JunosE™ Software for E Series™ Broadband Services Routers

Policy Management Configuration Guide

Release
14.1.x

Published: 2012-12-12

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

JunosE™ Software for E Series™ Broadband Services Routers Policy Management Configuration Guide
Release 14.1.x

Revision History
December 2012—FRS JunosE 14.1.x

The information in this document is current as of the date on the title page.

Abbreviated Table of Contents

Table of Contents

List of Figures

List of Tables

About the Documentation

E Series and JunosE Documentation and Release Notes

For a list of related JunosE documentation, see http://www.juniper.net/techpubs/software/index.html.

If the information in the latest release notes differs from the information in the documentation, follow the *JunosE Release Notes.*

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at http://www.juniper.net/techpubs/.

Audience

This guide is intended for experienced system and network specialists working with Juniper Networks E Series Broadband Services Routers in an Internet access environment.

E Series and JunosE Text and Syntax Conventions

Table 1 on page xx defines notice icons used in this documentation.

Table 1: Notice Icons

Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2 on page xx defines text and syntax conventions that we use throughout the E Series and JunosE documentation.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents commands and keywords in text.	• Issue the **clock source** command. • Specify the keyword **exp-msg**.
Bold text like this	Represents text that the user must type.	**host1(config)#traffic class low-loss1**

`Fixed-width text like this`	Represents information as displayed on your terminal's screen.	**host1#show ip ospf 2**
		`Routing Process OSPF 2 with Router ID 5.5.0.250`
		`Router is an Area Border Router (ABR)`

Italic text like this	• Emphasizes words. • Identifies variables. • Identifies chapter, appendix, and book names.	• There are two levels of access: *user* and *privileged*. • *clusterId, ipAddress.* • *Appendix A, System Specifications*
Plus sign (+) linking key names	Indicates that you must press two or more keys simultaneously.	Press Ctrl + b.
Syntax Conventions in the Command Reference Guide		
Plain text like this	Represents keywords.	terminal length
Italic text like this	Represents variables.	*mask*, *accessListName*

Table 2: Text and Syntax Conventions *(continued)*

Convention	Description	Examples
| (pipe symbol)	Represents a choice to select one keyword or variable to the left or to the right of this symbol. (The keyword or variable can be either optional or required.)	diagnostic | line
[ ] (brackets)	Represent optional keywords or variables.	[internal | external]
[ ]* (brackets and asterisk)	Represent optional keywords or variables that can be entered more than once.	[level1 | level2 | l1]*
{ } (braces)	Represent required keywords or variables.	{ permit | deny } { in | out } { *clusterId* | *ipAddress* }

Obtaining Documentation

To obtain the most current version of all Juniper Networks technical documentation, see the Technical Documentation page on the Juniper Networks Web site at http://www.juniper.net/.

To download complete sets of technical documentation to create your own documentation CD-ROMs or DVD-ROMs, see the Portable Libraries page at

http://www.juniper.net/techpubs/resources/index.html

Copies of the Management Information Bases (MIBs) for a particular software release are available for download in the software image bundle from the Juniper Networks Web site athttp://www.juniper.net/.

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation to better meet your needs. Send your comments to **techpubs-comments@juniper.net**, or fill out the documentation feedback form at https://www.juniper.net/cgi-bin/docbugreport/. If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- JTAC policies—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at http://www.juniper.net/us/en/local/pdf/resource-guides/7100059-en.pdf.
- Product warranties—For product warranty information, visit http://www.juniper.net/support/warranty/.
- JTAC hours of operation—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings: http://www.juniper.net/customers/support/
- Search for known bugs: http://www2.juniper.net/kb/
- Find product documentation: http://www.juniper.net/techpubs/
- Find solutions and answer questions using our Knowledge Base: http://kb.juniper.net/
- Download the latest versions of software and review release notes: http://www.juniper.net/customers/csc/software/
- Search technical bulletins for relevant hardware and software notifications: https://www.juniper.net/alerts/
- Join and participate in the Juniper Networks Community Forum: http://www.juniper.net/company/communities/
- Open a case online in the CSC Case Management tool: http://www.juniper.net/cm/

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: https://tools.juniper.net/SerialNumberEntitlementSearch/

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Management tool in the CSC at http://www.juniper.net/cm/.
- Call 1-888-314-JTAC (1-888-314-5822 toll-free in the USA, Canada, and Mexico).

For international or direct-dial options in countries without toll-free numbers, see http://www.juniper.net/support/requesting-support.html.

PART 1

Policy Management

CHAPTER 1

Managing Policies on the E Series Router

This chapter discusses the following topics:

Policy Management Overview

This chapter introduces policy-based routing management on E Series routers. Policy management enables you to configure, manage, and monitor policies that selectively cause packets to take different paths without requiring a routing table lookup. The JunosE Software's packet-mirroring feature uses secure policies.

Policy management enables network service providers to configure services that customize the treatment of individual packet flows received on a subscriber's interface. The main tool for implementing policy management is a policy list. A policy list is a set of rules, each of which specifies a policy action. A rule is a policy action optionally combined with a classification.

Packets are sorted at ingress or egress into packet flows based on attributes defined in classifier control lists (CLACLs). You can apply policy lists to packets arriving and leaving an interface. You can use policy management on ATM, Frame Relay, generic routing encapsulation (GRE), IP, IPv6, Layer 2 Tunneling Protocol (L2TP), Multiprotocol Label Switching (MPLS), and virtual local area network (VLAN) traffic.

Policy management provides:

- Policy routing—Predefines a classified packet flow to a destination port or IP address. The router does not perform a routing table lookup on the packet. This provides superior performance for real-time applications.
- Bandwidth management—Rate-limits a classified packet flow at ingress to enforce ingress data rates below the physical line rate of a port, A rate-limit profile with a policy rate-limit profile rule provides this capability. You can construct policies to provide rate limiting for individual packet flows or for the aggregate of multiple packet flows. Juniper

Networks E Series Broadband Services Router rate limits are calculated based on the layer 2 packet size. To configure rate limiting, you first create a rate-limit profile, which is a set of bandwidth attributes and associated actions. You next create a policy list with a rule that has rate limit as the action and associate a rate-limit profile with this rule. You can configure rate-limit profiles to provide a variety of services, including tiered bandwidth service where traffic conforming to configured bandwidth levels is treated differently than traffic that exceeds the configured values, and a hard-limit service where a fixed bandwidth limit is applied to a traffic flow. Finally, you can configure rate-limit profiles to provide a TCP-friendly rate-limiting service that works in conjunction with TCP's native flow-control functionality.

