
   809  07/24/2004 20:13:19  gold-service
```

The following command displays information about the multicast service profiles:

```
admin> igmp profiles
IGMP Service Profiles

          Service Name               : gold-service
          Filter Type                : MCAST_FILTER_INCLUSIVE
          Filter List                :
                 224.255.129.120
                 224.225.129.119

          Service Name               : bronze-service
          Filter Type                : MCAST_FILTER_INCLUSIVE
          Filter List                :
                 224.255.129.119
```

Sample multicast address range filter

In this example, the administrator wants to allow a user access to a set of group addresses from 234.1.1.1 to 234.1.1.5. The following commands configure the addresses in an `mcast-service` profile:

```
admin> new mcast-service test_234_1
admin> set active = yes
admin> set filter-type = inclusive
admin> set filter-list 1 active = yes
admin> set filter-list 1 mcast-ip-address = 234.1.1.1
admin> set filter-list 1 group-range-count = 5
admin> write -f
```

The following commands configure the user's `connection` profile to use the new `mcast-service` profile:

```
admin> new connection tg61
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 10.10.7.10/24
admin> set ip-options multicast-allowed = yes
admin> set ip-options multicast-service-profile = test_234_1
admin> set atm-options nailed-group = 251
admin> write -f
```

If the configuration results in an out-of-range multicast address, or if the same group address is specified in two `filter-list` subprofiles, the Stinger logs an error message.

Configuring multicast client interfaces

A multicast client interface is an IP-enabled ATM PVC across a DSL interface to a CPE router such as a CellPipe®.

Overview of multicast client `ip-options` settings

In addition to the ATM and IP options needed to configure a terminating PVC, the following parameters, shown with default values, are used to enable IP multicast on a client interface:

```
[in CONNECTION/"":ip-options]
multicast-allowed = no
multicast-rate-limit = 100
multicast-group-leave-delay = 0
multicast-group-leave-delay-msec = 0
multicast-max-groups = 0
fast-leave = no
multicast-server-vc = no
multiple-mcast-filter = [ "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "" "+]
```

Parameter	Setting
`multicast-allowed`	Enables or disables handling of IGMP requests and responses on the interface. The system does *not* forward multicast traffic on the basis of this setting.

Parameter	Setting
`multicast-rate-limit`	The rate at which the system accepts multicast packets from clients on the interface. For example, if you set the rate to 5, the system accepts one packet every 5 seconds from multicast clients on the interface, and discards subsequent packets received within that 5-second window. The default value of 100 disables multicast forwarding on the interface—the Stinger acts as a forwarder and handles IGMP packets, but does not accept packets from clients or forward multicast packets from the MBONE router. To enable multicast forwarding on the interface, you must set the rate to a number *less than* 100.
`multicast-group-leave-delay`	Number of seconds to delay before forwarding a Leave Group message. The sum of (`multicast-group-leave-delay` × 1000) plus `multicast-group-leave-delay-msec` is the number of milliseconds the system waits before forwarding to the MBONE router an IGMP version-2 Leave Group message it receives across a multicast client interface. With the zero default values, the system forwards the Leave Group messages immediately. For client interfaces that support multiple multicast sessions to the same group, set these parameters to a value from 10 to 20. When these parameters have nonzero values, the system sends back a query to make sure there are no active multicast sessions on the interface for that group, and if it receives a response before the specified, specified delay expires, it does not forward the Leave Group message.
`multicast-group-leave-delay-msec`	Number of milliseconds to add to the value of (`multicast-group-leave-delay` × 1000) to determine the total delay before forwarding a Leave Group message.
`multicast-max-groups`	Maximum number of accessible multicast group (from 0 to 250) for this client interface. You can set this to a lower number to limit multicast traffic to the interface. This value of this parameter limits the activation of new client links, but does not affect the existing client links.

Parameter	Setting
`fast-leave`	Enables or disables the fast leave feature, which bypasses the Group Leave delay. When set to `no` (the default), after a Leave Group message is received from the last group member, the system delays the Group Leave process to run a last member query, to ensure that the host is the last member. If a host sends a host membership report in response to the query, the Group Leave process is canceled and the host membership interval timer is started. The IP multicast traffic does not stop until the last host member timer expires. When set to `yes`, the system seizes the IP multicast traffic on an interface or sends Group Leave message to upstream router on MBONE interface as soon as the last member host leaves the multicast group. IGMP v2 or later is required for proper operation. This feature assumes that individual hosts are sending membership reports on receiving the general or group membership query, as is the case for well-known STBs. You can view active member interfaces by using the `igmp hosts` command on a LIM. See "IGMP diagnostics" on page A-4.
`multicast-server-vc`	Specifies whether the virtual circuit is a multicast server VC (`yes` or `no`). See "Overview of multicast server VC settings" on page 8-22.
`multiple-mcast-filter`	An array of 20 indexed subprofiles, to enable you to apply up to 20 `mcast-service` profiles per user. The profiles are applied in the order in which they are specified. Once a group address is found in an `mcast-service` profile in either the include list or exclude list, it is applied. If at least one multicast filter exists and is active, groups that are not in the include list are not allowed. If no `mcast-service` profiles are configured or none of the profiles are active, then all groups are allowed to join.

Setting IGMP-v2 timers (local profiles only)

The system calculates the timeout value for IGMPv1 and IGMPv2 members using the rule described in RFC 2236 *Internet Group Management Protocol, Version 2*. The rule is as follows:

`Timeout-value` = (`query-interval` x `robust-count` + `query-response-interval/10`)

You can control the timeout value by setting the `query-interval`, `robust-count`, or `query-response-interval` parameters for an IGMPv1 or IGMPv2 member.

The following parameters, shown with default values, are used to configure the timers defined in RFC 2236 on multicast client interfaces:

```
[in CONNECTION/"":ip-options:igmp-options]
robust-count = 2
query-interval = 125
query-response-interval = 100
last-member-query-interval = 10
last-member-query-count = 2
```

Parameter	Setting
`robust-count`	A threshold of packet losses (from 2 to 10) up to which the multicast subsystem will remain robust. If the interface is expected to have a high rate of packet loss, increase this value. IGMP is robust to (`robust-count` minus 1) packet losses. It cannot be set to zero and should not be set to 1. The default is 2.
`query-interval`	Number of seconds (from 0 to 1024) between general queries. You can increase this value from its default of 125 seconds to reduce the number of IGMP queries sent on the interface.
`query-response-interval`	Maximum response time in tenths of a second (from 0 to 1024) inserted into general queries. You can increase this value from its default of 10 seconds to make IGMP traffic less bursty, because host responses will be spread out over a larger interval. The number of seconds response time (this value divided by 10) must be less than the `query-interval` value.
`last-member-query-interval`	Maximum response time in tenths of a second (from 0 to 1024) inserted into group-specific queries sent in response to Leave Group messages. You can reduce this value from its default of 1 second to reduce the time it takes to detect that the last member of a group has left. The response time (this value divided by 10) must be less than the `query-interval` value.
`last-member-query-count`	Number of group-specific queries sent before the multicast router assumes there are no local members.

Example of using multiple multicast filters

In this example, an administrator defines one `mcast-service` profile for a basic set of 50 channels to be allowed to a subscriber, and another `mcast-service` profile for each available PPV event. Then, to allow a user to see a PPV event, the administrator adds the filter for the PPV event to the user's `connection` profile. The existing filter for the basic set of channels remains active and unchanged in the user's profile, and multiple PPV-related filters can be applied in the same profile as appropriate.

The following set of commands defines an `mcast-service` profile for a basic set of channels:

```
admin> new mcast-service
admin> set service-name = 50-channels
admin> set active = yes
admin> set filter-type = inclusive
admin> set filter-list 1 active = yes
admin> set filter-list 1 mcast-ip-address = 224.10.10.10
admin> set filter-list 1 group-range-count = 50
admin> write -f
```

The following set of commands defines an `mcast-service` profile for a PPV event:

```
admin> new mcast-service
admin> set service-name = ppv-07302004
admin> set active = yes
admin> set filter-type = inclusive
admin> set filter-list 1 active = yes
admin> set filter-list 1 mcast-ip-address = 226.10.10.10
admin> set filter-list 1 group-range-count = 1
admin> write -f
```

The following set of commands defines a `connection` profile and applies both filters:

```
admin> new connection mpoa-adsl
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 110.110.110.52/8
admin> set ip-options netmask-remote = 255.0.0.0
admin> set ip-options multicast-allowed = yes
admin> set ip-options multicast-max-groups = 250
admin> set ip-options multiple-mcast-filter 1 = 50-channels
admin> set ip-options multiple-mcast-filter 2 = ppv-07302004
admin> set atm-options nailed-group = 604
admin> write -f
```

The following commands displays information about the `mcast-service` profiles:

```
admin> igmp profiles
IGMP Service Profiles

Service Name               : 50-channels
Filter Type                : MCAST_FILTER_INCLUSIVE
Filter List( Group Range) :
224.10.10.10 - 224.10.10.59  ( Count = 50 )

Service Name               : ppv-7302004
```

```
Filter Type                  : MCAST_FILTER_INCLUSIVE
Filter List( Group Range) :
226.10.10.10 - 226.10.10.10  ( Count = 1 )
```

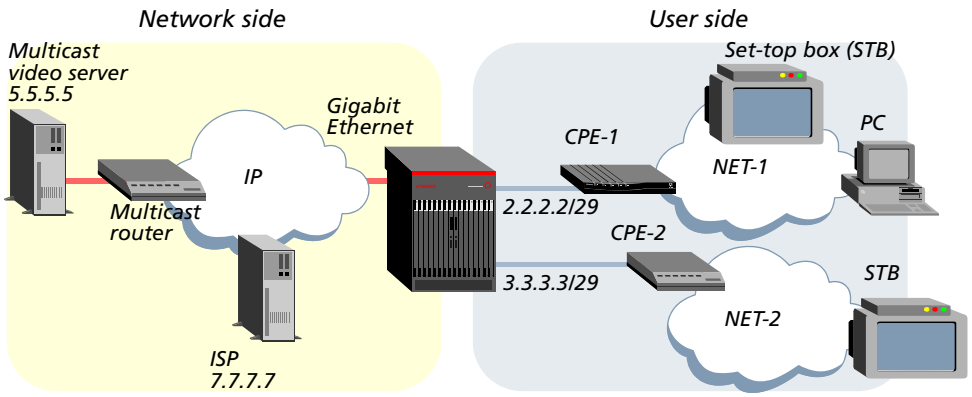
Sample multicast video configuration with filters

In this sample setup, the MBONE interface is the Gigabit Ethernet port of the IP2000. A multicast router on the local IP network sends video transmissions from a multicast video server. In the sample configurations for the setup shown in Figure 8-5, two types of filters are used:

- `mcast-service` profiles, to filter multicast groups and related services
- `filter` profiles, to filter out generic IP data

By introducing generic IP data filters, you can cause the system's IP interfaces to handle only multicast video services and block end users from accessing other services in the network.

Figure 8-5. DSL video application with a local MBONE interface



Configuring the local MBONE interface

The following commands enable the MBONE interface on the Gigabit Ethernet port:

```
admin> read ip-global
admin> set multicast-forwarding = yes
admin> set mbone-lan-interface 1 = { { 1 8 2 } 0 }
admin> write -f
```

Configuring multicast client PVCs

The profiles in this section apply `mcast-service` profiles to filter multicast groups and their related services. For background information, see "Managing multicast group memberships" on page 8-8.

The following commands configure a PVC for CPE-1 in Figure 8-5:

```
admin> new connection cpe-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 2.2.2.2/29
```

```
admin> set ip-options multicast-allowed = yes
admin> set ip-options multicast-rate-limit = 20
admin> set ip-options multicast-service-profile = bronze-service
admin> set ip-options multicast-max-groups = 1
admin> set atm-options vci = 100
admin> set atm-options nailed-group = 51
admin> write -f
```

The next set of commands configures the CPE-2 client interface in Figure 8-5:

```
admin> new connection cpe-2
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 3.3.3.3/29
admin> set ip-options multicast-allowed = yes
admin> set ip-options multicast-rate-limit = 20
admin> set ip-options multicast-service-profile = gold-service
admin> set ip-options multicast-max-groups = 2
admin> set atm-options vci = 100
admin> set atm-options nailed-group = 52
admin> write -f
```

Applying a filter that restricts the GigE interface to video traffic only

In this example, a filter is applied to the GigE interface to prevent the interface from handling traffic other than video data and related control messages. This filter has the advantage of being easy to define and apply, but it restricts the GigE interface from being used for other applications.

For background information about filters, see Chapter 10, "Filter Configuration."

The following commands create a new `filter` profile named `mcast-only`, and specify an input filter that forwards inbound multicast data traffic (239.100.100.0):

```
admin> new filter mcast-only
admin> set input-filters 1 valid-entry = yes
admin> set input-filters 1 forward = yes
admin> set input-filters 1 Type = ip-filter
admin> set input-filters 1 ip-filter source-address-mask = 255.255.255.0
admin> set input-filters 1 ip-filter source-address = 239.100.100.0
```

The following commands specify an output filter that allows client access to the video server at IP address 5.5.5.5 (Figure 8-5). Clients might require access to the server, for example, to download boot information for a set-top box.

```
admin> set output-filters 1 valid-entry = yes
admin> set output-filters 1 forward = yes
admin> set output-filters 1 Type = ip-filter
admin> set output-filters 1 ip-filter dest-address-mask = 255.255.255.255
```

```
admin> set output-filters 1 ip-filter dest-address = 5.5.5.5
```

Because the default setting of the `forward` parameter is `no`, the next set of commands explicitly drops all other output traffic:

```
admin> set output-filters 2 valid-entry = yes
admin> set output-filters 2 Type = ip-filter
admin> write -f
```

The following commands apply the `mcast-only` filter to the Gigabit Ethernet interface of the IP2000:

```
admin> read ethernet { 1 8 2 }
admin> set filter-name = mcast-only
admin> write -f
```

An alternative filter to restrict each client interface

If you must use the Gigabit Ethernet interface for other applications as well as multicast video, you cannot restrict the type of traffic allowed on the interface. In that case, you can define filters for individual multicast client interfaces, to restrict the those interfaces from handling traffic other than video data and related control messages. Individual users on the client interface will be unable to access other services through the Stinger.

Note The IP2000 does not currently support filters specified in RADIUS profiles.

This sample filter is specific to the multicast client network connected to CPE-1 in Figure 8-5 (page 8-16), with the CPE IP address 2.2.2.2/29.

The following commands create a new `filter` profile named `conn-input-filter`, and specify an input filter that forwards inbound traffic to the video server at IP address 5.5.5.5 (Figure 8-5). Clients might require access to the server, for example, to download boot information for a set-top box.

```
admin> new filter conn-input-filter
admin> set input-filters 1 valid-entry = yes
admin> set input-filters 1 forward = yes
admin> set input-filters 1 Type = ip-filter
admin> set input-filters 1 ip-filter dest-address-mask = 255.255.255.255
admin> set input-filters 1 ip-filter dest-address = 5.5.5.5
```

The next set of commands specifies an input filter that allows only inbound traffic that uses the IGMP protocol (protocol number 2) from a client on the subnet 2.2.2.*x*.

```
admin> set input-filters 2 valid-entry = yes
admin> set input-filters 2 forward = yes
admin> set input-filters 2 Type = ip-filter
admin> set input-filters 2 ip-filter protocol = 2
admin> set input-filters 2 ip-filter source-address-mask = 255.255.255.0
admin> set input-filters 2 ip-filter source-address = 2.2.2.0
```

Because the default setting of the `forward` parameter is `no`, the next set of commands explicitly drops all other inbound traffic from the client subnet:

```
admin> set input-filters 3 valid-entry = yes
admin> set input-filters 3 Type = ip-filter
admin> write -f
```

For the CPE-1 `connection` profile definition, see "Configuring multicast client PVCs" on page 8-16. The following commands apply this filter to the individual client interface for CPE-1:

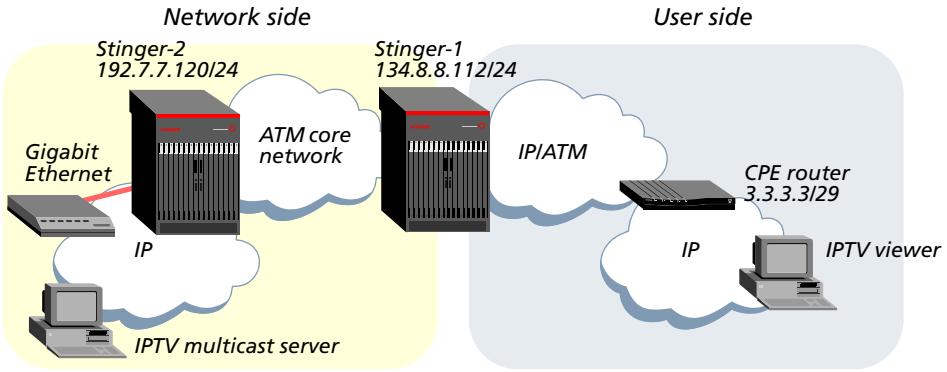
```
admin> read connection cpe-1
admin> set session-options data-filter = conn-input-filter
admin> write -f
```

Sample multicast video configuration with a remote MBONE interface

In the sample setup shown in Figure 8-6, the MBONE interface is configured in Stinger-2, and the multicast client interface is configured in Stinger-1.

The connection between the two Stinger units is an ATM PVC. It can use any ATM medium, and does not use IP processing. In this example, Stinger-2 sends the IPTV multicast data stream across the ATM cloud as a cell stream. Stinger-1 forwards the ATM stream to the multicast CPE router on the basis of an ATM circuit configuration.

Figure 8-6. IPTV video sample configuration



Stinger-1 requires an ATM circuit profile between the LIM interface to the CPE router, and the trunk interface to Stinger-2.

The following command on Stinger-1 displays the nailed-group number of the ADSL interface (LIM slot 4, port 5) to the CPE router in Figure 8-6:

```
admin> which -n { 1 4 5 }
Nailed group corresponding to port { shelf-1 slot-4 5 } is 155
```

The following command on Stinger-1 displays the nailed-group number of the OC3-ATM interface (trunk slot 17, port 2) that connects to the ATM core network in Figure 8-6:

```
admin> which -n { 1 17 2 }
Nailed group corresponding to port { shelf-1 trunk-module-1 2 } is 802
```

The following set of commands on Stinger-1 configures an ATM circuit between the two interfaces:

```
admin> new connection mcast-client-pvc
admin> set active = yes
admin> set encapsulation-protocol = atm-circuit
admin> set ip-options ip-routing-enabled = no
admin> set atm-options nailed-group = 155
admin> set atm-connect-options vci = 100
admin> set atm-connect-options nailed-group = 802
admin> write -f
```

Following is a comparable RADIUS profile:

```
permconn-st-2 Password = "pwd"
    Service-Type = Outbound,
    Framed-Protocol = ATM-CIR,
    User-Name = "mcast-client-pvc",
    Ascend-ATM-Group = 155,
    Ascend-Route-IP = Route-IP-No,
    Ascend-ATM-Vpi = 0,
    Ascend-ATM-Vci = 35,
    Ascend-ATM-Connect-Vpi = 0,
    Ascend-ATM-Connect-Vci = 100,
    Ascend-ATM-Connect-Group = 802
```

With this `connection` or RADIUS profile, ATM cells received by Stinger-1 from the CPE router are switched to the unit's trunk interface and transmitted across the ATM cloud. On Stinger-2, the PVC terminates and is packetized for transmission on the IP network.

The following command on Stinger-2 displays the nailed-group number of a DS3-ATM interface (trunk slot 17, port 1) that connects to the ATM core network in Figure 8-6:

```
admin> which -n { 1 17 1 }
Nailed group corresponding to port { shelf-1 trunk-module-1 1 } is 801
```

The following set of commands on Stinger-2 terminates the PVC on a trunk interface with VCI 100, and specifies the required IP address for Stinger-1.

```
admin> new connection term-18-100
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 3.3.3.3/29
admin> set atm-options vci = 100
admin> set atm-options nailed-group = 801
admin> write -f
```

Following is a comparable RADIUS profile:

```
permconn-st-1 Password = "pwd"
    Service-Type = Outbound,
    Framed-Protocol = ATM-1483,
    User-Name = "term-18-100",
    Framed-IP-Address = 3.3.3.3,
    Framed-IP-Netmask = 255.255.255.252,
    Ascend-ATM-Group = 801,
    Ascend-Route-IP = Route-IP-Yes,
    Ascend-ATM-Vpi = 0,
    Ascend-ATM-Vci = 100
```

After completing the configuration, starting the IPTV client software on the multicast client interface should create the 239.100.100.4 multicast group on Stinger-2. The following command on Stinger-2 checks that the group exists:

```
admin> igmp groups
Group Address          Members      Expire time       Counts
230.0.0.9              14           00:00:31      0 :: 0 S2
                       *(Mbone)                   0 :: 0 S2
```

The following command displays client interfaces (interface 14 representing the remote client interface):

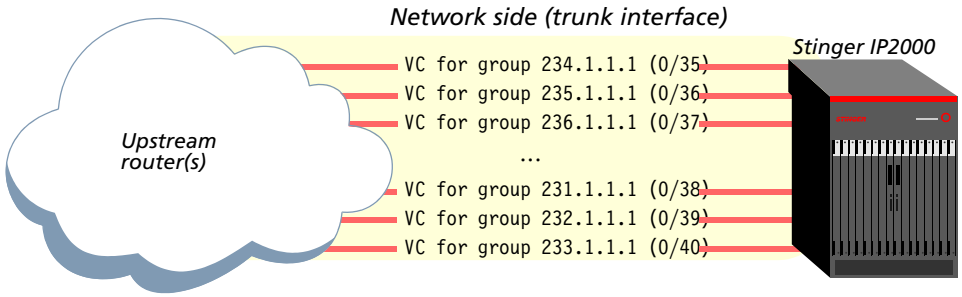
```
admin> igmp clients
   IGMP Clients
    Client             Version         RecvCount       CLU     ALU
     1(Mbone)            2                 0            0       0
     14                  2                 0            0       0
```

Multicast server virtual circuits

With this software version, you can configure up to 100 Stinger trunk PVCs as *multicast server virtual circuits (VCs)*. A multicast server VC is a a trunk terminating PVC that carries the multicast data stream of one multicast group.

The administrator of an upstream router maps a PVC to a particular multicast group address and ensures that only one multicast group is mapped to one VC, as shown in Figure 8-7. The Stinger system does not need to know which group address is carried on each VC to establish the connection.

Figure 8-7. One VC per multicast group for incoming multicast data streams



When the multicast server VC has been established in the Stinger system, it is registered as an MBONE interface. The VC acts as an MBONE interface for IGMP

joins coming from the DSL side. However, IGMP control messages are not exchanged across the multicast server VC. The system simply forwards multicast traffic received on the multicast server VC to all DSL clients who have joined the associated multicast group.

Note When `multiple-multicast-server-vc` is configured in the `ip-global` profile, you cannot also configure MBONE or PIM.

Overview of multicast server VC settings

Following are the parameters, shown with default settings, required for configuring multicast server VCs:

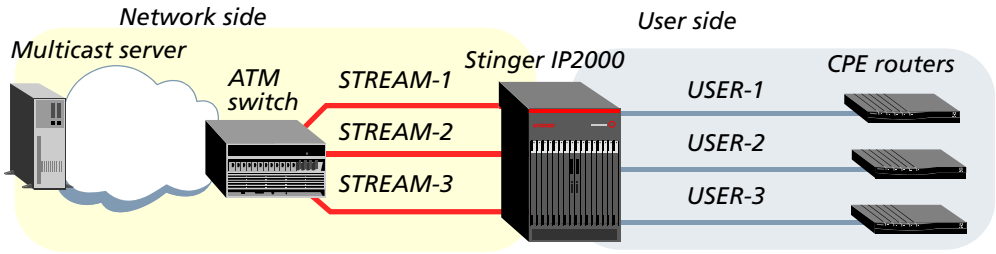
```
[in IP-GLOBAL]
multicast-forwarding = no
multiple-multicast-server-vc = no

[in CONNECTION/"":ip-options]
multicast-allowed = no
multicast-server-vc = no
```

Parameter	Setting
`multicast-forwarding`	Enables or disables multicast forwarding. When you change the value to `yes` and write the profile, the multicast subsystem reads the values in the `ip-global` profile and initiates the forwarding function. Set to `yes` for the multicast server VC feature.
`multiple-multicast-server-vc`	Enables or disables the multicast server VC feature systemwide. With the default `no` setting, the feature is disabled. Set it to `yes` to enable the feature. When this parameter is set to `yes`, you can configure up to 100 trunk PVCs Stinger to receive multicast data from one multicast group. When this parameter is set to `yes`, you cannot configure MBONE or PIM.
`multicast-allowed`	Enables or disables the handling of IGMP requests and responses on trunk MBONE interface.
`multicast-server-vc`	Specifies whether the virtual circuit is a multicast server VC (`yes` or `no`). A multicast server VC is a trunk terminating PVC that carries the multicast data stream of one multicast group. It acts as an MBONE for IGMP joins coming from the DSL side, and the system forwards multicast traffic from the multicast server VC to DSL clients who have joined the group. However, IGMP control messages are not exchanged on the interface. When this parameter is set to `yes`, the `encapsulation-protocol` parameter must be set to `atm`. The `multiple-multicast-server-vc` parameter in the `ip-global` profile must be set to `yes`.

Sample configuration of multicast server VCs

In the sample setup shown in Figure 8-8, `STREAM-1`, `STREAM-2`, `STREAM-3` are trunk terminated ATM PVCs configured as multicast server VCs, and `user-1`, `user-2`, and `user-3` are DSL user connections.

Figure 8-8. Multicast server VCs on a trunk interface



In this example, the Stinger system is receiving the multicast data stream of one multicast group on each of the three trunk-terminating PVCs. An upstream router (not shown in this example) is responsible for streaming multicast traffic on each VC.

Enabling the multicast server VC feature

The following commands on the Stinger system enable the multicast server VC feature systemwide:

```
admin> read ip-global
admin> set multicast-forwarding = yes
admin> set multiple-multicast-server-vc = yes
admin> write -f
```

Configuring the multicast server VCs

The following commands create the `STREAM-1`, `STREAM-2`, `STREAM-3` trunk-terminated ATM PVCs shown in Figure 8-8. The `multicast-server-vc` setting must be enabled in the VC `connection` profiles, and `encapsulation-protocol` must be set to `atm`.

```
admin> new connection STREAM-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 20.20.20.1/8
admin> set ip-options multicast-allowed = yes
admin> set ip-options multicast-server-vc = yes
admin> set atm-options vci = 40
admin> set atm-options nailed-group = 802
admin> write -f
admin> new connection STREAM-2
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 20.20.20.2/8
```

```
admin> set ip-options multicast-allowed = yes
admin> set ip-options multicast-server-vc = yes
admin> set atm-options vci = 41
admin> set atm-options nailed-group = 802
admin> write -f
admin> new connection STREAM-3
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 20.20.20.3/8
 admin> set ip-options multicast-allowed = yes
admin> set ip-options multicast-server-vc = yes
 admin> set atm-options vci = 43
admin> set atm-options nailed-group = 802
admin> write -f
```

Configuring the terminating connections for DSL users

Following is a sample terminating `connection` profile for the connection labeled `user-1` in Figure 8-8:

```
admin> new connection user-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 122.122.122.122/8
admin> set ip-options local-address = 111.111.111.111/8
admin> set ip-options multicast-allowed = yes
admin> set atm-options vci = 59
admin> set atm-options nailed-group = 505
admin> write -f
```

Administrative tools for IGMP operations

The system supports the `igmp` commands for administrative information about IGMP multicast operations. For examples, see "IGMP diagnostics" on page A-4.

PIM-SM v2 Configuration

9

Traditional multicast routing mechanisms, such as Distance Vector Multicast Routing Protocol (DVMRP) or Multicast OSPF (MOSPF) are intended for use in regions where groups are widely represented or bandwidth is universally plentiful. If these traditional schemes are used when multicast receivers and senders are distributed sparsely across a wide area, data packets and membership reports are sent over many links that do not lead to receivers or senders. For this reason, the traditional schemes can be inefficient for use in wide area networks.

Protocol Independent Multicast Sparse Mode (PIM-SM v2) is designed to operate efficiently across wide area networks, where groups are sparsely distributed. Each multicast group has a shared tree through which receivers learn about ("rendezvous with") sources. The rendezvous point (RP) is the root of this per-group shared tree. PIM SM uses RPs and explicit join/prune messages instead of the broadcast and prune mechanism used by PIM-Dense Mode or DVMRP.

This implementation of PIM-SM follows the current IETF drafts for PIM-SM v2 (`draft-ietf-pim-sm-v2-new-07.txt`, March 2003) and candidate bootstrap router (C-BSR) functionality (`draft-ietf-pim-sm-bsr-03.txt`, February 2003).

Note Stinger units support full interoperability with RFC 2362-compliant multicast routers that have not yet implemented the PIM-SM v2 IETF draft recommendations. However, Stinger units do not support interoperability with systems running PIM-SM v1. In addition, PIM-SM cannot be used with multiple MBONE interfaces in a Stinger unit.

PIM-SM features supported with this software version

With this software version, PIM-SM is supported on IP2000 Gigabit Ethernet and trunk interfaces. Stinger units cannot currently operate as PIM Multicast Border Router (PMBR). With the current software, Stinger units support PIM-SM functionality as shown in Table 9-1:

Table 9-1. Current level of support for PIM-SM functionality

PIM-SM capability	Support in this software version
PIM-SM general purpose states	(*,G) state
PIM join/prune messages	Join/prune messages for the (*,G) PIM-SM state
Bootstrap router (BSR)	Stinger units can be configured to act as C-BSR and take part in the BSR election process.
	If the Stinger becomes the elected BSR, it sends bootstrap messages (BSMs) to 224.0.0.13 on its PIM-enabled interfaces.
	If a PIM neighbor comes up and Stinger is the designated router on that interface, Stinger sends the recent BSM to the neighbor.
Designated routers (DR) and hello messages	On each PIM-enabled interface, the Stinger unit can be elected DR on the basis of hello priority.
	Currently, Stinger units do not support register messages and do not have the capability of processing IGMP messages on the LAN interface. For that reason, the Stinger unit must not be elected DR on the LAN interface if the LAN supports IGMP hosts.
RP-group mapping	Static configuration of active group range and their respective RPs.
	Dynamic RP-group mapping using the BSR router mechanism.
Data packet forwarding	Switch to SPT is not supported.
	Data packets are not forwarded from one PIM interface to another. Data packets flow from PIM interfaces to the users only.

Overview of PIM-SM configuration

To enable the system to act as a multicast router in a PIM domain, you must complete the following steps:

1 Enable multicast forwarding and the PIM protocol. If appropriate, specify a C-BSR configuration for the Stinger to participate in BSR elections and act as BSR if elected. See "Enabling multicast and PIM" on page 9-3.

2 Configure static mappings between multicast groups and PIM RPs. This is recommended as a failsafe configuration. See "Configuring static mappings between groups and rendezvous points" on page 9-5.

3 Configure an `ip-interface` or `connection` profile, to enable the system to operate as a PIM router on the Gigabit Ethernet interface, a trunk interface, or both. See "Configuring PIM on Gigabit Ethernet or trunk interfaces" on page 9-6.

For examples that show a system configuration that includes all three steps, see "Sample PIM-SM system configuration" on page 9-10.

Enabling multicast and PIM

A PIM BSR is a dynamically elected router within a PIM domain that is responsible for constructing the set of RPs and originating BSMs. A C-BSR is a PIM router configured to participate in the BSR election and to act as BSR if elected. One BSR is elected per PIM domain on the basis of highest priority and address.

For details about the `pim bsr` and `pim rp` commands, see "Administrative tools for PIM-SM routing" on page 9-11.

Overview of settings in the `ip-global` profile

Following are the parameters, shown with default values, for enabling PIM-SM and configuring its BSR capabilities in a Stinger unit:

```
[in IP-GLOBAL]
multicast-forwarding = no

[in IP-GLOBAL:pim-options]
enable = no
cbsr-enable = no
cbsr-ip-address = 0.0.0.0
cbsr-priority = 0
cbsr-interval = 60
```

Parameter	Setting
`multicast-forwarding`	Enables or disables multicast forwarding. This parameter must be set to `yes` for PIM-SM operations. When you change the value to `yes` and write the profile, the multicast subsystem reads the values in the `ip-global` profile and initiates the forwarding function.
`enable`	Enables or disables the PIM routing protocol systemwide. This parameter and the `multicast-forwarding` parameter must be set to `yes` to enable PIM.
`cbsr-enable`	Enables or disables the BSR router mechanism. When set to `yes`, the Stinger unit acts as a candidate BSR and takes part in electing a BSR in the PIM domain. With the `yes` setting, you must specify a `cbsr-ip-address` value. This setting is not used when `enable` is set to `no`.
`cbsr-ip-address`	Local IP address the Stinger unit uses to send BSMs when `cbsr-enable` is set to `yes`. This setting is not used when `cbsr-enable` is set to `no`.
`cbsr-priority`	BSR priority for the Stinger, from 0 (the default) to 255. The priority is used in the election of BSR. The system is more likely to be elected BSR with a higher priority value. To enable the system to exchange BSMs without becoming BSR, leave the default zero setting, or set a low numeric value. This setting is not used when `cbsr-enable` is set to `no`.

Parameter	Setting
cbsr-interval	Number of seconds, from 5 to 900, between transmission of BSMs. The default is to send BSMs every 60 seconds. This setting is not used when cbsr-enable is set to no.

Example showing BSR election and dynamic group-RP mappings

The following commands configure the Stinger unit to act as C-BSR using the IP address of the Gigabit Ethernet interface (1.1.1.101 in this example):