- Security—Provides a level of network security by using policy rules that selectively forward or filter packet flows. You can use a filter rule to stop a denial-of-service attack. You can use secure policies to mirror packets and send them to an analyzer.
- RADIUS policy support—Enables you to create and attach a policy to an interface through RADIUS.
- Packet tagging—Enables the traffic-class rule in policies to tag a packet flow so that the Quality of Service (QoS) application can provide traffic-class queuing. Policies can perform both in-band and out-of-band packet tagging.
- Packet forwarding—Allows forwarding of packets in a packet flow.
- Packet filtering—Drops packets in a packet flow.
- Packet mirroring—Uses secure policies to mirror packets and send them to an analyzer.
- Packet logging—Logs packets in a packet flow.

Policy management gives you the CLI tools to build databases, which can then be drawn from to implement a policy. Each database contains global traffic specifications. When building a policy, you specify input from one or more of these databases and then attach the policy to an interface. By combining the information from the various databases into policies, you can deploy a wide variety of services.

NOTE: When applying policies to interfaces that are managed by the SRC, avoid using any other policy management tools, such as CLI, RADIUS, CoA, or Service Manager. SRC is not compatible with other types of policy management tools. When policies are applied to the interface before SRC management begins, such as at access-accept time, these policies are properly replaced. However, if other policy managers change existing policies while SRC management is active, problems can occur. The precedence of each source when modifying configurations is:

- If you have a pre-configured policy through CLI as part of subscriber PVC/VLAN provisioning, SRC overwrites the policy when the SRC manages the interface
- If you have a policy in the Access-Accept, SRC overwrites the policy when the SRC manages the interface

Related Documentation

- Description of a Policy on page 5
- Monitoring Policy Management Overview on page 197
- Policy Management Configuration Tasks on page 6

Description of a Policy

A policy is a condition and an action that is attached to an interface. The condition and action cause the router to handle the packets passing through the interface in a certain way. A policy can be attached to IP interfaces and certain layer 2 interfaces such as Frame Relay, L2TP, MPLS, and VLAN interfaces. The policies do not need to be the same in both directions.

Packets are sorted at ingress or egress into packet flows based on attributes defined in classifier control lists. Policy lists contain rules that associate actions with these CLACLs. A rule is a policy action optionally combined with a classification.

When packets arrive on an interface, you can have a policy evaluate a condition before the normal route lookup; this kind of policy is known as an input policy. You can also have conditions evaluated after a route lookup; this kind of policy is known as a secondary input policy. You can use secondary input policies to defeat denial-of-service attacks directed at a router's local interface or to protect a router from being overwhelmed by legitimate local traffic. If you have a policy applied to packets before they leave an interface, this is known as an output policy.

Classification is the process of taking a single data stream in and sorting it into multiple output substreams. The classifier engine on an E Series router is a combination of PowerPC processors, working with a Field Programmable Gate Array (FPGA) for a hardware assist.

In the Differentiated Services (DiffServ) architecture, two basic types of classifiers exist. The first classifier type is a multifield (MF) classifier, which examines multiple fields in the IP datagram header to determine the service class to which a packet belongs. The second type of classifier is a behavior aggregate (BA) classifier, which examines a single field in an IP datagram header and assigns the packet to a service class based on what it finds.

There are two categories of hardware classifiers, depending on the type of line module being used. ES2 4G LM, ES2 10G Uplink LM, ES2 10G LM, OC48/STM16, GE-2, and GE-HDE line modules support content-addressable memory (CAM) hardware classifiers—all other line modules support FPGA hardware classifiers.

The maximum number of policies that you can attach to interfaces on an E Series router depends on the classifier entries that make up the policy and the number of attachment resources available on the interface. JunosE Software allocates interface attachment resources when you attach policies to interfaces. E Series routers support software and hardware classifiers. A policy can be made up of any combination of software and hardware classifiers.

Related Documentation

- Policy Management Overview on page 3

Policy Platform Considerations

Policy services are supported on all E Series routers.

For information about the modules supported on E Series routers:

- See the *ERX Module Guide* for modules supported on ERX7xx models, ERX14xx models, and the Juniper Networks ERX310 Broadband Services Router.
- See the *E120 and E320 Module Guide* for modules supported on the Juniper Networks E120 and E320 Broadband Services Routers.

Policy References

For more information about policy management, see the following resources:

- RFC 2474—Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers (December 1998)
- RFC 2475—An Architecture for Differentiated Services (December 1998)
- RFC 2697—A Single Rate Three Color Marker (September 1999)
- RFC 2698—A Two Rate Three Color Marker (September 1999)
- RFC 3198—Terminology for Policy-Based Management (November 2001)

Policy Management Configuration Tasks

Perform the required tasks and also any optional tasks that you need for your policy management configuration:

1. Create a CLACL (optional).

   See “Classifier Control Lists Overview” on page 9
2. Create a rate-limit profile (optional).

   See “Creating Rate-Limit Profiles” on page 79
3. Create a policy list.

   See “Policy Lists Overview” on page 17
4. Create a classifier group.

   See “Classifier Groups and Policy Rules Overview” on page 31
5. Create one or more policy rules within the classifier group.

   See “Rate Limits for Interfaces Overview” on page 64
6. Apply a policy list to an interface or profile.

   See “Classifier Groups and Policy Rules Overview” on page 31

Related Documentation

- Policy Management Overview on page 3
- Description of a Policy on page 5
- Monitoring Policy Management Overview on page 197

CHAPTER 2

Creating Classifier Control Lists for Policies

This chapter provides information for configuring policy-based routing management on E Series routers. See the *E120 and E320 Module Guide* for modules supported on the E120 and E320 Broadband Services Routers. The chapter discusses the following topics:

Classifier Control Lists Overview

Classifier control lists (CLACLs) specify the criteria by which the router defines a packet flow. Table 3 on page 9 lists the criteria that you can use to create CLACLs for different types of traffic flows.

Table 3: CLACL Criteria

Type of CLACL	Criteria
ATM	• CLP • Color • Traffic class • User packet class
Frame Relay	• Color • Mark discard eligibility (DE) bit • Traffic class • User packet class

Table 3: CLACL Criteria *(continued)*

Type of CLACL	Criteria
GRE	• Color • Traffic class • Type-of-service (ToS) byte • User packet class
IP	• Color • Destination IP address • Destination port • Destination route class • Internet Control Message Protocol (ICMP) • Internet Gateway Management Protocol (IGMP) • IP flags • IP fragmentation offset • Locally destined traffic • Protocol • Source IP address • Source port • Source route class • Transmission Control Protocol (TCP) • Traffic class • Type-of-service (ToS) byte • User Datagram Protocol (UDP) • User packet class
IPv6	• Color • Destination IPv6 address • Destination port • Destination route class • Internet Control Message Protocol version 6 (ICMPv6) • IPv6 traffic class • Locally destined traffic • Multicast Listener Discovery (MLD) • Next header • Source IPv6 address • Source port • Source route class • Traffic class • Transmission Control Protocol (TCP) • User Datagram Protocol (UDP) • User packet class
L2TP	• Color • Traffic class • User packet class

Table 3: CLACL Criteria *(continued)*

Type of CLACL	Criteria
MPLS	• Color • Mark experimental (EXP) bit • Traffic class • User packet class
VLAN	• Color • Traffic class • User packet class • User priority

You configure CLACLs with a name and then values to match in the IP datagram header. A CLACL does not include an action: actions take place when a match is included in a policy list.