```
admin> read ip-global
admin> set multicast-forwarding = yes
admin> set pim-options enable = yes
admin> set pim-options cbsr-enable = yes
admin> set pim-options cbsr-ip-address = 1.1.1.101
admin> write -f
```

The following command displays the BSR status immediately after writing the ip-global profile:

```
admin> pim bsr
Stinger BSR State : PENDING_BSR
Details of CURRENT BSR:
BSR IP Address       : 0.0.0.0
BSR Interface        : 0
BSR Priority         : 0
BSR holdtime         : 0
BSR Current Frag Tag : 0
BSR HASH masklen     : 0
```

After the bootstrap interval has elapsed and the system has received a BSM from another router in the domain, a repeat of the pim bsr command shows that the Stinger has become candidate BSR. For example:

```
admin> pim bsr
Stinger BSR State : CANDIDATE_BSR
Details of CURRENT BSR:
BSR IP Address       : 1.1.1.10
BSR Interface        : 1
BSR Priority         : 110
BSR holdtime         : 112
BSR Current Frag Tag : 0
BSR HASH masklen     : 30
```

The following commands modify the ip-global profile to specify the highest BSR priority for the Stinger system:

```
admin> set cbsr-priority = 255
admin> write -f
```

Following an exchange of BSMs, the Stinger is elected BSR. For example:

```
admin> pim bsr
Stinger BSR State : ELECTED_BSR
```

```
Details of CURRENT BSR:
BSR IP Address       : 1.1.1.101
BSR Interface        : 1
BSR Priority         : 255
BSR holdtime         : 57
BSR Current Frag Tag : 717
BSR HASH masklen     : 30
```

The system is also receiving group-RP mappings dynamically from other routers. For example, the following command displays both static (if any) and dynamic group-RP mappings:

```
admin> pim rp
Group           RP-Address      RPF neighbor    Priority   holdtime
224.0.0.0/8       1.1.1.3         -                40        75:73
234.0.0.0/8       1.1.1.3         -                40        75:73
234.3.0.0/16      1.1.1.3         -                40        75:73
234.4.0.0/16      1.1.1.3         -                40        75:73
234.5.0.0/16      1.1.1.3         -                40        75:73
234.6.0.0/16      1.1.1.3         -                40        75:73
Static Entries:
225.0.0.0/8       1.1.1.10        1.1.1.10         -         -
235.0.0.0/8       1.1.1.10        1.1.1.10         -         -
```

Configuring static mappings between groups and rendezvous points

For PIM-SM to operate properly, all routers in the domain must share the same set of group-to-RP mappings. You can statically configure associations between groups and RPs by using the `pim-group-rp-mapping` profile. These static mappings provide a basic interoperability mechanism if the automatic methods of obtaining mappings should fail.

Each `pim-group-rp-mapping` profile specifies a mapping between an RP (specified as a reachable IP address) and a range of multicast groups (specified as a group and mask). The system uses these mappings to determine an RP for a given group.

Following are the parameters, shown with their default settings, for configuring a static group-to-RP mapping:

```
[in PIM-GROUP-RP-MAPPING/""]
name* = ""
rp-address = 0.0.0.0
group-address = 0.0.0.0/0
group-mask = 0.0.0.0
```

Parameter	Setting
`name`	Text string, up to 31 characters long, that names the mapping between a multicast group and the IP address of a rendezvous point (RP).
`rp-address`	IP address of the RP. The address must be reachable throughout the domain.

Parameter	Setting
group-address	A multicast group address (a class D IP address). You can specify a full group address or a group range. If you specify a prefix (such as /8 in the value 226.0.0.0/8), the system automatically updates the group-mask parameter with the appropriate decimal value (such as 255.0.0.0). The combined group address and group mask must be unique in the system. You cannot write duplicate mappings for the same group or group range.
group-mask	A mask to be applied to the group-address value to obtain the group prefix mapped to the specified RP. For example, a value of 255.0.0.0 indicates a one-octet group prefix. If no mask is specified, the default mask of 255.255.255.255 is applied.

The following commands create a static mapping for multicast group 231.1.1.1:

```
admin> new pim-group-rp-mapping 231
admin> set rp-address = 1.1.1.3
admin> set group-address = 231.1.1.1
admin> write -f
```

The next command shows the static mapping in the group-RP set for the system:

```
admin> pim rp
Group           RP-Address      RPF neighbor    Priority   holdtime
Static Entries:
231.1.1.1/32    1.1.1.3         -               -           -
```

Configuring PIM on Gigabit Ethernet or trunk interfaces

The pim-options subprofiles in an ip-interface or connection profile specify settings to enable PIM-SM, and specify the handling of PIM hello and join/prune messages on the interface.

All PIM routers send hello messages periodically on each PIM-enabled interface, and record the hello information received from each PIM neighbor. Hello messages allow a router to learn about the neighboring PIM routers on the interface, and the priority field in these messages is used in DR election on the LAN interface.

PIM options in the **ip-interface** and **connection** profiles

Following are the parameters, shown with default values, for enabling PIM on the Gigabit Ethernet interface. The listing for trunk interface follows.

```
[in IP-INTERFACE/{ { shelf-1 first-control-module 2 } 0 }]
multicast-allowed = no
[in IP-INTERFACE/{ { shelf-1 first-control-module 2 } 0 }:pim-options]
enable = no
hello-interval = 30
hello-holdtime = 105
hello-priority-option = yes
```

```
hello-priority = 1
join-prune-interval = 60
join-prune-holdtime = 210
lan-delay-option = yes
lan-delay = 5000
override-interval = 2500
```

Note You cannot enable PIM on a virtual Ethernet interface.

Following are the parameters, shown with default values, for configuring PIM on a trunk interface:

```
[in CONNECTION/""]
encapsulation-protocol = atm-circuit

[in CONNECTION/"":atm-options]
atm1483type = aal5-llc
vpi = 0
vci = 35
nailed-group = 1

[in CONNECTION/"":ip-options]
ip-routing-enabled = yes
remote-address = 0.0.0.0/0
local-address = 0.0.0.0
multicast-allowed = no

[in CONNECTION/"":ip-options:pim-options]
enable = no
hello-interval = 30
hello-holdtime = 105
hello-priority-option = yes
hello-priority = 1
join-prune-interval = 60
join-prune-holdtime = 210
lan-delay-option = yes
lan-delay = 5000
override-interval = 2500
```

Parameter	Setting
`encapsulation-protocol`	Set to `atm` for MPOA terminating connections.
`atm1483type`	Stinger systems support the two encapsulation methods for carrying routed PDUs in the payload field of ATM adaptation layer (AAL) type 5, which are defined in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*. The `aal5llc` setting indicates LLC encapsulation, which is used for non-PPP terminated connections such as RFC 2684 PVCs that terminate in the system. For PPP connections (PPPoE and PPPoA), the `aal5vc` setting is used, which indicates AAL5 VC multiplexing.
`vpi`	VPI for the MPOA terminating PVC. For details, see the *Stinger ATM Configuration Guide*.

Parameter	Setting
`vci`	VCI for the MPOA terminating PVC. For details, see the *Stinger ATM Configuration Guide*.
`nailed-group`	Nailed-group number of the trunk interface. For details, see the *Stinger ATM Configuration Guide*.
`ip-routing-enabled`	IP routing must be enabled (as it is by default) for MPOA terminating connections.
`remote-address`	IP address of the remote device, which can include a subnet specification. If it does not include a subnet mask, the router software in the Stinger unit assumes a default subnet mask that is based on address class.
`local-address`	IP address assigned to the local side of a numbered-interface connection. This is a requirement for PIM-enabled trunk interfaces.
`multicast-allowed`	Must be set to `yes` for PIM-enabled interfaces.
`enable`	Enables or disables the PIM routing protocol on the interface.
`hello-interval`	Number of seconds between sending hello messages to PIM neighbors on this interface. The valid range is from 1 to 65535 with a default value of 30 seconds. The value must be less than that of the `hello-holdtime` parameter.
`hello-holdtime`	Number of seconds a receiver of hello messages must consider the sender reachable before timing out the sender. The valid range is from 1 to 65535 with a default value of 105 seconds. The value must be greater than that of the `hello-interval` parameter.
`hello-priority-option`	Whether the Stinger unit will participate in DR election on this interface (`yes` or `no`, with a default value of `yes`).
`hello-priority`	DR election priority for the Stinger unit on the trunk interface. The DR election priority is a 32-bit unsigned number contained in a hello message. A router with a numerically larger priority is preferred in electing a new DR. The valid range for this setting is from 0 to 4,294,967,295, with a default setting of 1.
`join-prune-interval`	Number of seconds between sending PIM join/prune messages to PIM neighbors on this interface. A join/prune message consists of a list of groups and a list of joined and pruned sources for each group. The valid range is from 1 to 65535 with a default value of 60 seconds. The value must be less than that of the `join-prune-holdtime` parameter. **Note** Stinger units do not currently support the (S,G) state, so it always sends (*,G) join/prune messages.

Parameter	Setting
`join-prune-holdtime`	Number of seconds a receiver of join/prune messages must consider the list valid before timing out the information. The valid range is from 1 to 65535 with a default value of 210 seconds. The value must be greater than that of the `join-prune-interval` parameter.
`lan-delay-option`	Whether the Stinger unit will expect propagation delay over an Ethernet interface (`yes` or `no`, with a default value of `yes`). The `lan-delay` option is not sent in hello messages on a trunk interface, even if it is configured, because it applies only to LAN interfaces.
`lan-delay`	Number of milliseconds of expected propagation delay over the Ethernet interface. The valid range is from 1 to 65535 with a default value of 5000 milliseconds. This value is not sent in hello messages on a trunk interface, even if it is configured, because it applies only to LAN interfaces.
`override-interval`	A delay interval, in milliseconds, used to randomize when scheduling a delayed join message. The valid range is from 1 to 65535 with a default value of 2500 milliseconds. Because the `override-interval` is sent along with `lan-delay`, this value is not sent in hello messages on a trunk interface, even if it is configured.

Example of enabling PIM on the Gigabit Ethernet interface

The following commands enable PIM on the IP2000 Gigabit Ethernet interface:

```
admin> read ip-interface { { 1 8 2 } 0 }
admin> set ip-address = 1.1.1.2
admin> set multicast-allowed = yes
admin> set pim-options enable = yes
admin> write -f
```

Example of enabling PIM on a trunk interface

The following commands enable PIM on an MPOA terminating PVC on a trunk port. A numbered interface is required. For details about numbered interfaces, see "Example of a numbered interface using local-address" on page 4-36.

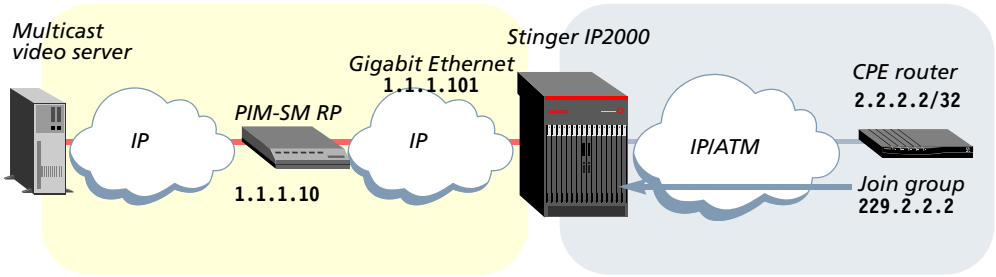
```
admin> new connection pim-trunk
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 123.123.123.2/24
admin> set ip-options local-address = 123.123.123.1/24
admin> set ip-options multicast-allowed = yes
admin> set ip-options pim-options enable = yes
admin> set telco-options nailed-groups = 851
```

```
admin> set mp-options enabled = no
admin> set atm-options vci = 38
admin> set atm-options nailed-group = 851
admin> write -f
```

Sample PIM-SM system configuration

For details about multicast client configuration, see "Configuring multicast client interfaces" on page 8-11. In the sample setup shown in Figure 9-1, PIM-SM is enabled on the IP2000 Gigabit Ethernet port.

Figure 9-1. PIM-SM on Gigabit Ethernet and trunk interface



The following commands enable PIM-SM and configure the Stinger unit to act as C-BSR using the IP address of the Gigabit Ethernet interface (1.1.1.101 in this example):

```
admin> read ip-global
admin> set multicast-forwarding = yes
admin> set pim-options enable = yes
admin> set pim-options cbsr-enable = yes
admin> set pim-options cbsr-ip-address = 1.1.1.101
admin> write -f
```

The following commands configure a PVC for multicast client CPE router in Figure 9-1:

```
admin> new connection mcast-client
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 2.2.2.2/32
admin> set ip-options multicast-allowed = yes
admin> set ip-options multicast-rate-limit = 20
admin> set atm-options vci = 100
admin> set atm-options nailed-group = 51
admin> write -f
```

The following commands enable multicast and PIM-SM on the IP2000 Gigabit Ethernet interface:

```
admin> read ip-interface { { 1 8 2 } 0 }
admin> set ip-address = 1.1.1.101
admin> set multicast-allowed = yes
admin> set pim-options enable = yes
admin> write -f
```

The following commands configure a group-to-RP mapping specifying the RP shown in Figure 9-1:

```
admin> new pim-group-rp-mapping 229
admin> set rp-address = 1.1.1.10
admin> set group-address = 229.0.0.0/8
admin> write -f
```

Following is the relevant group mapping:

```
admin> pim rp
Group           RP-Address      RPF neighbor    Priority   holdtime
Static Entries:
229.0.0.0/8     1.1.1.10        -               -           -
```

The following command shows the PIM neighbor across the Gigabit Ethernet interface:

```
admin> pim nbr
Neighbor        Interface    Priority     Holdtime        DR
1.1.1.10            1         100         105:96          No
```

Administrative tools for PIM-SM routing

The `pim` command displays PIM-related information for active PIM-enabled interfaces in the system. For details, see "PIM-SM diagnostics" on page A-11.

The system provides SNMP MIB support for the PIM protocol as defined in `draft-ietf-pim-mib-v2-01.txt` (the PIMv2 MIB). The PIMv2 MIB is placed in the MIB tree under `experimental 61`. For details, see "PIMv2 MIB support" on page A-30.

In addition, the output of the `netstat -s` command now includes the total PIM statistics for all PIM-enabled interfaces in the system. For example:

```
admin> netstat -s
...
pim:
     25 packets received
     24 hello packet received
     1 C-RP packets received
     38 packets transmitted
     26 hello packets sent
     12 Bootstrap packets sent
```

Filter Configuration

10

In a `filter` profile, the IP2000 controller currently supports `ip-filter`, `route-filter`, and `ethernet-filter` subprofile configurations. In addition, the `gen-filter` subprofile has one supported use—to define an explicit default rule for an IP filter.

Filter overview

The IP2000 supports the following types of filter:

- IP filters, for filtering packets based on IP packet header fields.
- Route filters, for filtering specific routes in RIP update packets.
- Ethernet input filters, for filtering inbound packets on the basis of Ethernet protocol or MAC addresses (or both).

After you define a filter and apply it to an interface, the system monitors traffic on the interfaces and, when a match occurs, takes the specified action.

Table 10-1. Default filtering behavior

Filters	Action on packets
No filter is applied to the connection.	No packets are dropped due to filtering.
A filter rule is applied to the connection.	All packets except those matching the filter rule are dropped. Packets that match the filter rule are forwarded or dropped according to the definition in the rule (`forward = yes` or `forward = no`).
An explicit default rule is applied to the connection.	All packets that do not match specific filter rules are forwarded or dropped according to the definition in the default rule (`forward = yes` or `forward = no`).

Filter rules

A `filter` profile includes up to 12 input filters and 12 output filters, each of which specifies a set of rules in one direction (inbound or outbound). Rules define comparisons that are made in a defined order and an action to be taken when a match occurs.

The system applies the rules in sequence, from 1 to 12. The action specified in the rule can apply to packets that match the rule, or to all packets *except* those that match the rule.

The filtering process stops immediately if conditions in a packet match a rule, and the action in that rule is applied to the packet. If a packet does not match a rule, the next rule is applied to the packet, and so forth. If no matches occur, the filtering subsystem drops the nonmatching packet. However, if an explicit default rule is specified as the last rule in the sequence, the system applies the action specified in that rule.

Explicit default filter rules

For filtered connections, the system default is to drop all packets that do not match the filter rules. If an explicit default rule is defined, the action specified in that rule overrides the system default. An explicit default rule has following settings:

- All filter fields of the default rule must be set to `0` (default values when the filter is created).
- The `valid-entry` field must be set to `yes`
- The `forward` entry must be set to the desired default action (`drop` or `forward`).
- The `type` of the filter must be set.

If a filter specifies more than one explicit default rule, only the first one is taken into account.

Defining IP filters

IP filters use IP packet header fields to select traffic to be filtered.

Overview of `ip-filter` settings

Following are the `filter` profile settings for defining IP filters. The parameters are shown with their default values for input filter rules. The same values apply for output filter rules—setting the parameters in an input filter affects the inbound data stream, and setting them in an output filter rule affects the outbound data stream.

```
[in FILTER/""]
filter-name* = ""

[in FILTER/"":input-filters[1]]
valid-entry = no
forward = no
type = gen-filter

[in FILTER/"":input-filters[1]:ip-filter]
protocol = 0
source-address-mask = 0.0.0.0
source-address = 0.0.0.0
dest-address-mask = 0.0.0.0
```

```
dest-address = 0.0.0.0
Src-Port-Cmp = none
source-port = 0
Dst-Port-Cmp = none
dest-port = 0
tcp-estab = no
```

Parameter	Setting
`filter-name`	Name of the filter profile, up to 36 characters. You apply a filter to an interface by referring to this name.
`valid-entry`	Enable/disable the input or output rule. With a setting of `no` (the default), the system skips this rule when filtering the data stream. Set this parameter to `yes` for each rule you define.
`forward`	Forwarding action for the rule. The default value of `no` causes the system to discard matching packets.
`type`	Type of filter (`ip-filter`, `gen-filter`, `route-filter`, and `ethernet-filter`). Only the parameters in the corresponding subprofile are applicable for the rule.
`protocol`	Protocol number. A number of 0 (zero) matches all protocols. A nonzero number is compared to the Protocol field in each packet. For a list of assigned protocol numbers, see RFC 1700, *Assigned Numbers*.
`source-address-mask`	Mask to be applied to the `source-address` value before comparing that value to the source address of a packet.
`source-address`	IP address. After applying the `source-address-mask` value, the system compares the result to the source address in a packet. See "Filtering on source or destination IP addresses" on page 10-4.
`dest-address-mask`	A mask to be applied to the `dest-address` value before comparing that value to the destination address of a packet.
`dest-address`	IP address. After applying the `dest-address-mask` value, the system compares the result to the destination address in a packet. See "Filtering on source or destination IP addresses" on page 10-4.
`src-port-cmp`	Type of comparison to be done on the source UDP/TCP port. The `less` (less than) and `gtr` (greater than) operators are not supported when comparing source port values in traffic destined for an external system. See "Filtering on port numbers" on page 10-5.
`source-port`	Port number to be compared against the source port of a packet.
`dst-port-cmp`	Type of comparison to be done on the destination UDP/TCP port. See "Filtering on port numbers" on page 10-5.

Parameter	Setting
`dest-port`	Port number to be compared against the destination port of a packet.
`tcp-estab`	*Not used by the IP2000.* Setting a value for this parameter does not cause the system to display a warning message, and makes no difference to the filtering functionality.

Details of IP filter comparison passes

The system compares a packet to rule #1. If the comparison fails (the packet does not match the rule), the system proceeds to rule #2, and so forth. When a comparison succeeds, the filtering process stops and the forwarding action in that filter is applied to the packet. Within each input or output filter, the comparisons proceed as follows:

1. Apply the `source-address-mask` value to the `source-address` value and compare the result to the source address of the packet. If they are not equal, the comparison fails.
2. Apply the `dest-address-mask` value to the `dest-address` value and compare the result to the destination address in the packet. If they are not equal, the comparison fails.
3. If the `protocol` parameter is zero (which matches any protocol), the comparison succeeds. If it is nonzero and not equal to the protocol field in the packet, the comparison fails. If port numbers are being used in the rule, either alone or with IP addresses, the `protocol` parameter must be set to the corresponding value.
4. If the `src-port-cmp` parameter is not set to `none`, compare the `source-port` number to the source port number of the packet. If they do not match as specified by the `src-port-cmp` parameter, the comparison fails.
5. If the `dst-port-cmp` parameter is not set to `none`, compare the `dest-port` number to the destination port number of the packet. If they do not match as specified by the `dst-port-cmp` parameter, the comparison fails.

If all comparisons fail, the packet does not match the filter. For security purposes, the Stinger IP2000 does not automatically forward nonmatching packets unless the filter explicitly allows nonmatching packets to pass.

Filtering on source or destination IP addresses

When you specify a source or destination address in an IP filter, the system applies the filter's forwarding action to packets received from or sent to that address. If you also specify a subnet mask, the system applies the mask to the address value before comparing the resulting value to the source or destination address in a packet.

To apply the mask, the system translates both the mask and address values into binary format and then uses a logical AND to apply the mask to the address. The mask hides the bits whose positions match those of the binary zeros in the mask. A mask of all zeros (the default) masks all bits. If the address value itself is also all zeros (the default), the filter matches any source or destination address. A mask of all ones (255.255.255.255) masks no bits, so the full address for a single host is compared to the address value.

You can use the address mask to mask out the host portion of an address, for example, or the host and subnet portion, so the rule matches the address to or from any host on a given network.

Filtering on port numbers

IP filters can specify a port number to be compared to the source or destination port (or both) in a packet. A port number of zero matches nothing. TCP and UDP port numbers are typically assigned to services. For a list of well-known port assignments, see RFC 1700, *Assigned Numbers*.

Note For security purposes, Lucent Technologies recommends that you filter all services from outside your domain that are not required. UDP-based services make your network particularly vulnerable to certain types of security attacks.

The type of comparison determines when a match occurs. For source port values, filters applied to traffic destined for an external system support the `none` (no comparison is made) or `eql` (equal to) operators.

For other traffic, the following operators can be used to compare source port or destination port values:

- `none` (no comparison is made)
- `eql` (equal to)
- `less` (less than)
- `gtr` (greater than)

Note The `neq` (not equal to) operator is not supported for port comparisons.

The following commands show an *illegal* rule that uses the unsupported `neq` operator to forward packets with a source port not equal to 50:

```
admin> set input-filters 1 valid-entry = yes
admin> set input-filters 1 Type = ip-filter
admin> set input-filters 1 forward = yes
admin> set input-filters 1 ip-filter protocol = 17
admin> set input-filters 1 ip-filter Src-Port-Cmp = neq
admin> set input-filters 1 ip-filter source-port = 50
```

If the filter containing this unsupported rule is applied to a connection or an interface, the system logs the following warning message when the connection or the interface goes into the UP state.

```
LOG warning, Shelf 1, Controller-1, Time: 02:23:31--
IP Filters: Not equal operation not supported for source port comparison
```

In this case, the faulty rule specifying the `neq` operator is not applied. The other rules of the filter are applied to the traffic stream.

The following commands show a legal workaround using the `less` and `gtr` comparison operator in two rules to accomplish the same effect as using the unsupported `neq` operator:

```
admin> set input-filters 1 valid-entry = yes
admin> set input-filters 1 Type = ip-filter
admin> set input-filters 1 forward = yes
```

```
admin> set input-filters 1 ip-filter protocol = 17
admin> set input-filters 1 ip-filter Src-Port-Cmp = less
admin> set input-filters 1 ip-filter source-port = 50
admin> set input-filters 2 valid-entry = yes
admin> set input-filters 2 Type = ip-filter
admin> set input-filters 2 forward = yes
admin> set input-filters 2 ip-filter protocol = 17
admin> set input-filters 2 ip-filter Src-Port-Cmp = gtr
admin> set input-filters 2 ip-filter source-port = 50
```

Sample IP filters

The following sections present sample IP filters and illustrate some of the issues you might consider when writing your own IP filters. The sample filters presented here do not address the fine points of network security. You might want to use these filters as a starting point and augment them to address your security requirements.

Note Some of the sample IP filters in the next sections include explicit default rules of type `ip-filter`. In general, an explicit default rule of type `generic-filter` is recommended to ensure the expected behavior in all types of profiles. For an example, see "Sample filter using a generic explicit default rule" on page 10-9.

Preventing address spoofing

Spoofing IP packets allows an intruder on a remote network to impersonate a local system's IP address. This section presents an example of an IP filter that prevents address spoofing when applied to a CPE interface.

The following commands create input filter #, which drops packets that have a local source address. In this example, the local network has an IP address of 192.100.50.128, with a subnet mask of 255.255.255.192. These values are just arbitrary examples. Because `forward` is set to `no` (the default), inbound packets with a source address on the LAN will be dropped.

```
admin> new filter ip-spoof
admin> set input 1 valid = yes
admin> set input 1 type = ip-filter
admin> set input 1 ip-filter source-address-mask = 255.255.255.192
admin> set input 1 ip-filter source-address = 192.100.50.128
```

The next set of commands creates input filter #2, which drops packets with a source address equal to the loopback address (127.0.0.0).

```
admin> set input 2 valid = yes
admin> set input 2 type = ip-filter
admin> set input 2 ip-filter source-address-mask = 255.0.0.0
admin> set input 2 ip-filter source-address = 127.0.0.0
```

The next set of commands creates input filter #3, which explicitly accepts all remaining source addresses and forwards them to the local network. Except for `forward` = `yes`, the third filter uses all default values. Because `forward` is set to `yes`,

the system forwards all remaining packets (those with nonlocal source addresses) to the LAN.

```
admin> set input 3 valid = yes
admin> set input 3 forward = yes
admin> set input 3 type = ip-filter
```

The next set of commands creates an output filter and sets the forwarding action to `yes`. This filter specifies the source mask and address for the local network. (Packets originating on the local network should be forwarded to the CPE.)

```
admin> set output 1 valid = yes
admin> set output 1 type = ip-filter
admin> set output 1 forward = yes
admin> set output 1 ip-filter source-address-mask = 255.255.255.192
admin> set output 1 ip-filter source-address = 192.100.50.128
admin> write -f
```

An IP filter for more complex security issues

In this example, the local network supports a Web server, and the administrator needs to carry out the following tasks:

- Provide client access to the server's IP address.
- Restrict ingress traffic to all other hosts on the local network.

However, many local IP hosts need to access the Internet and use IP-based applications such as `telnet` or `ftp`, so their response packets need to be directed appropriately to the originating host. In this example, the Web server's IP address is 192.9.250.5. The filter will be applied in `connection` profiles as a data filter.

The following commands create the first input filter, which sets `forward` to `yes` and allows packets to reach the Web server's destination address at a destination TCP port that can be used for `telnet` or `ftp`:

```
admin> new filter web-access
admin> set input 1 valid = yes
admin> set input 1 forward = yes
admin> set input 1 type = ip-filter
admin> set input 1 ip-filter protocol = 6
admin> set input 1 ip-filter dest-address-mask = 255.255.255.255
admin> set input 1 ip-filter dest-address = 192.9.250.5
admin> set input 1 ip-filter dst-port-cmp = eql
admin> set input 1 ip-filter dest-port = 80
```

The next set of commands creates the second input filter, which allows inbound TCP packets in response to a local user's outbound `telnet` request, by specifying that TCP packets whose destination port number is higher than that of the source port are forwarded. (The `telnet` requests go out on port 23, and responses come back on some random port above port 1023.)

```
admin> set input 2 valid = yes
admin> set input 2 forward = yes
admin> set input 2 type = ip-filter
admin> set input 2 ip-filter protocol = 6
admin> set input 2 ip-filter dst-port-cmp = gtr
admin> set input 2 ip-filter dest-port = 1023
```

The next set of commands creates the third input filter, which allows inbound RIP updates, by specifying that inbound UDP packets are forwarded if the destination port number is higher than that of the source port. (For example, suppose a RIP packet goes out as a UDP packet to destination port 520. The response to this request goes to a random destination port above port 1023.)

```
admin> set input 3 valid = yes
admin> set input 3 forward = yes
admin> set input 3 type = ip-filter
admin> set input 3 ip-filter protocol = 17
admin> set input 3 ip-filter dst-port-cmp = gtr
admin> set input 3 ip-filter dest-port = 1023
```

The following commands create the fourth input filter. This filter uses all default values, which allows unrestricted use of `ping` and `traceroute`.

```
admin> set input 4 valid = yes
admin> set input 4 forward = yes
admin> set input 4 type = ip-filter
admin> write -f
```

Sample filter with no explicit default rule

When you do not define an explicit default rule, the system uses the implicit default which discards all packets that do not match the input and output filter rules. If a filter defines rules only in one direction, traffic in the other direction is not filtered. For example, the following sample filter specifies that UDP packets from a source port less than 50 should be forwarded. Because it does not specify an explicit default rule, this filter causes all packets that do not match input filter 1 to be dropped.

```
admin> new filter input-filter-1
admin> set input-filters 1 valid-entry = yes
admin> set input-filters 1 forward = yes
admin> set input-filters 1 Type = ip-filter
admin> set input-filters 1 ip-filter protocol = 17
admin> set input-filters 1 ip-filter source-address-mask = 255.255.255.255
admin> set input-filters 1 ip-filter source-address = 192.168.2.2
admin> set input-filters 1 ip-filter Src-Port-Cmp = less
admin> set input-filters 1 ip-filter source-port = 50
admin> write -f
```

Sample filter with explicit default rule

Defining an explicit default rule allows more flexibility in specifying which packets to drop. For example, the following sample filter specifies that UDP packets from a source port less than 50 should be *dropped*, and includes an explicit default rule that causes all other incoming IP packets to be *forwarded*. Because this filter does not define output rules, all packets are forwarded in the output direction.

```
admin> new filter input-filter-2
admin> set filter-name = input-filter
admin> set input-filters 1 valid-entry = yes
admin> set input-filters 1 forward = no
admin> set input-filters 1 Type = ip-filter
admin> set input-filters 1 ip-filter protocol = 17
admin> set input-filters 1 ip-filter source-address-mask = 255.255.255.255
admin> set input-filters 1 ip-filter source-address = 192.168.2.2
admin> set input-filters 1 ip-filter Src-Port-Cmp = less
admin> set input-filters 1 ip-filter source-port = 50
admin> set input-filters 2 valid-entry = yes
admin> set input-filters 2 forward = yes
admin> set input-filters 2 Type = ip-filter
admin> write -f
```

Sample filter using a generic explicit default rule

Note The IP2000 does not support generic packet filters, but you can specify a generic explicit default rule in an IP filter. This is the only supported use of `generic-filter` for connections terminating on the IP2000. A generic default rule affects *all* packet types that do not match the filter rules for a certain direction (input or output).

To define an explicit default rule that affects *all* packets that do not match the filter rules in a certain direction, set the `Type` value to `generic-filter`.

If the explicit default filter rule is of type `ip-filter`, ARP packets (and other non-IP packets) will not be affected by the default. For example, if you want to forward all nonmatching packets including non-IP packets such as ARP packets, you must create a `generic-filter` rule as default with the action set to `forward`.

The following filter specifies a generic explicit default rule to allow forwarding of all incoming packets that do not match the input filter rules. Because the output direction does not specify an explicit default rule, all packets that do not match the output filter rules will be dropped.

```
admin> new filter input-output
admin> set input-filters 1 valid-entry = yes
admin> set input-filters 1 forward = no
admin> set input-filters 1 Type = ip-filter
admin> set input-filters 1 ip-filter protocol = 17
admin> set input-filters 1 ip-filter source-address-mask = 255.255.255.255
```

```
admin> set input-filters 1 ip-filter source-address = 192.168.2.2
admin> set input-filters 1 ip-filter Src-Port-Cmp = less
admin> set input-filters 1 ip-filter source-port = 50
admin> set input-filters 2 valid-entry = yes
admin> set input-filters 2 forward = yes
admin> set input-filters 2 Type = generic-filter
admin> set output-filters 1 valid-entry = yes
admin> set output-filters 1 forward = yes
admin> set output-filters 1 Type = ip-filter
admin> set output-filters 1 ip-filter protocol = 17
admin> set output-filters 1 ip-filter Src-Port-Cmp = less
admin> set output-filters 1 ip-filter source-port = 50
admin> write -f
```

Defining route filters

Route filters examine RIP update packets and exclude specific routes from the local system's routing table, or include routes in the table only after modifying their metrics. For route filters, the forwarding action in the filter has no effect.

Overview of `route-filter` settings

In a `filter` profile, the `route-filter` subprofile contains the following parameters. The parameters are shown with their default values for input filters. The same values apply for output filter rules.