NOTE: Do not use the asterisk (*) for the name of a classifier list. The asterisk is used as a wildcard for the **classifier-group** command.

If you do not specify one of the **frame-relay**, **gre-tunnel**, **ip**, **ipv6**, **l2tp**, **mpls**, or **vlan** keywords, the router creates an IP classifier list. This version of the command has been deprecated and may be removed in a future release.

Related Documentation

- Policy Resources Overview on page 163
- Monitoring Policy Management Overview on page 197

Creating or Modifying Classifier Control Lists for ATM Policy Lists

You can create or modify a classifier control list that can be used only in ATM policy lists.

- Issue the **atm classifier-list** command:

  host1(config)#**atm classifier-list atmclassifier color red user-packet-class 10 clp 1**

Related Documentation

- Classifier Control Lists Overview on page 9
- Policy Lists Overview on page 17

- atm classifier-list

Creating or Modifying Classifier Control Lists for Frame-Relay Policy Lists

You can create or modify a classifier control list that can be used only in Frame Relay policy lists.

- Issue the **frame-relay classifier-list** command.;

```
host1(config)#frame-relay classifier-list frclassifier color red user-packet-class 10
  de-bit 1
```

Related Documentation

- Classifier Control Lists Overview on page 9
- Policy Lists Overview on page 17
- frame-relay classifier-list

Creating or Modifying Classifier Control Lists for GRE Tunnel Policy Lists

You can create or modify a classifier control list that can be used only in GRE tunnel policy lists.

- Issue the **gre-tunnel classifier-list** command:

```
host1(config)#gre-tunnel classifier-list greClassifier50 color yellow  user-packet-class
  7 dsfield 40
```

Related Documentation

- Classifier Control Lists Overview on page 9
- Policy Lists Overview on page 17
- gre-tunnel classifier-list

Creating or Modifying Classifier Control Lists for IP Policy Lists

Tasks to create or modify classifier control lists for IP policy lists:

Creating Classifier Control List for Only IP Policy Lists

You can create or modify a classifier control list that can be used only in IP policy lists. The behavior of multiple-element classifier-list classification is the logical OR of the elements in the CLACL.

- Issue the **ip classifier-list** command to match all packets that have a source IP address of 192.168.30.100 or have a destination IP address of 192.168.30.200:

```
host1(config)#ip classifier-list boston5 ip host 192.168.30.100 any
host1(config)#ip classifier-list boston5 ip any host 192.168.30.200
```

Setting Up an IP Classifier Control List to Accept Traffic from All Sources

You can set up a CLACL to accept IP traffic from all source addresses on the subnet.

- Issue the **ip classifier-list** command:

```
host1(config)#ip classifier-list XYZCorpPermit ip 192.168.0.0 0.0.255.255 any
```

Classifying IP Traffic Based on Source and Destination Addresses

You can classify traffic based on source and destination addresses, You can specify the address as a host address, or a subnet with a wildcard. If you specify the address as a subnet, the mask, in binary notation, must be a series of contiguous zeros, followed by a series of contiguous ones. The **any** keyword is the address wildcard, matching traffic for any address.

- Issue the **ip classifier-list** command to classify traffic on any source or destination address:

```
host1(config)#ip classifier-list YourListName ip any any
host1(config)#ip classifier-list YourListName ip host 10.10.10.10 any
host1(config)#ip classifier-list YourListName ip 10.10.0.0 0.0.255.255 host  10.10.10.2
```

Using IP Classifier Control Lists to Match Route Class Values

You can set up classifier control lists to match route-class values. In this example, svale20 matches the source address lookup route-class value of 1, svale30 matches the destination address lookup route-class value of 1 and a ToS byte value of 10, svale40 matches the source address lookup route-class value of 1 and the packets destined to a local interface, and west20 matches the source address lookup route-class value of 1 and packets that are not destined for a local interface (packets destined for remote interfaces).

- Issue the **ip classifier-list** command:

```
host1(config)#ip classifier-list svale20 source-route-class 1 ip any any
host1(config)#ip classifier-list svale30 destination-route-class 1 ip any any
  tos 10
host1(config)#ip classifier-list svale40 source-route-class 1 local true ip any any
host1(config)#ip classifier-list west20 source-route-class 1 local false ip any any
```

Creating IP Classifier Control Lists for TCP and UDP Ports

You can specify a single TCP or UDP port or a range of ports, where packets are matched with source address 198.168.30.100 and UDP source port numbers in the range 1–10.

- Issue the **ip classifier-list** command to create a CLACL on a UDP host:

```
host1(config)#ip classifier-list YourListName udp host 192.168.30.100 range 1  10 any
```

  To create a CLACL that matches all traffic on UDP source ports greater than 100:

```
host1(config)#ip classifier-list XYZCorpUdp udp any gt 100 172.17.2.1 0.0.255.255
```

To match a non-TCP packet originating from IP address 172.28.100.52:

```
host1(config)#ip classifier-list YourListName not tcp host 172.28.100.52 any
```

To specify a single TCP or UDP port or range of ports, an ICMP code and optional type, or an IGMP type, which matches packets with source address 198.168.30.100 and ICMP type 2 and code 10:

```
host1(config)#ip classifier-list YourListName icmp host 192.168.30.100 any 2 10
```

Creating an IP Classifier Control List That Matches the ToS Byte

You can create an IP CLACL that matches the ToS byte in the IP header.

- Issue the **ip classifier-list** command using the **tos** keyword.

```
host1(config)#ip classifier-list tos128 ip any any tos 128
host1(config)#ip classifier-list low-drop-prec ip any any dsfield 10
host1(config)#ip classifier-list priority ip any any precedence 1
```

Creating an IP Classifier Control List That Filters ICMP Echo Requests

You can create a CLACL that filters all ICMP echo requests headed toward an access link under a denial-of-service attack.

- Issue the **ip classifier-list** command:

```
host1(config)#ip classifier-list XYZCorpIcmpEchoReqs icmp any any 8 0
host1(config)#ip classifier-list XYZCorpIgmpType1 icmp any any
```

Creating IP Classifier Control Lists That Use TCP or IP Flags

You can create CLACLs that use TCP or IP flags. For both IP flags and TCP flags, if you specify only a single flag, the logical equation does not require quotation marks.