```
[in FILTER/""]
filter-name* = ""

[in FILTER/"":input-filters[1]]
valid-entry = no
forward = no
type = gen-filter

[in FILTER:input-filters[1]:route-filter]
source-address-mask = 0.0.0.0
source-address = 0.0.0.0
route-mask = 0.0.0.0
route-address = 0.0.0.0
add-metric = 0
action = none
```

If you set the parameters in an input filter, only inbound packets are examined. If you set them in an output filter, only outbound packets are examined.

Parameter	Setting
`filter-name`	Name of the filter profile, up to 36 characters. You apply a filter to an interface by referring to this name.
`valid-entry`	Enable/disable the rule.

Parameter	Setting
forward	This setting does not apply to route filters.
type	Type of filter (ip-filter, gen-filter, route-filter, and ethernet-filter). Only the parameters in the corresponding subprofile are applicable for the rule.
source-address-mask	Mask to be applied to the source-address value before comparing that value to the source address of a RIP update packet.
source-address	IP address. After applying the source-address-mask value, the system compares the result to the source address in a RIP packet.
route-mask	Mask to be applied to the destination address of a route.
route-address	IP address. After applying the route-mask value, the system compares the result to routes in a RIP packet. If it finds a route with a matching destination, it takes the specified action.
add-metric	Number from 1 to 15, to be added to the metric value for a route that matches the filter rule, if the specified value for the action parameter is Add.
action	An action to take on a route that matches the filter rule. Valid values are none (the default), accept (accept the route by allowing it to affect the routing table), deny (deny the route by not allowing it to affect the routing table), or add (add the value of the add-metric parameter to the route metric and accept the route).

Sample route filters

The following sample filters show how to exclude a route and how to modify a route's metrics.

Sample route filter that excludes a route

In this example, the defined input filters accept all inbound RIP packets except those with a destination of 90.0.0.0. Following are the commands entered to define the filter, and the system's responses:

```
admin> new filter route-test
admin> set input 1 valid = yes
admin> set input 1 type = route-filter
admin> set input 1 route route-mask = 255.0.0.0
admin> set input 1 route route-address = 90.0.0.0
admin> set input 1 route action = deny
admin> set input 2 valid = yes
admin> set input 2 type = route-filter
admin> set input 2 route action = accept
admin> write -f
```

In this sample route filter, any route that matches rule 1 is rejected, and all other routes are accepted (because they match rule 2).

Sample route filter that configures a route's metric

In this example, an output filter identifies the route 11.0.0.0 in outbound RIP packets and assigns a high metric to that route.

```
admin> new filter metrics
admin> set output 1 valid = yes
admin> set output 1 type = route-filter
admin> set output 1 route route-mask = 255.0.0.0
admin> set output 1 route route-address = 11.0.0.0
admin> set output 1 route add-metric = 7
admin> set output 1 route action = add
admin> write -f
```

Defining Ethernet input filters

The system supports Ethernet input filters for filtering input streams on the basis of Ethernet type and MAC addresses. Ethernet filters examine the EtherType field in the Ethernet header and forward or drop only the specified type of traffic (for example, PPPoE traffic). MAC address filters examine source and destination MAC addresses in the Ethernet header and forward or drop frames from a particular source or destination.

Currently, Ethernet filters apply only to input streams.

Overview of `ethernet-filter` settings

Following are the parameters of the new `ethernet-filter` subprofile, shown with default settings:

```
[in FILTER/""]
filter-name* = ""
[in FILTER/"":input-filters[1]]
valid-entry = no
forward = no
type = gen-filter
[in FILTER/"":input-filters[1]:ethernet-filter]
source-mac-address = 00:00:00:00:00:00
destination-mac-address = 00:00:00:00:00:00
ethernet-type = 00:00
```

Parameter	Setting
`filter-name`	Name of the filter profile, up to 36 characters. You apply a filter to an interface by referring to this name.
`valid-entry`	Enable/disable the rule.
`forward`	Forwarding action for the rule. The default value of `no` causes the system to discard matching packets.

Parameter	Setting
type	Type of filter (ip-filter, gen-filter, route-filter, and ethernet-filter). Only the parameters in the corresponding subprofile are applicable for the rule.
source-mac-address	A source MAC address. A six-byte hexadecimal number uniquely representing the source host; for example, 12:34:56:78:9a:bc. The system will forward or discard packets from this address, depending on how the rule is defined.
destination-mac-address	A destination MAC address. A six-byte hexadecimal number uniquely representing the destination host; for example, 12:34:56:78:9a:bc. The system will forward or discard packets addressed to this host, depending on how the rule is defined.
ethernet-type	Ethernet type. A two-byte hexadecimal number representing the Ethernet protocol. PPPoE values are 8863 (PPPoE Discovery Stage packets), and 8864 (PPP Session Stage packets). The system will forward or discard packets of the specified types, depending on how the rule is defined.

Sample PPPoE and MAC address filter

The following sample filter shows both PPPoE filtering (explicitly allowing only inbound PPPoE traffic) and MAC address filtering (explicitly discarding packets to and from specific devices).

```
admin> new filter
admin> set filter-name = enet-filter2
```

The following commands configure input-filter 1 to forward packets with ethertype = 0x8863 (PPPoE Discovery Stage) and input-filter 2 to forward packets with ethertype = 0x8864 (PPP Session Stage). All other, nonmatching types of Ethernet frames will be discarded.

```
admin> set input 1 Type = ethernet-filter
admin> set input 1 valid-entry = yes
admin> set input 1 forward = yes
admin> set input 1 ethernet-filter ethernet-type = 88:64
admin> set input 2 Type = ethernet-filter
admin> set input 2 valid-entry = yes
admin> set input 2 forward = yes
admin> set input 2 ethernet-filter ethernet-type = 88:63
```

The next commands configure input-filter 3 to discard packets from the MAC address 11:22:33:44:55:66 and to the MAC address 66:22:33:44:55:11:

```
admin> set input 3 Type = ethernet-filter
admin> set input 3 valid-entry = yes
admin> set input 3 ethernet-filter source-mac = 11:22:33:44:55:66
```

```
admin> set input 3 ethernet-filter destination-mac = 66:22:33:44:55:11
admin> write -f
```

Applying a filter to an interface

This section describes how to apply a filter to a PVC that terminates on the IP2000, and to the module's Ethernet interfaces.

Settings in `connection` and `ethernet` profiles

To apply a filter to an IP interface or a CPE device, set the following parameters (shown with their default settings):

```
[in CONNECTION/"":session-options]
data-filter = ""
[in CONNECTION/"":ip-options]
route-filter = ""
[in IP-INTERFACE/{ { shelf-1 first-control-module 2 } }]
route-filter = ""
[in ETHERNET { any-shelf any-slot 0 }]
filter-name= ""
```

Parameter	Setting
`data-filter`	Name of a `filter` profile containing IP or Ethernet filter rules to be applied to the WAN interface.
`route-filter`	Name of a `filter` profile containing route filter rules to be applied to the interface.
`filter-name`	Name of a `filter` profile to be applied to the Ethernet interface.

Examples of applying filters to a CPE interface

When you apply an IP filter in a CPE `connection` profile, it prevents certain inbound packets from reaching the LAN side of the system, or certain outbound packets from reaching the CPE router. When you apply a route filter, it affects which routes received in RIP updates on this interface will be allowed to affect the Stinger system's routing table. Following is an example of applying both an IP filter and a route filter to a terminating PVC:

```
admin> read connection cpe-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options remote-address = 3.3.3.200/30
admin> set atm-options vpi = 8
admin> set atm-options vci = 100
admin> set session-options data-filter = ip-filter1
```

```
admin> set ip-options route-filter = route-filter1
admin> set atm-options nailed-group = 201
admin> write -f
```

Example of applying a filter to a LAN interface

Filters are not currently supported in the `vlan-ethernet` profile. However, if you apply a filter to an Ethernet interface, it is applied to all VLANs on that interface. A filter applied to an Ethernet interface takes effect immediately. If you change any settings in a `filter` profile, the changes apply as soon as you save the `filter` profile.

Note Use caution when applying a filter to the Ethernet interface. You could inadvertently render the Stinger IP2000 inaccessible from the LAN.

The following set of commands applies an Ethernet filter to the IP2000 Gigabit Ethernet interface:

```
admin> read ethernet { 1 8 2 }
admin> set filter-name = enet-filter1
admin> write -f
```

Sample application of an Ethernet filter to a VLAN bridging connection

The following commands create a subscriber VLAN bridging connection and apply the sample Ethernet filter (`enet-filter2`) created in "Sample PPPoE and MAC address filter" on page 10-13.

```
admin> new connection
admin> set station = vlan-pppoe-1
admin> set active = yes
admin> set encapsulation-protocol = atm
admin> set ip-options ip-routing-enabled = no
admin> set bridging-options bridging-group = 451
admin> set bridging-options bridge = yes
admin> set bridging-options bridge-type = transparent-bridging
admin> set session-options data-filter = enet-filter2
admin> set atm-options vci = 60
admin> set atm-options nailed-group = 125
admin> write -f
```

Administrative tools for filters

If you have access to the debug environment, you can use the debug-level `info np` command to verify that packets are being discarded and forwarded properly on a filtered interface. For details about using this command and enabling the debug environment, see "Network processor-related diagnostics" on page A-20.

The system also supports the system-level `filterdisp` command for displaying information about filters in use on all terminating connections. With no command-line arguments, the command displays all active sessions and their filter names. For example,

```
admin> filterdisp
ID   Username    Src    Route-Filter  Data-Filter   Call-Filter   TOS-Filter
-------------------------------------------------------------------------
015  term        loc                 f1
021  pvc2        loc                  f1
022  pvc4        loc                  f1
023  pvc5        loc
<end user list> 4 active user(s)
```

The first column of the output displays a session ID number, followed by a username and the name of the filter. To display details for a particular session, specify the session ID as an argument on the `filterdisp` command line.

For example, the following sample output shows that no filters are applied to session 23:

```
admin> filterdisp 23
Hostname:       pvc5
No associated filters
```

The following sample output shows filters applied to an externally authenticated session:

```
admin> filterdisp 17
Hostname:       edleung
searching for external filters...
Externally obtained filters exist

Data Filter
Direction: Out

Forward = yes
Type = IP Filter
protocol = 0
source-address-mask = 0.0.0.255
source-address = 1.1.1.2
destination-address-mask = 0.0.0.0
destination-address = 0.0.0.0
Src-Port-Cmp = none
source-port = 0
Dst-Port-Cmp = none
dest-port = 0
tcp-estab = no
```

IP2000 Diagnostics

A

This appendix describes the `gmac`, `igmp`, and `pim` system-level commands. Information provided by the `gmac` command is also accessible to an external management utility through the `ip2kstats.mib` management information base (MIB). Information displayed by the pim command can also be accessed through the PIMv2 MIB (currently defined in `draft-ietf-pim-mib-v2-01.`txt) under `experimental 61`.

In addition, this appendix describes the following debug-level commands, which are not documented in the *Stinger Reference* and are not supported.

- `diag igmpsp`
- `diag igmp`
- `brtbls`
- `ifmgr`
- `diag brtbl`
- `sar`
- `info np`

Caution Debug-level commands can be used to display low-level details about IP2000 operations. However, they are introduced into the system for development purposes, and are not part of the supported software environment. Use debug-level commands with caution!

Enabling the debug environment

To access the debug environment, log in using the default `super` (super-user) profile. For example:

```
admin> auth super
Password:
```

The default password for this account is `Ascend`. If the password has not been changed, you should change it now to prevent unauthorized super-user access to the system.

The debug environment contains many hidden commands and parameters that are not intended for general use. The debug environment is not supported, and the documentation provided here is not comprehensive.

Caution Under most circumstances, debug commands are not required for monitoring Stinger operations, and under some circumstances, these commands might produce undesirable results. Use the information with caution. Contact Lucent OnLine Customer Support at `http://www.lucent.com/support` with questions or concerns.

Gigabit Ethernet diagnostics

The `gmac` command provides diagnostic output about the Gigabit Ethernet media access controller (GMAC) driver. This command is new with the introduction of the IP2000.

gmac

Description Provide diagnostics on the Gigabit Ethernet (GigE) driver.

Permission level `system`

Usage `gmac [`*`options`*`]`

Command element	Description
`-v`	Show `gmac` version.
`-i [-u/d]`	With no additional option, initialize/reset the GigE port. `-i -u` Force GigE link up. `-i -d` Force GigE link down.
`-n`	Set up network processor to communicate with GMAC.
`-s`	Set up a SAR channel for communicating with GMAC .

Command element	Description
`-l [-i/e/d/p]`	Loopback.
	`-l -i` Set port for internal loopback.
	`-l -e` Set port for external loopback.
	`-l -d` Set port for no loopback
	`-l -p` Run loopback test for Ethernet power-on self test (POST).
`-p`	Ping test.
`-r`	Read a PHY register.
`-w`	Write to a PHY register.
`-d [-c/a/e]`	With no additional option, display all statistics.
	`-d -c` Clear GMAC statistics.
	`-d -a` Display ATM statistics.
	`-d -e` Display Ethernet statistics.
`-t`	Set debug level (0 through 3).
`-?`	Display a summary of commands.

Example The `gmac -n` command sets up the network processor for communication with the GMAC port:

```
super> gmac -n
NP setup for gmac done.
```

Example The `gmac -s` command sets up a SAR channel for communicating with the GMAC port:

```
super> gmac -s
GMAC: SAR conn. open with vpi = 0, vci = 200
```

Example The `gmac -v` command displays the GMAC version:

```
super> gmac -v
GMAC version : 0x0b
```

Example The `gmac -i` command resets the Gigabit Ethernet port:

```
super> gmac -i
gigE port reset.
```

Example The `gmac -d` command displays the total transmit and receive statistics for the GigE interface of the IP2000 controller. For details about the `gmac -d` output fields, see the descriptions of MIB objects in "Total transmit statistics" on page A-26 and "Total receive statistics" on page A-27. For example, the following command displays the current GMAC statistics:

```
super> gmac -d
Gigabit Ethernet port statistics :

txOctetsLow     = 1040
txOctetsHigh    = 0
txGoodPackets   = 4
```

```
txPkt64         = 0
txPkt65127      = 0
txPkt128255     = 0
txPkt256511     = 4
txPkt5121023    = 0
txPkt1O24Max    = 0
txPktDefer      = 0
txPktUndSz      = 0
txUnderFlow     = 0
txPfcf = 0
txPfcc = 0
txRfcf = 0
txRfcc = 0
txOverFlow      = 0
txAlmostFull    = 0

rxOctetsLow     = 1646718
rxOctetsHigh    = 0
rxGoodPackets   = 2059
rxPkt64         = 766
rxPkt65127      = 0
rx128255        = 0
rx256511        = 160
rx5121023       = 0
rx1O24Max       = 1133
rxMacType       = 0
rxCrcErrors     = 0
rxUnderSize     = 0
rxOverSize      = 0
rxAlmostFull    = 0
rxOverRun       = 0
rxMulticastPackets      = 1896
rxBroadcastPackets      = 46
rxJabber        = 0
rxPfc   = 0
rxRfc   = 0
```

IGMP diagnostics

The `igmp` command provides information about IGMP multicast operations. The `profile` argument has been added to support `mcast-service` profiles.

The `diag igmpsp` and `diag igmp` commands are available only in the debug environment. See "Enabling the debug environment" on page A-2 for related information.

igmp

Description Displays or clears multicast information about Internet Group Management Protocol (IGMP) groups and clients.

Permission level `system`

The system-level `igmp` command supports new set of arguments for displaying information obtained by IGMP snooping. The `igmp` command is supported both on the shelf and the individual LIM slots. Following is the new usage statement, which shows both the existing and new arguments:

```
admin> igmp
Igmp Commands Usage:
   igmp <mbone | groups | clients | slots | vc >
        <profile [mcast-profile] | delete [grp_addr][if_num]>
Igmp Snooping Commands Usage:
   igmp <smbone | sgroups [bridgeNum] | sclients [bridgeNum]>
        <sslots | sdelete [bridgeNum][grp_addr][if_num]>
```

Command element	Description

`mbone`	Display multicast backbones. For example:
	`admin> igmp mbone`
	`Mbone is currently:`
	`Slot 1:8 ifNum = 1`

`groups`	Display currently registered multicast group addresses and interfaces.
`clients`	Display multicast clients.
`slots`	Display multicast enable slots.

`vc`	Check the multicast VCs that are configured and active at any point of time. For example:
	`admin> igmp vc`
	`Current VCs:`
	`Profile Name IfNum Active`
	`oc3-multi 17 yes`
	`oc3-multi2 25 yes`
	`oc3-multi1 24 yes`

`profile [`*`mcast-profile`*`]`	Display multicast profiles. If you specify an `mcast-service` profile name, the command also displays the `connection` profiles associated with that profile. This option is available only on the system control module.
	For example, the following command displays information about an `mcast-service` profile named `test`:
	`admin> igmp profile test`
	`Mcast Service : test`
	`Filter Type : MCAST_FILTER_INCLUSIVE`
	`Filter List :`
	`239.10.10.10 - 239.10.10.10 ( Count = 1 )`
	`Connection Profiles :`
	`1) stg61`
	`2) stg63`

Command element	Description

`delete [grp_addr [if_num]]`	With the `delete` option alone, delete all currently registered multicast groups and their members. If a group address is specified, delete all members of that group. If a group address and interface number are specified, delete that member of the specified group. For example:
	`admin> igmp delete 226.1.1.2 12`
	`LOG notice, Shelf 1, Controller-1, Time: 10:25:53-- Multicast client 226.1.1.2 link DOWN interface number 12`
	When executed from the shelf controller prompt, only groups that have been joined from the trunk side can be deleted. You must open a session to the LIM to delete groups that have been joined from the LIM side.

`sdelete [bridgeNum] [grp_addr]`	Delete all the members of this group, for this bridge on this slot.
`sdelete [bridgeNum] [grp_addr][if_num]`	Delete a unique instance.

Example The `igmp client` command shows local MBONE and IGMP client interfaces. For example:

```
super> igmp client
IGMP Clients
        Client      Version  RecvCount  CLU    ALU
          2(Mbone)      2          0     0      0
          4(Mbone)      2          0     0      0
          1(Mbone)      2       3370     0      0
          5             2          0     0      0
          6             2          0     0      0
```

The output contains the following fields:

Field	Description
`Client`	In `igmp client` command output, the `Client` field displays the interface ID (the `ifIndex` value) on which the client resides. The 1 value represents the Gigabit Ethernet interface. Other numbers are WAN interfaces, numbered according to when they became active. The interfaces labeled `(Mbone)` receive multicast data from multicast routers.
`Version`	IGMP version.
`RecvCount`	Number of IGMP messages received on that interface.
`CLU/ALU`	CLU is current line utilization, and ALU is average line utilization. Both indicate the percentage of bandwidth used across this interface. If bandwidth utilization is high, some IGMP packet types are not forwarded.

Example The `igmp groups` command displays information about MBONE interfaces. Details about client member interfaces are maintained on the LIM itself. For example, the following command is invoked on the controller of a Stinger system with multicast clients on a DSL interface in slot 6 and an MBONE configured on the IP2000 Gigabit Ethernet interface:

```
super> igmp groups
IGMP Group address Routing Table
Up Time: Od 1:25:05
Group Address   MemberIf  Expire time    Counts
239.100.100.5   *         (Mbone)        0::0 S2
                Slot 1:6
```

The next commands open a session with the DSL LIM and invoke `igmp groups` on the LIM:

```
super> open 1 6
```

```
dadsl-atm-24-1/6> igmp groups
IGMP Group address Routing Table
Up Time: Od 1:23:46
Group Address   MemberIf  Expire time    Counts
239.100.100.5   6         00:04:07       0::0 S2
```

When the command is executed on the LIM, the output displays details about the corresponding client member interface. The output contains the following fields:

Field	Description
`Group address`	Multicast address used for the group. An asterisk indicates the IP multicast address being monitored. If a group has no members, the system forwards multicast traffic for the group to the MBONE interface (the default route).
`MemberIf`	Interface ID of multicast group members.
`Expire time`	When this membership expires. The system sends out IGMP queries every 60 seconds, so the expiration time is usually renewed. If the expiration time is reached, the system removes the entry from the table. If the field contains periods, this membership never expires. A string of periods means that the default route never times out.
`Counts`	Number of packets forwarded to the client, the number of packets dropped due to lack of resources, and the state of the membership.

Example The `igmp slot` command displays information about slots supporting IGMP clients:

```
super> igmp slot
   IGMP Client Slots
   Shelf:Slot         Group          SendCount
   1:8                230.0.0.9        0
   1:5                230.0.0.9        0
   1:2                230.0.0.9        0
```

The output contains the following fields:

Field	Description
Shelf:Slot	Shelf and slot card the MBONE connection is on.
Group	Interface number of connection.
SendCount	Number of packets sent across the interface.

Example The `igmp profile` command displays information about `mcast-service` profiles. If you execute the command on the control module and specify a profile name, the command will also display the `connection` profiles assocated with it.

```
super> igmp profile

IGMP Service Profiles

          Service Name               : gold-service
          Filter Type                : MCAST_FILTER_INCLUSIVE
          Filter List                :
                 224.255.129.120
                 224.225.129.119

          Service Name               : bronze-service
          Filter Type                : MCAST_FILTER_INCLUSIVE
          Filter List                :
                 224.255.129.119
```

The output contains the following fields:

Field	Description
Service Name	Name of the multicast service profile.
Filter Type	Inclusive or exclusive multicast group filtering in the named profile.
Filter List	Multicast group addresses to be filtered.

igmp hosts

Description Displays host addresses per member interface for fast-leave purposes. This command is not available on the controller. To use it, you must open a session to the module on which the member interfaces are established.

Permission level `system`

Usage `igmp hosts`

Example The following commands can be used to verify the hosts that have joined a multicast group on an ADSL module in slot 6:

```
admin> open 1 6

dadsl-atm-24-1/6> igmp hosts
Group Address   MemberIf  Host Address
230.1.1.1       1         10.10.10.2
                          10.10.10.3
```

In the command output, 230.1.1.1 is the group address and two hosts, 10.10.10.2 and 10.10.10.3, have joined the group on interface 1.

diag igmpsp

Description Enable low-level diagnostics on IGMP services profiles.

Permission level debug

Usage diag igmpsp

Example This example first uses a system-level command to view IGMP multicast service profile characteristics. It then uses the debug environment to display low-level diagnostic information.

```
super> igmp profile
IGMP Service Profiles

          Service Name               : gold-service
          Filter Type                : MCAST_FILTER_INCLUSIVE
          Filter List                :
                 239.100.100.4
          Service Name               : bronze-service
          Filter Type                : MCAST_FILTER_EXCLUSIVE
          Filter List                :
                 239.100.100.4
```

The mcast-service profile named bronze-service has been applied to a connection profile that uses the first port of an ADSL LIM in slot 6, and the profile named gold-service has been applied to a connection on the second port of that LIM. The following commands open a session with the ADSL LIM and display diagnostics on the service profiles:

```
super> open 1 6

dadsl-atm-24-1/6> diag igmpsp
mcastJoinFilterVerify: Received CLASS D address 239.100.100.4 in report
_profileNameCompare: Compare ( bronze with bronze )
mcastJoinFilterVerify: [ bronze ] is multicast service profile for [ adsl-6-
1 ] connection
mcastJoinFilterVerify: Found 239.100.100.4 in EXCLUSIVE filter list

mcastJoinFilterVerify: Received CLASS D address 239.100.100.4 in report
_profileNameCompare: Compare ( gold-ser with gold-ser )
mcastJoinFilterVerify: [ gold-ser ] is multicast service profile for [ term-
6-2 ] connection

mcastJoinFilterVerify: Received CLASS D address 239.100.100.5 in report
_profileNameCompare: Compare ( bronze with bronze )
mcastJoinFilterVerify: [ bronze ] is multicast service profile for [ adsl-6-
1 ] connection

mcastJoinMaxClientVerify: ifnum 5, multicast membership count 0

_profileNameCompare: Compare ( bronze with bronze )
mcastJoinMaxClientVerify: [ bronze ] is multicast service for [ adsl-6-1 ]
```

```
connection
mcastClientLinkUp: Multicast client up trap

mcastJoinFilterVerify: Received CLASS D address 239.100.100.5 in report
_profileNameCompare: Compare ( bronze with bronze )
mcastJoinFilterVerify: [ bronze ] is multicast service profile for [ adsl-6-
1 ] connection
```

The following output indicates that the system rejected a request to join the group listed in an exclusive filter list in the `bronze-service` profile:

```
mcastJoinFilterVerify: Received CLASS D address 239.100.100.4 in report
_profileNameCompare: Compare ( bronze with bronze )
mcastJoinFilterVerify: [ bronze ] is multicast service profile for [ adsl-6-
1 ] connection
mcastJoinFilterVerify: Found 239.100.100.4 in EXCLUSIVE filter list
```

The next output indicates that the system accepted the message received from a client on the second LIM interface:

```
mcastJoinFilterVerify: Received CLASS D address 239.100.100.4 in report
_profileNameCompare: Compare ( gold-ser with gold-ser )
mcastJoinFilterVerify: [ gold-ser ] is multicast service profile for [ term-
6-2 ] connection
```

The following output indicates that the system accepted a request to join a group that is not listed in the exclusive filter list in the `bronze-service` profile:

```
mcastJoinFilterVerify: Received CLASS D address 239.100.100.5 in report
_profileNameCompare: Compare ( bronze with bronze )
mcastJoinFilterVerify: [ bronze ] is multicast service profile for [ adsl-6-
1 ] connection
```

diag igmp

Description Enable low-level diagnostics on IGMP protocol messages.

Permission level `debug`

Usage `diag igmp`

Example This example displays IGMP join messages from a client. The following output shows a group message to a client on LIM slot 6, port 6 and the client's response requesting to join multicast group 230.0.0.9:

```
super> diag igmp
igmp debug is ON
igmpParseMsg:   IGMP packet to 230.0.0.9 type 6 on interface 6 port 6
      Receiving Version 2 Response from 6
      Joining Group 230.0.0.9
      IGMP: Joining new group 230.0.0.9
         _sendIGMPTableUpdateMsg: sending IGMP_TAB_ADD to 1:8
         _sendUpdateMsgToShelf : client 6 join group 230.0.0.9 vRouterID 0
igmpParseMsg: IGMP packet to 230.0.0.9 type 6 on interface 6 port 6
      Receiving Version 2 Response from 6
      IGMP: Refreshing group 230.0.0.9 input ifNum 6
```

PIM-SM diagnostics

The `pim` command provides information about Protocol Independent Multicast-Sparse Mode (PIM-SM v2) operations.

pim

Description Displays PIM-related information.

Permission level `system`

Usage `pim [ group | rp | nbr | if` *`ifnum`* `| bsr | hash` *`group-addr`* `]`

Command element	Description
`groups`	Displays information about multicast groups.
`rp`	Displays group-RP mappings.
`nbr`	Displays information about PIM neighbor routers.
`if` *`ifnum`*	Displays PIM interface statistics.
`bsr`	Displays information about candidate BSRs or the elected BSR.
`hash` *`group-addr`*	Displays the best RP for a group or group range.

Example The `pim groups` command displays information about all multicast groups. For example:

```
super> pim groups
Group Addr  RP/Source Addr upJPTimer  Tree(Rpt/Spt)
223.1.1.1   192.168.101.1  40         RPT
224.4.4.4   10.10.10.10    70         RPT
```

Output field (`pim groups`)	Description
`Group Addr`	Multicast group address for the entry
`RP/SourceAddr`	With the current software version, the RP/Source address is the RP's IP address.
`upJPTimer`	The interval in seconds following which the Stinger sends another PIM join/prune message.
`Tree(Rpt/Spt)`	With the current software version, the value of this field is always RPT (RP tree).

Example The `pim rp` command displays PIM information for candidate Rendezvous Points (C-RPs) for IP multicast groups. When the local router is the BSR, this information is obtained from received C-RP-Advertisements. When the local router is not the BSR, this information is obtained from received RP-Set messages. For example:

```
super> pim rp
Group           RP-Address      RPF neighbor    Priority   holdtime
224.0.0.0/8       1.1.1.3         1.1.1.3           40        75:62
234.0.0.0/8       1.1.1.3         1.1.1.3           40        75:62
234.3.0.0/16      1.1.1.3         1.1.1.3           40        75:62
```

```
234.4.0.0/16      1.1.1.3         1.1.1.3          40        75:62
234.5.0.0/16      1.1.1.3         1.1.1.3          40        75:62
234.6.0.0/16      1.1.1.3         1.1.1.3          40        75:62
```

Static entries (if any) appear following a `Static Entries` label at the end of the list. Static entries represent the `pim-group-rp-mapping` profile entries. Fields in the command output have the following meaning:

Output field (`pim rp`)	Description
`Group`	The IP multicast group address and group mask for which this entry contains information about the C-RP.
`RP-Address`	IP address of the C-RP.
`RPF neighbor`	IP address of the Reverse Path Forwarding (RPF) neighbor-router to which a join message for this group would be directed to under certain circumstances. The RPF neighbor for an RP is calculated when the Stinger unit receives an IGMP join message for a group range that uses the RP.
`Priority`	Priority of this mapping, to be used in selecting a C-RP if multiple mappings are found for a group range.
`holdtime`	Holdtime of the C-RP. If the Stinger is the BSR, the first value in this field represents the holdtime received in the C-RP message for this RP and the second value shows the time left before this C-RP will be removed. If the Stinger is not the elected BSR, the first value in this field represents the holdtime received in BSR messages and second value will be zero.

Example The `pim nbr` command displays information about PIM neighbors on active PIM interfaces in the Stinger unit. For example:

```
super> pim nbr

Neighbor   Interface   Priority  Holdtime   DR
1.1.101.2    2           2        100       YES
```

Fields in the command output have the following meaning:

Output field (`pim nbr`)	Description
`Neighbor`	The IP address of the PIM neighbor.
`Interface`	The value of `ifIndex` for the interface used to reach this PIM neighbor.
`Priority`	Hello priority of the neighbor. The 0 value indicates that the neighbor does not support the priority option, or the neighbor supports the priority option but has an assigned hello priority of 0.
`Holdtime`	Interval in seconds before the Stinger times out the neighbor if no hello message is received.
`DR`	Whether the neighbor is DR on the interface.

Example The `pim if` command displays information and statistics about the specified PIM interface. For example:

```
super> pim if 1
pimHelloIntvl      30
pimHelloHoldtime   105
pimHelloPriority   1
pimJpIntvl         60
pimJpHoldtime      210
pimLanPruneDelay   5000
pimOdDelay         2500
pimDR              FALSE
genId              22305411
PIM Statistics
        4 packets received
        0 bad checksum packets received
        0 bad version packets received
        3 hello packet received
        0 join/prune packets received
        0 Boot strap packets received
        1 C-RP Adv packets received
        5 packets transmitted
        4 hello packets sent
        0 join/prune packets sent
        1 boot strap packets sent
```

The statistics list the number of various packet types sent or received on this interface. They reach a maximum value and are then reset to zero. The `pimDR` field indicates whether the Stinger unit is DR on the interface. The other fields in `pim if` command output correspond to the `pim-options` configuration for the interface, as described in "PIM options in the ip-interface and connection profiles" on page 9-6.

Example The `pim bsr` command shows information about candidate BSRs within the domain if the Stinger is acting as candidate BSR, otherwise it shows information about the elected BSR. For example, the following output occurs when the Stinger unit has been elected BSR:

```
super> pim bsr
Stinger BSR State : ELECTED_BSR
Details of CURRENT BSR:
BSR IP Address       : 1.1.1.101
BSR Interface        : 1
BSR Priority         : 255
BSR holdtime         : 57
BSR Current Frag Tag : 717
BSR HASH masklen     : 30
```

Fields in the command output have the following meaning:

Output field (`pim bsr`)	Description
`Stinger BSR State`	State of the system relevant to BSR election. If C-BSR is not enabled in the ip-global profile, this field displays `Stinger is not a C-BSR.`

Output field (`pim bsr`)	Description
`BSR IP Address`	The IP address of the bootstrap router (BSR) for the local PIM region.
`BSR Interface`	The value of `ifIndex` for the interface used to reach the BSR.
`BSR Priority`	Priority of the current BSR, from 0 to 255.
`BSR holdtime`	The bootstrap holdtime when the BSR is a C-RP in the local domain. If Stinger is elected BSR, holdtime represents the interval after which next BSM will be sent.
`BSR Current Frag Tag`	The value used in the current BSM to identify fragmented BSMs.
`BSR HASH masklen`	A value (30 by default) used to calculated the hash value for a group range when two RPs have the same priority

Example The `pim hash` command displays the IP address of the best RP for a group or group range. For example:

```
super> pim hash 234.1.1.1
Best RP for group 234.1.1.1 is 1.1.1.10
```

To determine the best RP for a particular group, the system searches the dynamic list of RPs first. If the system does not find an RP in the dynamic list, the Stinger system looks into the static RP list. For dynamic entries, the system selects the best RP on the basis of the longest prefix match, RP priority, and the RP IP address. For static entries, the selection of best RP is based on longest prefix match.

VLAN-related diagnostics

The `brtbls` (bridge tables), `ifmgr`, `diag brtbls`, and `vlanstats` commands are available only in the debug environment. See "Enabling the debug environment" on page A-2 for related information.

brtbls

Description The `brtbls` command supports diagnostics related to VLAN bridge circuits.

Permission level `debug`

Usage `brtbls [-c] |[-i n][-p ifnum] [-s n]`

Command element	Description
`-c`	Show all bridge circuits.
`-i n`	Show interfaces on bridge circuit *n*, or within bridge-group *n*.
`-p ifnum`	Show partner information on *ifnum*.
`-s n`	Show stack user VLANs on bridge circuit *n*.

Example This example displays bridge table information for VLAN bridge circuits.