- Issue the **ip classifier-list** command with the **tcp-flags** keyword and a logical equation (a quotation-enclosed string using ! for NOT, & for AND) to match one or more of the **ack, fin, psh, rst, syn, or urg TCP flags**:

```
host1(config)#ip classifier-list telnetConnects tcp 192.168.10.0 0.0.0.255 host
  10.10.10.10 eq 23 tcp-flags "syn & !ack"
```

- Issue the **ip classifier-list** command with the **ip-flags** keyword and a logical equation (a quotation-enclosed string using ! for NOT, & for AND) to match one or more of the **dont-fragment**, **more-fragments**,, or **reserved** IP flags:

```
host1(config)#ip classifier-list dontFragment ip any any ip-flags "dont-fragment"
```

Creating IP Classifier Control Lists That Match the IP Fragmentation Offset

You can create CLACLs that match the IP fragmentation offset.

- Issue the **ip classifier-list** command with the **ip-frag-offset** keyword and the **eq** or **gt** operator to match an IP fragmentation offset equal to 0, 1, or greater than 1:


```
host1(config)#ip classifier-list fragOffsetAttack ip any host 10.10.10.10  ip-frag-offset
  eq 1
host1(config)#ip policy-list dosProtect
host1(config-policy-list)#filter classifier-group fragOffsetAttack
host1(config-policy-list)#forward
```

Related Documentation

- Classifier Control Lists Overview on page 9
- ip classifier-list

Creating or Modifying Classifier Control Lists for IPv6 Policy Lists

You can create or modify a classifier control list that can be used only in IPv6 policy lists.

- Issue the **ipv6 classifier-list** command:

```
host1(config)#ipv6 classifier-list ipv6classifier color red user-packet-class 5  tcfield 10
host1(config)#ipv6 classifier-list YourListName udp destination-port eq 75
host1(config)#ipv6 classifier-list telnetConnects tcp destination-port eq 23  tcp-flags
  "syn & !ack"
host1(config)#ipv6 classifier-list listname icmpv6 icmp-type 3 icmp-code 6
host1(config)#ipv6 classifier-list listname icmpv6 icmp-type 3
host1(config)#ipv6 classifier-list svale20 source-route-class 1
host1(config)#ipv6 classifier-list svale30 destination-route-class 1 tcfield 10
host1(config)#ipv6 classifier-list svale40 source-route-class 1 local true
host1(config)#ipv6 classifier-list west25 source-route-class 1 local false
host1(config)#ipv6 classifier-list YourClaclList source-host 2001:db8:1::8001
  destination-address 2001:db8:3::/48
```

Related Documentation

- Classifier Control Lists Overview on page 9
- Policy Lists Overview on page 17
- ipv6 classifier-list

Creating or Modifying Classifier Control Lists for L2TP Policy Lists

You can create or modify a classifier control list that can be used only in L2TP policy lists.

- Issue the **l2tp classifier-list** command:

```
host1(config)#l2tp classifier-list l2tpclassifier color red user-packet-class 7
```

Related Documentation

- Classifier Control Lists Overview on page 9
- Policy Lists Overview on page 17
- l2tp classifier-list

Creating or Modifying Classifier Control Lists for MPLS Policy Lists

You can create or modify a classifier control list that can be used only in MPLS policy lists.

- Issue the **mpls classifier-list** command:

```
host1(config)#mpls classifier-list mplsClass user-packet-class 10 exp-bits 3 exp-mask
  5
```

Related Documentation

- Classifier Control Lists Overview on page 9
- Policy Lists Overview on page 17
- mpls classifier-list

Creating or Modifying Classifier Control Lists for VLAN Policy Lists

You can create or modify a classifier control list that can be used only in VLAN policy lists.

- Issue the **vlan classifier-list** command:

```
host1(config)#vlan classifier-list lowLatencyLowDrop user-priority 7
host1(config)#vlan classifier-list lowLatencyLowDrop user-priority 6
host1(config)#vlan classifier-list lowLatency user-priority 5
host1(config)#vlan classifier-list excellentEffort user-priority 4
host1(config)#vlan classifier-list bestEffort user-priority 3
host1(config)#vlan classifier-list bestEffort user-priority 2
host1(config)#vlan classifier-list bestEffort user-priority 1
host1(config)#vlan classifier-list bestEffort user-priority 0
```

NOTE: You cannot configure classifier control lists (CLACLs) for policy lists to be attached to VLAN interfaces, without specifying the criteria by which the router defines a packet flow. Although the carriage return, **<cr>**, option is displayed when you type a question mark (**?**) after entering the **vlan classifier list** *classifierName* command without defining any other keyword or CLACL criterion, an error message is displayed when you press **Enter** to configure the VLAN CLACL with only the name. You must specify at least one criterion for the VLAN CLACL to be successfully configured.

Related Documentation

- Classifier Control Lists Overview on page 9
- Policy Lists Overview on page 17
- vlan classifier-list

CHAPTER 3

Creating Policy Lists

This chapter provides information for configuring policy lists on E Series Broadband Services Routers. See the *E120 and E320 Module Guide* for modules supported on the E120 and E320 Broadband Services Routers. The chapter discusses the following topics:

Policy Lists Overview

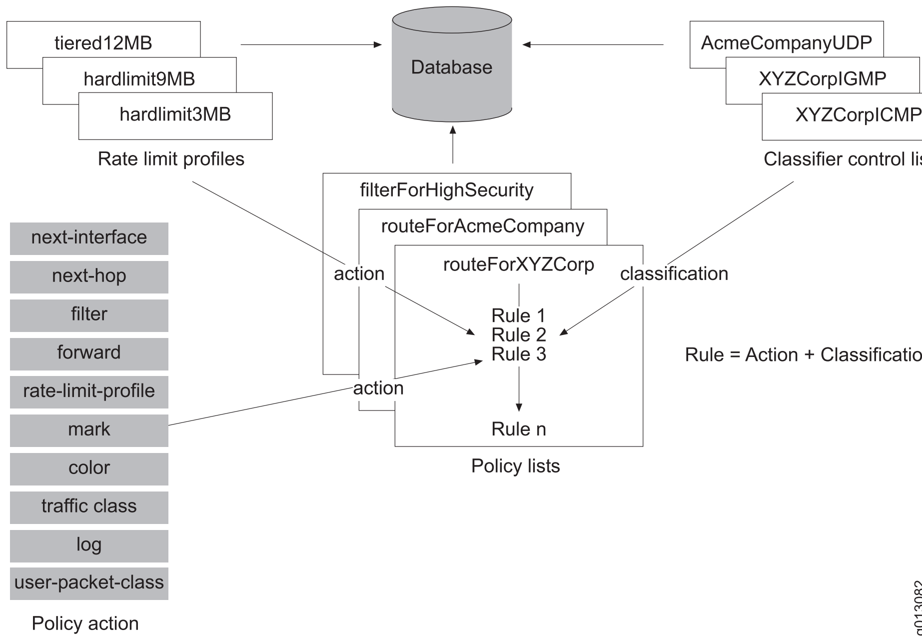
You create a policy rule by specifying a policy action within a classifier group that references a CLACL. These rules become part of a policy list that you can attach to an interface as either an input policy, secondary-input policy, or output policy. The router applies the rules in the attached policy list to the packets traversing that interface.

You can apply policy lists to packets:

- Arriving at an interface (input policy); on IP and IPv6 interfaces the packets arrive before route lookup
- Arriving at the interface, but after route lookup (secondary input policy); secondary input policies are supported only on IP and IPv6 interfaces
- Leaving an interface (output policy)

Figure 1 on page 18 shows how a sample IP policy list is constructed.

Figure 1: Constructing an IP Policy List



You can create a policy list with an unlimited number of classifier groups, each containing an unlimited number of rules. These rules can reference up to 512 classifier entries.

If you enter a **policy-list** command and then enter **exit**, the router creates a policy list with no rules. If the router does not find any rules in a policy, it inserts a default filter rule. Attaching this policy list to an interface filters all packets on that interface.

NOTE: If you do not specify one of the **frame-relay**, **gre-tunnel**, **ip**, **ipv6**, **l2tp**, **mpls**, or **vlan** keywords, the router creates an IP policy list. This version of the command has been deprecated and may be removed in a future release.

You can create policy lists for ATM, Frame Relay, IP, IPv6, GRE tunnels, L2TP, MPLS, and VLANs.

NOTE: Commands that you issue in Policy Configuration mode do not take effect until you exit from that mode.

Related Documentation

- Classifier Control Lists Overview on page 9
- Monitoring Policy Management Overview on page 197

Statistics Collection for Output Policies on Tunnel Interfaces Overview

You can configure the policy manager application to collect and store statistical counters for output policies attached to tunnel interfaces as a measure of the number of fragments.

In certain network environments, it might be useful to monitor and track the outgoing traffic from a tunnel interface to which policies are applied in terms of number of fragments, instead of monitoring the outgoing policed traffic in terms of number of packets. Based on the topology needs and management of services for subscribers, you can configure statistics for traffic on tunnel interfaces with output policies to be counted as either numbers of fragments or numbers of packets.

Related Documentation

- Configuring Statistics Collection for Output Policies on Tunnel Interfaces on page 29
- Verifying Statistics Collection for Output Policies on Tunnel Interfaces on page 231

- enable-frag-stats
- show enable-frag-stats

Creating Policy Lists for ATM

In the following example, you create two policies: one for CBR traffic and one for UBR traffic. One policy is attached to an interface that contains CBR traffic and the other to an interface that contains UBR traffic.

1. Create a CBR policy list.

   ```
   host1(config)#atm policy-list polCbr
   host1(config-policy-list)#
   ```

2. Create the classification group and assign a strict priority traffic class and color green.

   ```
   host1(config-policy-list)#classifier-group *
   host1(config-policy-list-classifier-group)#traffic-class strict-priority
   host1(config-policy-list-classifier-group)#color green
   ```

3. Exit Policy List Configuration mode to save the configuration.

   ```
   host1(config-policy-list-classifier-group)#exit
   host1(config-policy-list)#exit
   host1(config)#
   ```

4. Create a UBR policy that maps to the strict best-effort traffic class and color red.

   ```
   host1(config)#atm policy-list polUbr
   host1(config-policy-list)#classifier-group *
   host1(config-policy-list-classifier-group)#traffic-class best-effort
   host1(config-policy-list-classifier-group)#color red
   ```

5. Exit Policy List Configuration mode to save the configuration.

   ```
   host1(config-policy-list-classifier-group)#exit
   host1(config-policy-list)#exit
   host1(config)#
   ```

6. Attach the policies to ATM subinterfaces.

   ```
   host1(config)#interface atm 0/0.100
   host1(config-if)#atm policy input polUbr statistics enabled
   host1(config-if)#exit
   host1(config)#interface atm 0/0.101
   host1(config-if)#atm policy input polCbr statistics enabled
   ```

host1(config-if)#exit

7. Display the policy lists.

host1#**show atm subinterface atm 0/0.100**

```
Circuit                  Interface
 Interface  ATM-Prot VCD VPI VCI  Type   Encap MTU  Status   Type
----------- -------- --- --- --- ------- ----- ---- ------ ---------
ATM 0/0.100 RFC-1483 100   0 100 PVC     SNAP  9180 up     Static

Auto configure status         : static
Auto configure interface(s)   : none
Detected 1483 encapsulation   : none
Detected dynamic interface    : none
Interface types in lockout    : none

Assigned profile (IP)         : none assigned
Assigned profile (BridgedEnet): none assigned
Assigned profile (PPP)        : none assigned
Assigned profile (PPPoE)      : none assigned
Assigned profile (any)        : none assigned

SNMP trap link-status: disabled

InPackets:                0
InBytes:                  0
OutPackets:               0
OutBytes:                 0
InErrors:                 0
OutErrors:                0
InPacketDiscards:         0
InPacketsUnknownProtocol: 0
OutDiscards:              0
ATM policy input polUbr
  Statistics are disabled
1 interface(s) found
```

host1#**show atm subinterface atm 0/0.101**

```
                                 Circuit                  Interface
 Interface  ATM-Prot VCD VPI VCI  Type   Encap MTU  Status   Type
----------- -------- --- --- --- ------- ----- ---- ------ ---------
ATM 0/0.101 RFC-1483 101   0 101 PVC     SNAP  9180 up     Static

Auto configure status         : static
Auto configure interface(s)   : none
Detected 1483 encapsulation   : none
Detected dynamic interface    : none
Interface types in lockout    : none

Assigned profile (IP)         : none assigned
Assigned profile (BridgedEnet): none assigned
Assigned profile (PPP)        : none assigned
Assigned profile (PPPoE)      : none assigned
Assigned profile (any)        : none assigned

SNMP trap link-status: disabled

InPackets:                0
InBytes:                  0
OutPackets:               0
OutBytes:                 0
InErrors:                 0
```



```
OutErrors:                 0
InPacketDiscards:          0
InPacketsUnknownProtocol:  0
OutDiscards:               0
ATM policy input polCbr
  classifier-group *
    3096 packets, 377678 bytes
    traffic-class best-effort
    color green
1 interface(s) found
```

Related Documentation

- Creating or Modifying Classifier Control Lists for ATM Policy Lists on page 11
- atm policy-list

Creating Policy Lists for Frame Relay

The following example creates a Frame Relay policy that on egress marks the DE bit to 1, and on ingress colors frames with a DE bit of 1 as red.