```
super> brtbls -c
 bridgeGroup  bridgeType   circuit   interfaceList    items on list
     10       VLAN_CKT   0x84019f30       8400f040          1
   1000     VLAN_STACK   0x84016300       82c4d830          3
2 circuits.
```

Example The next command displays interfaces on bridge circuit 11:

```
super> brtbls -i 11
    ifNum      ifType    ifName
     11           135    vlan11
     13            49     wan13
```

The next command lists the interfaces in bridge group 1000:

```
super> brtbls -i 1000
    ifNum      ifType    ifName
     23          49      wan23
     22          49      wan22
     20         135      vlan1000
```

Example The next command shows paired bridge-circuit interface information for interface 13:

```
super> brtbls -p 13
[ifNum 13 iff 80c4aa88] <----> [ifNum 11 iff 80c4a678]
```

ifmgr

Description The `ifmgr` command displays interface-table entries, toggles the debug display, and marks an interface as enabled or disabled. You can enter this command only from the control module.

Permission level `debug`

Usage `ifmgr [options]`

Command element	Description
`[-r vrouter]`	Display routing entries. If a virtual router name is specified on the command line, the command displays only the table of the virtual router. If no virtual router name is specified, the command displays the tables for all virtual routers.

Command element	Description
[-r *vrouter*] -k *ifNum*	Enables manual deletion of a virtual interface on a BIR connection. If *ifNum* is the interface number of a virtual interface on a BIR connection, the command deletes that virtual interface. If *ifNum* is the interface number of the main configured BIR connection, the command deletes all virtual interfaces associated with that interface. (The main BIR connection interface is never deleted through the use of this command.) If *ifNum*is neither the interface number of a virtual interface or a main BIR interface, the command displays an error message.
-d [*ifname*|*ifnum*]	Displays interface table entries.
-n *conn-profile name*	Display slot and interface number of the connection.
[up|down] [*ifnum*|*ifname*]	Enables or disables the specified interface.

Example The `ifmgr -d` command displays the interface table:

```
super> ifmgr -d
bif slot sif  u m p ifname host-name  remote-addr       local-addr
-------------------------------------------------------------------------------
000 1:08 000    *   ie0       -       0.0.0.0/32      134.112.26.132/32
001 1:08 001    *   ie1       -       0.0.0.0/32       201.168.53.123/32
002 1:08 002    *   lo0       -       0.0.0.0/32           127.0.0.1/32
003 0:00 000    *   rj0       -       0.0.0.0/32           127.0.0.2/32
004 0:00 000    *   bh0       -       0.0.0.0/32           127.0.0.3/32
005 1:08 000    *   wanabe    -       0.0.0.0/32           127.0.0.3/32
006 0:00 000    *   local     -       0.0.0.0/32           127.0.0.1/32
007 0:00 000    *   mcast     -       0.0.0.0/32           224.0.0.0/32
008 0:00 000    -   tunnel0   -       0.0.0.0/32       134.112.26.132/32
009 0:00 000    *    vr0_main   -       0.0.0.0/32        134.112.26.132/32
010 0:00 000   -    sip0              0.0.0.0/32              0.0.0.0/32
011 1:06 001  *  p  wan11  adsl-6-1   30.30.7.30/32    134.112.26.132/32
012 1:06 005  *  p  wan12  adsl-6-2   10.10.7.10/32    134.112.26.132/32
013 1:14 0    *     ie1-14-1  -       0.0.0.0/32           60.60.7.60/32
014 1:14 002  *     ie1-14-2  -       0.0.0.0/32              0.0.0.0/32
015 1:06 004  *  p  wan15     pvc     0.0.0.0/32              0.0.0.0/32
016 1:14 062  *  p  wan16     ppp     20.20.7.20/32    134.112.26.132/32
<end>
```

Command element	Description
bif	Bundle interface number. There is one interface number per bundle, including Multilink Protocol Plus (MP+) connections. This number is the global interface-table number.
slot	Shelf and slot to which the interface is assigned.
sif	Slot interface.

Command element	Description
u	Whether the interface is enabled (*) or disabled (-).
m	The interface is part of an Multilink Protocol (MP) bundle.
p	Whether the interface is permanent. A p indicates a permanent interface. A hyphen (-) or a blank indicates that the interface is not permanent. A permanent interface is an interface that is configured in the command-line interface and stored in Stinger NVRAM. All the Ethernet interfaces and the interfaces based on connection profiles are permanent. Transient interfaces are those the Stinger unit builds from RADIUS. These interfaces have no interface entry when the connection is not active.
ifname	Interface name. The name ie1 is the GigE interface.
host-name	Hostname of remote device.
remote-addr	Remote address of device as configured in a connection profile.
local-addr	Address of the local interface.

Example The next command displays information about the Gigabit Ethernet interface:

```
super> ifmgr -d 1
iff             0x82ec86cc
inUse:          Yes
hostName:
dialoutName:
Authentication Source:  In: local       Out: local
ExternFilters:  No
ExternRoutes @  0
miscInfo @      0
reDirectDest:   0.0.0.0
DLCI routeId:   0
MP(P) id:       0
rtIf:   1:08:1
virtual id: 0, virtual next @ -1, virtual main @ -1
minor device:   1
device status:  0x221
output func:    0x801b5f18
input func:     0x801b61a0
mtu:            1500
ip_addr:        201.168.53.123
dstip_addr:     0.0.0.0
netmask:        255.255.255.0
net:            201.168.53.0
subnet:         201.168.53.0
bcast:          201.168.53.255
nbcast:         201.168.53.255
directed-bcast:         yes
```

```
management only:        no
macaddr:        00c07b65d579
inp_qcnt:       0
out_qcnt:       0
nexthop:        0.0.0.0
proxy_arp_mode: 0
proxy_arp_head: 0
vRouterID:      0
if_redirServer: 0.0.0.0
if_redirPort:   0
if_redirPort:   0
ATMP tunnel:    DISABLED
No associated connection profile
SNMP ifType:    6
multicastServiceProfile :
multicastMaxGroups      :       0
```

diag brtbls

Description Enable diagnostic `printf` input for bridge tables.

Permission level `debug`

Usage `diag brtbls`

Example The next commands enable and disable bridge table diagnostics.

```
super> diag brtbls
brtbls                          ( Bridge Tables diagnostic )
brtbls debug is ON
super> diag brtbls
brtbls                          ( Bridge Tables diagnostic )
brtbls debug is OFF
```

vlanstats

Description Display or clear VLAN statistics. For details about the `vlanstats` output fields, see the descriptions of MIB objects in "Virtual LAN (VLAN) statistics tables" on page A-29.

Permission level `debug`

Usage `vlanstats [-c] {shelf slot item} vlan-id`

Command element	Description
`-c`	Clear statistics for the specified VLAN.

Example The following command displays statistics about VLAN 1 on the first controller's GigE interface:

```
super> vlanstats { 1 8 2 } 1
rxOctetsHigh      : 0
rxOctetsLow       : 0
rxFrames          : 0
```

```
rxUnicastFrames   : 0
rxMulticastFrames : 0
rxBroadcastFrames : 0

txOctetsHigh      : 0
txOctetsLow       : 0
txFrames          : 0
txUnicastFrames   : 0
txMulticastFrames : 0
txBroadcastFrames : 0
```

Example The following command clears the statistics for VLAN 1:

```
super> vlanstats -c { 1 8 2 } 1
Statistics for VLAN 1 cleared
```

SAR-related diagnostics

The `sar` command is available only in the debug environment. See "Enabling the debug environment" on page A-2 for related information.

sar

Description Display the Segmentation and Reassembly (SAR) interface and routing tables, protocol statistics, and active sockets.

Permission level `debug`

Usage `sar [`*`options`*`]`

Command element	Description
`-b`	Show free transmission buffer list.
`-c`	Clear SAR statistics.
`-i`	Reset the SAR (destructive).
`-l`	List open channels.
`-p`	Configure transmission packet display.
`-q`	Configure receive packet display.
`-m`	Show SAR control memory information.
`-r`	Read shaper.
`-s`	Show SAR statistics.
`-t`	Execute SAR PHY test (control module).
`-v`	Show SAR virtual circuit table.
`-w`	Write shaper.
`-x`	List open connections.
`-y`	Loopback cell statistics (shelf).
`-z [-d|-o|-i|-c|-a]`	List open WAN connections (Stinger control module).

Example The following command analyzes contents of a frame on Gigabit Ethernet to check the VLAN ID:

```
super> sar -p -a -100
SAR: now dumping the contents of all transmitted packets

super> ping -c 1 20.1.2.10
PING 20.1.2.10 (20.1.2.10): 56 data bytes
tx 1/61(d) @ a1d46b00 packet len  102
TX packet: (task "_brouterPacketTask" at 0x81697c40, time: 8907.26) 102
octets @ 0xa1d46b00
  [0000]: 01 20 01 02 03 04 00 d0  52 01 02 04 81 00 00 64    . ......
R......d
  [0010]: 08 00 45 00 00 54 c9 2a  00 00 ff 01 c6 6e 14 01    ..E..T.*
.....n..
  [0020]: 02 04 14 01 02 0a 08 00  8a 47 23 a5 00 00 60 00    ........
.G#...'.
  [0030]: ea 12 00 00 00 00 00 00  00 00 00 00 00 00 00 00    ........
........
  [0040]: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00    ........
........
  [0050]: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00    ........
........
  [0060]: 00 00 00 00                                         ....
```

Network processor-related diagnostics

The `info` ommand is available only in the debug environment. See "Enabling the debug environment" on page A-2 for related information.

`info np`

Description Display hardware, protocol, and connection information.

Permission level `debug`

Usage `info np` *`option`*

Option	Description
`addr`	Display PayloadPlus addr information
`arp [`*`vrouter`*`]`	Display ARP entries. If a virtual router name is specified on the command line, the command displays only the table of the virtual router. If no virtual router name is specified, the command displays the tables for all virtual routers.
`bridge` *`option`*	`group` *`group-num`* Display bridging group details. `table` *`group-num ifnum`* Display bridging table details.
`conn [`*`slot`*`] [`*`ifnum`*`]`	Display connections.
`did [`*`didnum`*`]`	Display destination ID (DID) table contents.
`ipqos [`*`ifnum`*`]`	Display IP QoS details.

Option	Description
`lns`	Display LNS entries
`mbl`	Display Mobile entries
`mgrp`	Display multicast groups.
`psl` *`lport`* [*`cnt`*] [*`rspslot`*]	Display lport-scheduler information
`qid` *`qidnum`* [*`rspslot`*]	Display Queue information
`rsc`	Display the resources collection list.
`rt` [*`vrouter`*]	Display routing entries. If a virtual router name is specified on the command line, the command displays only the table of the virtual router. If no virtual router name is specified, the command displays the tables for all virtual routers.
`sed` *`didnum`* [*`cnt`*]	Display SED parameters indexed by DID
`sid` *`sidnum`* [*`rspslot`*]	Display scheduler information
`spoof` [*`ifnum`*]	Display spoof details.

Example Without arguments, the `info np conn` command displays all connection handles. For example:

```
super> info np conn
Connection Handles
if  slot act cPort dPort type uctl udata   qos    dctl   ddata  qos
19  2     Y  000001 000001 LIM 0/124 0/123  default 0/124   0/123  default
1   8     Y  000000 000000  GE 0/864 0/864  default 0/864   0/864  default
6   8     Y  000000 000000  GE 0/864 0/864  default 0/864   0/864  default
7   8     Y  000000 000000  GE 0/864 0/864  default 0/864   0/864  default
24  8     Y  000000 000000  GE 0/864 0/864  default 0/864   0/864  default
```

To dump all the connection handles for a given slot, or for a given port on a slot, specify the slot/port numbers on the command line. For example:

```
super>  info np conn 3 20
Connection Handles
if  slot act cPort dPort type uctl udata   qos    dctl   ddata  qos
20  3     Y  000002 000002 LIM 0/128 0/127  default 0/128   0/127  default
Encaps: AAL5-LLC - MTU=1560 (50)

Control & Data Service Queues:
------------------------------
ControlQID                280    Lport              0002
Scheduler                 UBR   CosQueue               1

DataQID                   276    Lport              0002
Scheduler                 UBR   CosQueue               3

PDU Ids:
---------
PduID      port     vpi/vci     Tree
065668    0x0202     0/128       3072
```

```
000133    0x0202     0/127       3072

VLAN Configuration       : Vlan is not enabled

Packet Flow configuration: No Packet Flow is attached

Multicast Configuration  : Multicast Client

Bridge Configuration     : Bridge Group 50 Transparent Bridging Enabled

Filter Configuration     : No Filers applied

DID List:
Did            Type          QId       cosQid
504          CNTL_EX          281          1
505          DATA_EX          277          1
12600        MULTICAST        278          2
13110        MULTICAST        278          2
43526        UNICAST          279          3
```

Example To display all routes on the network processor, use the `info np rt` command. For example:

```
super> info np rt
Local If Routes for vrouter:main
Destination        Gateway         IF          DID   needArp
1.1.1.3/32         -               local       000000   N
9.0.0.0/8          -               ie1-1       000000   N
9.9.9.91/32        -               local       000000   N
9.9.9.92/32        -               local       000000   N
12.0.0.0/8         -               ie1         000000   N
12.12.12.12/32     -               local       000000   N
127.0.0.0/8        -               bh0         000000   N
127.0.0.1/32       -               local       000000   N
127.0.0.2/32       -               rj0         000000   N
135.254.196.0/24   210.210.210.1   ie0         000000   N
210.0.0.0/8        -               ie0         000000   N
210.210.210.75/32  -               local       000000   N
224.0.0.0/4        -               mcast       000000   N
224.0.0.1/32       -               local       000000   N
224.0.0.2/32       -               local       000000   N
224.0.0.9/32       -               local       000000   N
224.0.0.13/32      -               local       000000   N
255.255.255.255/32 -               ie0         000000   N
Route Table for vrouter:main
Destination        Gateway         IF          DID   needArp
1.1.1.0/24         1.1.1.12        wan19       000000   Y
1.1.1.12/32        1.1.1.12        wan19       000000   Y
Total 20 routes <end>
```

Example The `info np mgrp` displays multicast group information. For example:

```
super>  info np mgrp
IP Address  MaxMtu  Member  FirstDid   LastDid   LstInList  NdDrain
224.1.1.1   1610      1      12855      12855      12855       No
```

Example To display IP QoS details, IP QoS monitoring must be enabled on the interface. For information about enabling QoS monitoring, see "Administrative tools for monitoring IP QoS" on page 5-31. The `info np ipqos` command displays information being monitored on the specified interface. For example:

```
super> info np ipqos 19
IP QOS monitoring information for IF 19
IP QOS profile: Pav-Pbit
Number of priority     :4           Lowest priority        :1
Default classification :provided    Total Mcast Rules      :0
Classified DIDs        :3           Classification rules   :4

Mapping priority/COS
P[0]=x P[1]=5 P[2]=x P[3]=x P[4]=4 P[5]=3 P[6]=2 P[7]=x

Main DID list
-------------
DID       Qid      COS    Type
43514     285      5      ucast

Classified DID list
-------------------
DID       Qid      COS    Type
43516     282      2      ucast
43518     283      3      ucast
43520     284      4      ucast
```

Example To display detailed information about a bridge group, use the `info np brig` command. Its argument is the internal bridge group number. For exampe:

```
super> info np brig 1
Number of interfaces in bridge group #50 : 3
Internal bridge group number            : 1
Time left to Mac Aging Timer expiry     : 30

rif  type  bport UDID  FDID  host-name
-----------------------------------------------
0020 wan   0001  43526 12600 minal-3.28
0021 wan   0002  43528 12601 minal-13.4
0022 vlan  0003  43530 12602 vlan22

Flood list pattern :
Type Learn Tree Act   Value    Pattern
BRI    Y   3120  4  00303138   16 48 00000001 00000000

Flood list Information :
list size  num   first  last   lastinlist  Not-drained
-----------------------------------------------------
 0    255     3  12600  12602  12602           0
```