1. Create the policy list used to mark egress traffic, then create the classifier group for packets conforming to CLACL frMatchDeSet. Add a rule that marks the DE bit as 1.

   host1(config)#**frame-relay policy-list frOutputPolicy**
   host1(config-policy-list)#**classifier-group frMatchDeSet**
   host1(config-policy-list-classifier-group)#**mark-de 1**
   host1(config-policy-list-classifier-group)#**exit**
   host1(config-policy-list)#**exit**

2. Create the policy list used for the ingress traffic. and create the classifier group conforming to CLACL frMatchDeSet. Add a rule that colors the ingress traffic.

   host1(config)#**frame-relay policy-list frInputPolicy**
   host1(config-policy-list)#**classifier-group frGroupA**
   host1(config-policy-list-classifier-group)#**color red**
   host1(config-policy-list-classifier-group)#**exit**
   host1(config-policy-list)#**exit**

3. Apply the policy lists.

   host1(config)#**interface serial 5/0:1/1.1**
   host1(config-subif)#**frame-relay policy output frOutputPolicy statistics enabled**
   host1(config-subif)#**ip address 10.0.0.1 255.255.255.0**
   host1(config-subif)#**exit**
   host1(config)#**interface serial 5/1:1/1.1**
   host1(config-subif)#**frame-relay policy input frInputPolicy statistics enabled**
   host1(config-subif)#**exit**

4. Display interface information to view the applied policies.

   host1#**show frame-relay subinterface**

```
Frame relay sub-interface SERIAL5/0:1/1.1, status is up
Number of sub-interface down transitions is 0
Time since last status change 03:04:59
No baseline has been set
  In bytes: 660                      Out bytes: 660
  In frames: 5                       Out frames: 5
```



```
  In errors: 0                     Out errors: 0
  In discards: 0                   Out discards: 0
  In unknown protos: 0
Frame relay policy output frOutputPolicy
  classifier-group frGroupA entry 1
     5 packets, 640 bytes
     mark-de 1
Frame relay sub-interface SERIAL5/1:1/1.1, status is up
Number of sub-interface down transitions is 0
Time since last status change 03:05:09
No baseline has been set
  In bytes: 660                    Out bytes: 660
  In frames: 5                     Out frames: 5
  In errors: 0                     Out errors: 0
  In discards: 0                   Out discards: 0
  In unknown protos: 0
Frame relay policy input frInputPolicy
  classifier-group frMatchDeSet entry 1
     5 packets, 660 bytes
     color red
```

5. Display the classifier list.

   host1#**show classifier-list detailed**

```
                         Classifier Control List Table
                         ---------- ------- ---- -----
Frame relay Classifier Control List frMatchDeSet
   Reference count:      1
   Entry count:          1

   Classifier-List frMatchDeSet Entry 1
      DE Bit:                     1
```

6. Display the policy lists.

   host1#**show policy-list**

```
                                  Policy Table
                                  ------ -----

Frame relay Policy frOutputPolicy
   Administrative state: enable
   Reference count:      0
   Classifier control list: frMatchDeSet, precedence 100
      mark-de 1

Frame relay Policy frInputPolicy
   Administrative state: enable
   Reference count:      0
   Classifier control list: frGroupA, precedence 100
     color red
```

Related Documentation

- Creating or Modifying Classifier Control Lists for Frame-Relay Policy Lists on page 11
- frame-relay policy-list

Creating Policy Lists for GRE Tunnels

The following example creates a GRE tunnel policy list named routeGre50. For information about creating the CLACL used in this example, see the previous sections.

1. Create the policy list routeGre50.

   host1(config)#**gre-tunnel policy-list routeGre50**

2. Create the classification group for the CLACL named gre8 and assign a precedence of 150 to it.

   host1(config-policy-list)#**classifier-group gre8 precedence 150**
   host1(config-policy-list-classifier-group)#

3. Add two rules for traffic based on the CLACL named gre8: one rule to color packets as red, and a second rule that specifies the ToS DS field value to be assigned to the packets.

   host1(config-policy-list-classifier-group)#**color red**
   host1(config-policy-list-classifier-group)#**mark dsfield 20**
   host1(config-policy-list-classifier-group)#

4. Exit Policy List Configuration mode to save the configuration.

   host1(config-policy-list-classifier-group)#**exit**
   host1(config-policy-list)#**exit**
   host1(config)#

5. Display the policy list.

   host1#**show policy-list routeGre50**

```
                                  Policy Table
                                  ------ -----
GRE Tunnel Policy routeGre50
   Administrative state: enable
   Reference count:      0
   Classifier control list: gre8, precedence 150
      color red
      mark dsfield 20
```

Related Documentation

- Creating or Modifying Classifier Control Lists for GRE Tunnel Policy Lists on page 12
- gre-tunnel policy-list

Creating Policy Lists for IP

The following example creates an IP policy list named routeForABCCorp. For information about creating the CLACLs and rate-limit profile used in this example, see the previous sections.

1. Create the policy list routeForABCCorp.

   host1(config)#**ip policy-list routeForABCCorp**
   host1(config-policy-list)#

2. Create the classification group for the CLACL named ipCLACL10 and assign the precedence to the classification group.

   host1(config-policy-list)#**classifier-group ipCLACL10 precedence 75**
   host1(config-policy-list-classifier-group)#

3. Add a rule that specifies a group of forwarding solutions based on classifier list ipCLACL10.

   host1(config-policy-list-classifier-group)#**forward next-hop 192.0.2.12 order 10**
   host1(config-policy-list-classifier-group)#**forward next-hop 192.0.100.109 order 20**
   host1(config-policy-list-classifier-group)#**forward next-hop 192.120.17.5 order 30**
   host1(config-policy-list-classifier-group)#**forward interface ip 3/1 order 40**

4. Add a rule that sets a ToS byte value of 125 for packets based on classifier list ipCLACL10.

   host1(config-policy-list-classifier-group)#**mark tos 125**

5. Add a rule that uses rate-limit profile ipRLP25.

   host1(config-policy-list-classifier-group)#**rate-limit-profile ipRLP25**

6. Exit Classifier Group Configuration mode for ipCLACL10, then create a new classification group for classifier list ipCLACL20. Add a rule that filters packets based on classifier list ipCLACL20.

   host1(config-policy-list-classifier-group)#**exit**
   host1(config-policy-list)#**classifier-group ipCLACL20 precedence 125**
   host1(config-policy-list-classifier-group)#**filter**

7. Exit Policy List Configuration mode to save the configuration.

   host1(config-policy-list-classifier-group)#**exit**
   host1(config-policy-list)#**exit**
   host1(config)#

8. Display the policy list.

   host1#**show policy-list routeForABCCorp**

```
                                  Policy Table
                                  ------ -----
IP Policy routeForABCCorp
   Administrative state: enable
   Reference count:      0
   Classifier control list: ipCLACL10, precedence 75
      forward
         Virtual-router: default
         List:
           next-hop 192.0.2.12, order 10, rule 2 (active)
           next-hop 192.0.100.109, order 20, rule 3 (reachable)
           next-hop 192.120.17.5, order 30, rule 4 (reachable)
           interface ip3/1, order 40, rule 5
      mark tos 125
      rate-limit-profile ipRLP25
   Classifier control list: ipCLACL20, precedence 125
      filter
```

Related Documentation

- Creating or Modifying Classifier Control Lists for IP Policy Lists on page 12
- Creating Multiple Forwarding Solutions with IP Policy Lists on page 42
- ip policy-list

Creating Policy Lists for IPv6

The following example creates an IPv6 policy list named routeForIPv6. For information about creating the CLACL used in this example, see the previous sections.

1. Create the policy list routeForIPv6.

   host1(config)#**ipv6 policy-list routeForIPv6**
   host1(config-policy-list)#

2. Create the classification group for the CLACL named ipv6tc67 and assign the precedence to the classification group.

   host1(config-policy-list)#**classifier-group ipv6tc67 precedence 75**
   host1(config-policy-list-classifier-group)#

3. Add a rule to color packets as red, and a second rule that sets the traffic class field of the packets to 7.

   host1(config-policy-list-classifier-group)#**color red**
   host1(config-policy-list-classifier-group)#**mark tcfield 7**

4. Add a rule that specifies a group of forwarding solutions based on classifier control list ipv6tc67.

   host1(config-policy-list-classifier-group)#**forward next-hop 3001:82ab:1020:87ec::/64 order 10**
   host1(config-policy-list-classifier-group)#**forward next-hop 2001:82ab:1020:87ec::/64 virtual-router vr1 ignore-default-route order 20**

5. Exit Policy List Configuration mode to save the configuration.

   host1(config-policy-list-classifier-group)#**exit**
   host1(config-policy-list)#**exit**
   host1(config)#

6. Display the policy list.

   host1#**show policy-list routeForIPv6**

```
                                 Policy Table
                                 ------ -----
IPv6 Policy routeForIPv6
   Administrative state: enable
   Reference count:      0
   Classifier control list: C1, precedence 90
   Classifier control list: ipv6tc67, precedence 75
      forward
         Virtual-router: default
         List:
           next-hop 3001:82ab:1020:87ec::/64, order 10, rule 2 (active)
         Virtual-router: vr1
         List:
           next-hop 2001:82ab:1020:87ec::/64, ignore-default-route, order 20,
```

```
rule 3
     color red
     mark tc-precedence 7
```

You use the **exception http-redirect** command to create an exception rule within a policy classifier group to specify the client application for the destination of packets rather than forwarding them using the forwarding controller (FC).

In lower-numbered releases, the **exception http-redirect** command only supported the creation of exception rules within IPv4 policy lists. You can now configure the **exception http-redirect** command to create exception rules within IPv4 and IPv6 policy lists.

The following example creates an IPv6 policy list, **epIPv6** for the http-redirect exception:

1. Create the policy list epIPv6.

   host1(config)#**ipv6 policy-list epIPv6**
   host1(config-policy-list)#

2. Create the classification group to match all packets.

   host1(config-policy-list)#**classifier-group ***
   host1(config-policy-list-classifier-group)#

3. Create an exception policy for http-redirect.

   host1(config-policy-list-classifier-group)#**exception http-redirect**

4. Exit Policy List Configuration mode to save the configuration.

   host1(config-policy-list-classifier-group)#**exit**
   host1(config-policy-list)#**exit**
   host1(config)#

5. Display the policy list.

   host1#**show policy-list epIPv6**

```
                                  Policy Table
                                  ------ -----
IPv6 Policy epIPv6
   Administrative state: enable
   Reference count:      0
   Classifier control list: *, precedence 100
      exception http-redirect
```

Related Documentation

- Creating or Modifying Classifier Control Lists for IPv6 Policy Lists on page 15
- ipv6 policy-list
- exception http-redirect

Creating Policy Lists for L2TP

The following example creates an L2TP policy list.

1. Create the policy list routeForl2tp.

host1(config)#**l2tp policy-list routeForl2tp**
host1(config-policy-list)#

2. Create the classification group to match all packets.

host1(config-policy-list)#**classifier-group ***
host1(config-policy-list-classifier-group)#

3. Add a rule to color packets as red, and a second rule that uses the rate-limit profile l2tpRLP10.

host1(config-policy-list-classifier-group)#**color red**
host1(config-policy-list-classifier-group)#**rate-limit-profile l2tpRLP10**

4. Exit Policy List Configuration mode to save the configuration.

host1(config-policy-list-classifier-group)#**exit**
host1(config-policy-list)#**exit**
host1(config)#

5. Display the policy list.

host1#**show policy-list routeForl2tp**

```
                                 Policy Table
                                 ------ -----
L2TP Policy routeForl2tp
   Administrative state: enable
   Reference count:      0
   Classifier control list: *, precedence 100
      color red
      rate-limit-profile l2tpRLP20
```

Related Documentation

- Creating or Modifying Classifier Control Lists for L2TP Policy Lists on page 15
- l2tp policy-list

Creating Policy Lists for MPLS

The following example creates an MPLS policy list.

1. Create the policy list routeForMpls.

host1(config)#**mpls policy-list routeForMpls**
host1(config-policy-list)#

2. Create the classification group.

host1(config-policy-list)#**classifier-group * precedence 200**
host1(config-policy-list-classifier-group)#

3. Add one rule that sets the EXP bits for all packets to 2, and a second rule that uses the rate-limit profile mplsRLP

host1(config-policy-list-classifier-group)#**mark-exp 2**
host1(config-policy-list-classifier-group)#**rate-limit-profile mplsRLP5**

4. Exit Policy List Configuration mode to save the configuration.

host1(config-policy-list-classifier-group)#**exit**
host1(config-policy-list)#**exit**

host1(config)#

5. Display the policy list.

```
host1#show policy-list routeForMpls
                                    Policy Table
                                    ------ -----
MPLS Policy routeForMpls
   Administrative state: enable
   Reference count:      0
   Classifier control list: *, precedence 200
      mark-exp 2 mask 7
      rate-limit-profile mplsRLP5
```

NOTE: In JunosE releases numbered lower than 11.3.x, when you attached a QoS profile to an interface configured for MPLS on an ES2 10G LM, the egress traffic from the MPLS interface that was classified as non-best-effort traffic was forwarded using the best-effort queue. In higher-numbered releases, when an output policy is specified with a non-best-effort traffic class and applied on an MPLS interface, the correct traffic class is used to classify and forward outgoing packets from the interface. For example, when an output policy is applied on an MPLS interface configured with a non-best-effort class, the outgoing traffic from the interface is correctly categorized according to the defined traffic class and does not use the best-effort queue.

Related Documentation

- Creating or Modifying Classifier Control Lists for MPLS Policy Lists on page 16
- mpls policy-list

Creating Policy Lists for VLANs

The following example creates a VLAN policy list named routeForVlan. The classifier group lowLatencyLowDrop uses the default precedence of 100.