```
Multicast Groups :
GrpIp      Msize Maxmtu Member First  Last LstInList nDrain
225.1.1.1  255      0      1  13110  13110  13110    0
    Type Learn Tree Act Value   Pattern
    BMP    Y   3123  0 00103336 48  0 00000001 E1010101
235.1.1.1  255      0      1  12855  12855  12855    0
    Type Learn Tree Act Value   Pattern
    BMP    Y   3123  0 00103237 48  0 00000001 EB010101
```

Example The `info np brit` command displays bridge table entries. Its arguments are the bridge group number and (optionally) an interface number. For example:

```
super> info np brit 50

Group    PortBlock     if    Did      Destination
-----------------------------------------------------
00050       ROUTER    0022  43530  00:00:00:00:00:09
00050       ROUTER    0022  43530  00:00:00:00:00:0a
00050       ROUTER    0022  43530  00:00:00:00:00:0b
00050       ROUTER    0022  43530  00:00:00:00:00:0c
00050       ROUTER    0022  43530  00:00:00:00:00:0d

Table has 5 entries.
```

SNMP MIB for GMAC and VLAN statistics

The `ip2kstats.mib` MIB gathers statistics about the GMAC interface of the IP2000 control module, and also collects statistics on a per-VLAN basis. It is implemented as the following proprietary Lucent enterprise MIB:

```
ip2kStatsGroup   OBJECT IDENTIFIER ::= { ascend 51 }
```

The transmit and receive statistics represented in this MIB are also accessible in the command-line interface by using the `gmac -d` command. The VLAN statistics are also accessible in the command-line interface by using the debug-level `vlanstats` command.

History maintained at 15-minute intervals

The system maintains a history of the following fields by averaging them at fixed 15-minute intervals for the last 24 hours (96 intervals):

- Tx and Rx octets (transmit and receive traffic streams)
- Multicast traffic (receive side only)
- Unicast traffic (transmit and receive traffic streams)
- Broadcast traffic (receive side only)
- Cyclic redundancy check (CRC) errors (receive side only)

Gigabit Ethernet (GigE) statistics tables

Gigabit Ethernet statistics are represented in five MIB tables:

- GigE configuration
- GigE interval transmit statistics
- GigE interval receive statistics
- GigE total transmit statistics
- GigE total receive statistics

Gigabit Ethernet configuration

The `gigEConfigTable` is a configuration table for the IP2000 GigE interface. It is indexed by the interface index. This MIB table contains the objects shown in Table A-1:

Table A-1. GigEConfigTable MIB objects

MIB object	Description
`gigEValidIntervals`	The number of previous intervals for which data was collected. The value will be 96 unless the interface was brought online within the last 24 hours, in which case the value will be the number of complete 15-minute near-end intervals since the interface has been online.
`gigELastInitTime`	System time when the GigE port was last initialized. The time is represented in text format.

Interval transmit statistics

The `gigETxIntervalTable` is a table containing transmit statistics for the GigE interface in a 15-minute interval. It is indexed by the interface index and the interval number (contained in the `gigETXIntervalNumber` variable). This MIB table contains the objects shown in Table A-2:

Table A-2. GigETxIntervalTable MIB objects

MIB object	Description
`gigETxIntervalNumber`	A number between 1 and 96, where 1 is the most recently completed 15-minute interval and 96 is the 15-minute interval completed 23 hours and 45 minutes prior to interval 1.
`gigETxIntervalOctetsLow` `gigETxIntervalOctetsHigh`	The lower 32 bits and upper 32 bits of the 64-bit interval transmit packet byte counter, which contains a count of how many bytes were transmitted in error-free packets during the interval.
`gigETxIntervalUnicastPackets`	Total number of Unicast packets transmitted during the interval.

Total transmit statistics

The `gigETxTotalTable` is a table containing the total transmit statistics for the GigE interface. It is indexed by the interface index. The transmit counters in this table are also displayed in the output of the `gmac -d` command. See "" on page A-28. This MIB table contains the objects shown in Table A-3:

Table A-3. GigETxTotalTable MIB objects

MIB object	Description
`gigETxTotalOctetsLow` `gigETxTotalOctetsHigh`	The lower 32 bits and upper 32 bits of the 64-bit transmit packet byte counter, which contains a total count of how many bytes have been transmitted in error-free packets.
`gigETxTotalGoodPackets`	Total count of packets transmitted without error.
`gigETxTotalPkt64`	Total count of transmitted 64-byte packets.
`gigETxTotalPkt65To127`	Total count of transmitted packets from 65 to 127 bytes in length.
`gigETxTotalPkt128To255`	Total count of transmitted packets from 128 to 255 bytes in length.
`gigETxTotalPkt256To511`	Total count of transmitted packets from 256 to 511 bytes in length.
`gigETxTotalPkt512To1023`	Total count of transmitted packets from 512 to 1023 bytes in length.
`gigETxTotalPkt1024ToMax`	Total count of transmitted packets from 1024 bytes up to the MAX bytes in length (1536 for non-jumbo packets and 9728 for jumbo packets).[a]
`gigETxTotalPktDefer`	Total count of packets that were deferred because the interframe gap was in excess of 96 bits when a packet was available for transmission. The IFG is a delay between code division multiple access/carrier detect (CDMA/CD) packets, intended to provide interframe recovery time for other CSMA/CD sublayers and for the physical medium.
`gigETxTotalPktUndSz`	Total count of packets less than 64 bytes in length.
`gigETxTotalUnderFlow`	Total count of packets that were truncated because of an empty FIFO.
`gigETxTotalPfcf`	Total count of Pause Flow Control packets that were sent because the receive FIFO exceeded its highwater mark. The Pause function is a mechanism for full duplex flow control.
`gigETxTotalPfcc`	Total count of Pause Flow Control packets that were sent because the client requested them.

Table A-3. GigETxTotalTable MIB objects (Continued)

MIB object	Description
gigETxTotalRfcf	Total count of Reset Flow Control packets that were sent because the receive FIFO went below its lowwater mark.
gigETxTotalRfcc	Total count of Reset Flow Control packets that were sent because the client requested them.
gigETxTotalOverFlow	Total number of packets in which a write from the physical signaling interface was attempted to a full transmit FIFO.
gigETxTotalAlmostFull	Total number of packets in which the transmit FIFO Almost Full flag was set.

a. Jumbo packets are not currently supported on the Gigabit Ethernet interface of the IP2000 controller.

Interval receive statistics

The gigERxIntervalTable is a table containing receive statistics for the GigE interface in a 15 minute interval. It is indexed by the interface index and the interval number (contained in the gigERXIntervalNumber variable). This MIB table contains the objects shown in Table A-4:

Table A-4. GigERxIntervalTable MIB objects

MIB object	Description
gigERxIntervalNumber	A number between 1 and 96, where 1 is the most recently completed 15 minute interval and 96 is the 15 minutes interval completed 23 hours and 45 minutes prior to interval 1.
gigERxIntervalOctetsLow gigERxIntervalOctetsHigh	The lower 32 bits and upper 32 bits of the 64-bit interval receive packet byte counter, which contains a count of how many bytes were received in error free packets during the interval.
gigERxIntervalUnicastPackets	Count of packets of the Unicast Type received during the interval.
gigERxIntervalMulticastPackets	Count of packets of the Multicast Type received during the interval.
gigERxIntervalBroadcastPackets	Count of packets of the Broadcast Type received during the interval.
gigERxIntervalCrcErrors	Count of packets that failed CRC received during the interval.

Total receive statistics

The gigERxTotalTable is a table containing the total receive statistics for the GigE interface. It is indexed by the interface index. The receive counters in this table are

also displayed in the output of the `gmac -d` command. See "" on page A-28. This MIB table contains the objects shown in Table A-5:

Table A-5. GigERxTotalTable MIB objects

MIB object	Description
`gigERxTotalOctetsLow` `gigERxTotalOctetsHigh`	The lower 32 bits and upper 32 bits of the 64-bit receive packet byte counter, which contains a total count of how many bytes have been received in error free packets.
`gigERxTotalGoodPackets`	Total count of packets received without error.
`gigERxTotalPkt64`	Total count of received 64-byte packets.
`gigERxTotalPkt65To127`	Total count of received packets from 65 to 127 bytes in length.
`gigERxTotalPkt128To255`	Total count of received packets from 128 to 255 bytes in length.
`gigERxTotalPkt256To511`	Total count of received packets from 256 to 511 bytes in length.
`gigERxTotalPkt512To1023`	Total count of received packets from 512 to 1023 bytes in length.
`gigERxTotalPkt1024ToMax`	Total count of received packets from 1024 bytes up to the MAX bytes in length (1536 for non-jumbo packets and 9728 for jumbo packets).[a]
`gigERxTotalMacType`	Total count of MAC Type packets received
`gigERxTotalCrcErrors`	Total count of packets received that failed CRC.
`gigERxTotalUnderSize`	Total count of packets that were less than 64 bytes in length.
`gigERxTotalOverSize`	Total count of packets received that were greater than the MAX bytes in length (1536 for non-jumbo packets and 9728 for jumbo packets).
`gigERxTotalAlmostFull`	Total number of packets in which the receive FIFO exceeded its highwater mark.
`gigERxTotalOverRun`	Total number of receive packets in which a write to a full receive FIFO was attempted.
`gigERxTotalMulticastPackets`	Total number of receive packets that were of the Multicast Type.
`gigERxTotalBroadcastPackets`	Total number of receive packets that were of the Broadcast Type.
`gigERxTotalJabber`	Total count of receive packets that were classified as Jabber. Jabber is the condition of abnormally long transmissions, usually due to a fault condition.

Table A-5. GigERxTotalTable MIB objects (Continued)

MIB object	Description
gigERxTotalPfc	Total count of Pause Flow Control packets that were received. The Pause function is a mechanism for full duplex flow control.
gigERxTotalRfc	Total count of Reset Flow Control packets that were received.

a. Jumbo packets are not currently supported on the Gigabit Ethernet interface of the IP2000 controller.

Virtual LAN (VLAN) statistics tables

To capture and clear VLAN statistics, the following MIB tables are supported:

- GigE VLAN statistics
- GigE VLAN clear statistics

VLAN statistics

The GigEVlanStatTable is the VLAN statistics table for the IP2000 GigE interface. It is indexed by the interface index and by VLAN ID (contained in the gigEVlanId variable). The counters in this table are also displayed in the output of the vlanstats debug-level command. See "vlanstats" on page A-18.

This MIB table contains the objects shown in Table A-6:

Table A-6. GigEVlanStatTable MIB objects

MIB object	Description
gigEVlanId	VLAN ID. Identifies a specific VLAN on the GigE interface of the IP2000.
gigEVlanRxOctetsLow	Receive Frame Byte Counter Low Double Word.
gigEVlanRxOctetsHigh	Receive Frame Byte Counter High Double Word.
gigEVlanRxGoodFrames	Receive frame counter. Indicates the total number of frames received on the GigE interface for this VLAN.
gigEVlanRxUnicastFrames	Receive Unicast frame counter. Indicates the total number of unicast frames received for this VLAN.
gigEVlanRxMulticastFrames	Receive Multicast frame counter. Indicates the total number of multicast frames received for this VLAN.
gigEVlanRxBroadcastFrames	Receive Broadcast frame counter. Indicates the total number of broadcast frames received for this VLAN.
gigEVlanTxOctetsLow	Transmit frame byte counter Low Double word.

Table A-6. GigEVlanStatTable MIB objects (Continued)

MIB object	Description
gigEVlanTxOctetsHigh	Transmit Frame Byte Counter High Double word.
gigEVlanTxGoodFrames	Transmit frame counter. Indicates the total number of Ethernet frames transmitted on the GigE interface for this VLAN.
gigEVlanTxUnicastFrames	Unicast transmit frame counter. Indicates the total number of unicast frames transmitted for this VLAN.
gigEVlanTxMulticastFrames	Multicast transmit frame counter. Indicates the total number of multicast frames transmitted for this VLAN.
gigEVlanTxBroadcastFrames	Broadcast transmit frame counter. Indicates the total number of broadcast frames transmitted for this VLAN.

VLAN clear statistics

The GigEVlanClearStatTable is the table for clearing statistics for a given VLAN on the IP2000 GigE interface. It is indexed by the interface index and by VLAN ID (contained in the gigEVlanId variable). This table contains the objects shown in Table A-7:

Table A-7. GigEVlanClearStatTable MIB objects

MIB object	Description
gigEVlanClearStatCmd	Command to clear statistics for a particular VLAN. Values are: none(1)–No action. clearAllStats(1)–Clear all statistics. A GET operation on this object will always return none(1).

PIMv2 MIB support

The system provides SNMP MIB support for the PIM protocol as defined in draft-ietf-pim-mib-v2-01.txt (the PIMv2 MIB). The PIMv2 MIB is placed in the MIB tree under experimental 61. The current software supports PIMv2 MIB tables as shown in Table A-8:

Table A-8. Current level of support for PIMv2 MIB tables

PIMv2 MIB table	Support in this software version
pimInterfaceTable	YES
pimNeighborTable	YES

Table A-8. Current level of support for PIMv2 MIB tables (Continued)

PIMv2 MIB table	Support in this software version
pimIpMRouteTable	NO
pimNextHopGroup	NO
pimRPTable	NO
pimRPSetTable	YES
pimCandidateRPTable	NO
pimComponentTable	YES

The snmpwalk, get, and getnext routines are supported for objects in the supported tables. Because Stinger units operate within a single PIM domain, the component index in pimComponentTable is always set to 1. For example, the following snmpwalk from an SNMP manager displays information from the pimInterfaceTable:

```
$ snmpwalk -m all -O 50.50.50.5 public
experimental.pimMIB.pimMIBObjects.pim.pimInterfaceTable
pimInterfaceAddress.2 = IpAddress: 1.1.1.2
pimInterfaceNetMask.2 = IpAddress: 255.0.0.0
pimInterfaceMode.2 = sparse(2)
pimInterfaceDR.2 = IpAddress: 1.1.1.2
pimInterfaceHelloInterval.2 = 60 seconds
pimInterfaceStatus.2 = active(1)
pimInterfaceJoinPruneInterval.2 = 60 seconds
pimInterfaceTrigHelloInterval.2 = 5 seconds
pimInterfaceHelloHoldtime.2 = 105 seconds
pimInterfaceLanPruneDelay.2 = on(1)
pimInterfacePropagationDelay.2 = 7500 milliseconds
pimInterfaceOverrideInterval.2 = 2500
pimInterfaceGenerationID.2 = on(1)
pimInterfaceJoinPruneHoldtime.2 = 210 seconds
```

The following snmpwalk command displays information from the pimNeighborTable:

```
$ snmpwalk -m all -O 50.50.50.5 public
experimental.pimMIB.pimMIBObjects.pim.pimNeighborTable
pimNeighborIfIndex.167837953 = 1
pimNeighborExpiryTime.167837953 = Timeticks: (98) 0:00:00.98
pimNeighborMode.167837953 = sparse(2)
pimNeighborLanPruneDelay.167837953 = 0
pimNeighborOverrideInterval.167837953 = 0
pimNeighborTBit.167837953 = off(0)
pimNeighborDRPresent.167837953 = true(1)
```

The following command from an SNMP manager displays information from the pimRPSetTable:

```
$ snmpwalk -m all -O s 50.50.50.5 public
experimental.pimMIB.pimMIBObjects.pim.pimRPSetTable
pimRPSetHoldTime.1.224.255.50397441 = 75
pimRPSetHoldTime.1.234.255.50397441 = 75
pimRPSetHoldTime.1.1002.65535.50397441 = 75
pimRPSetHoldTime.1.1258.65535.50397441 = 75
```

```
pimRPSetHoldTime.1.1514.65535.50397441 = 75
pimRPSetHoldTime.1.1770.65535.50397441 = 75
pimRPSetExpiryTime.1.224.255.50397441 = Timeticks: (56) 0:00:00.56
pimRPSetExpiryTime.1.234.255.50397441 = Timeticks: (56) 0:00:00.56
pimRPSetExpiryTime.1.1002.65535.50397441 = Timeticks: (56) 0:00:00.56
pimRPSetExpiryTime.1.1258.65535.50397441 = Timeticks: (56) 0:00:00.56
pimRPSetExpiryTime.1.1514.65535.50397441 = Timeticks: (56) 0:00:00.56
pimRPSetExpiryTime.1.1770.65535.50397441 = Timeticks: (56) 0:00:00.56
```

Following is sample output of an `snmpwalk` on `pimComponentTable`:

```
$ snmpwalk -m all -O s 50.50.50.5 public
experimental.pimMIB.pimMIBObjects.pim.pimComponentTable
pimComponentBSRAddress.1 = IpAddress: 1.1.1.101
pimComponentBSRExpiryTime.1 = Timeticks: (45) 0:00:00.45
pimComponentCRPHoldTime.1 = 0 seconds
pimComponentStatus.1 = active(1)
```

Index

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W