1. Create the policy list routeForVlan.

   host1(config)#**vlan policy-list routeForVlan**
   host1(config-policy-list)#

2. Create the classification group.

   host1(config-policy-list)#**classifier-group lowLatencyLowDrop**
   host1(config-policy-list-classifier-group)#

3. Create a rule that adds the lowLatencyLowDrop traffic class for all packets that fall into the lowLatencyLowDrop classification.

   host1(config-policy-list-classifier-group)#**traffic-class lowLatencyLowDrop**

4. Add a rule that sets the drop precedence for all packets that fall into the lowLatencyLowDrop classification to green.

   host1(config-policy-list-classifier-group)#**color green**

5. Add a rule that sets the user-priority bits for all packets that fall into the lowLatencyLowDrop classification to 7.

   host1(config-policy-list-classifier-group)#**mark-user-priority 7**

6. Exit to Policy List Configuration mode, then add traffic class rules for packets that conform to different CLACLs.

   host1(config-policy-list-classifier-group)#**exit**
   host1(config-policy-list)#**classifier-group lowLatency**
   host1(config-policy-list-classifier-group)#**traffic-class lowLatency**
   host1(config-policy-list-classifier-group)#**exit**
   host1(config-policy-list)#**classifier-group excellentEffort**
   host1(config-policy-list-classifier-group)#**traffic-class excellentEffort**
   host1(config-policy-list-classifier-group)#**exit**
   host1(config-policy-list)#**classifier-group bestEffort**
   host1(config-policy-list-classifier-group)#**traffic-class bestEffort**

7. Exit Policy List Configuration mode to save the configuration.

   host1(config-policy-list-classifier-group)#**exit**
   host1(config-policy-list)#**exit**
   host1(config)#

8. Display the policy list.

   host1#**show policy-list routeForVlan**

```
                                  Policy Table
                                  ------ -----
VLAN Policy routeForVlan
   Administrative state: enable
   Reference count:      0
   Classifier control list: lowLatencyLowDrop, precedence 100
      traffic-class lowLatencyLowDrop
      color green
      mark-user-priority 7
   Classifier control list: lowLatency, precedence 100
      traffic-class lowLatency
   Classifier control list: excellentEffort, precedence 100
      traffic-class excellentEffort
   Classifier control list: bestEffort, precedence 100
      traffic-class bestEffort
```

Related Documentation

- Creating or Modifying Classifier Control Lists for VLAN Policy Lists on page 16
- vlan policy-list

Configuring Statistics Collection for Output Policies on Tunnel Interfaces

To enable collection and preservation of statistics for output policies on tunnel interfaces based on fragments:

- From Global Configuration mode, enable the capability to collect output policy statistics for tunnel interfaces based on the number of fragments.

  host1(config)#**enable-frag-stats**

To enable collection and preservation of statistics for output policies on tunnel interfaces based on packets:

- From Global Configuration mode, enable the capability to collect output policy statistics for tunnel interfaces based on packets. By default, output policy counters are computed based on the number of packets.

  host1(config)#**no enable-frag-stats**

You cannot configure fragment-based collection of output policy statistics for tunnel interfaces that are managed by the SRC client running on the router. The fragment-based statistics collection functionality is preserved across unified ISSU, stateful SRP switchover, and stateful line module switchover operations.

You can use the **show enable-frag-stats** command to verify whether collection of output policy statistics for traffic on tunnel interfaces is enabled.

Related Documentation

- Statistics Collection for Output Policies on Tunnel Interfaces Overview on page 18
- Verifying Statistics Collection for Output Policies on Tunnel Interfaces on page 231

- enable-frag-stats
- show enable-frag-stats

CHAPTER 4

Creating Classifier Groups and Policy Rules

This chapter provides information for configuring policy-based routing management on E Series routers. See the *E120 and E320 Module Guide* for modules supported on the E120 and E320 routers. The chapter discusses the following topics:

Classifier Groups and Policy Rules Overview

Classifier groups contain the policy rules that make up a policy list. A policy rule is an association between a policy action and an optional CLACL. The CLACL defines the packet flow on which the policy action is taken.

A policy list might contain multiple classifier groups—you can specify the precedence in which classifier groups are evaluated. Classifier groups are evaluated starting with the lowest precedence value. Classifier groups with equal precedence are evaluated in the order of creation.

NOTE: For IP policies, the **forward** command supports the **order** keyword, which enables you to order multiple forward rules within a single classifier group. (See "Using Policy Rules to Provide Routing Solutions" on page 35.)

From Policy Configuration mode, you can assign a precedence value to a CLACL by using the **precedence** keyword when you create a classifier group. The default precedence value is 100. For example:

```
host1(config-policy-list)#classifier-group ipCLACL25 precedence 21
host1(config-policy-list-classifier-group)#
```

The **classifier-group** command puts you in Classifier Group Configuration mode. In this mode you configure the policy rules that make up the policy list. For example:

```
host1(config-policy-list-classifier-group)#forward next-hop 172.18.20.54
```

To stop and start a policy rule without losing statistics, you can suspend the rule. Suspending a rule maintains the policy rule with its current statistics, but the rule no longer affects packets in the forwarding path.

From Classifier Group Configuration mode, you can suspend a rule by using the **suspend** version of that policy rule command. The **no suspend** version reactivates a suspended rule. For example:

```
host1(config-policy-list-classifier-group)#suspend forward next-hop 172.18.20.54
host1(config-policy-list-classifier-group)#no suspend forward next-hop 172.18.20.54
```

You can add, remove, or suspend policy rules while the policy is attached to one or more interfaces. The modified policy takes effect once you exit Policy Configuration mode.

Related Documentation

- Policy Rule Precedence on page 32

Policy Rule Precedence

Because of the flexibility in creating policy lists and classifier groups, you can configure a classifier group that has multiple policy rules.

If a classifier group has multiple rules, the router uses the rules according to their precedence—not in the order in which you created the rules. The first rule listed (the forward rule) for a policy list type has the highest precedence and the last rule has the lowest. The precedence is based on the order in which the router performs rules. Rules are performed in order from lower to higher precedence. In the event of a conflict, a higher precedence rule overrides the lower precedent rule.

The precedence of rules is important if you want a specific rule to be applied. For example, if an IP policy list has both a rate-limit-profile rule (which specifies a color) and a color rule in the same classifier-group, the color specified by the color rule is always used rather than the color implied in the rate-limit-profile rule (the color rule has a higher precedence).

Table 4 on page 33 lists the policy rule commands that you can use for each type of policy list. The table lists the rules in their order of precedence.

NOTE: **The ES2 10G Uplink LM and the ES2 10G LM support only IP, MPLS, and VLAN interfaces.**

Table 4: Policy Rule Commands and Precedence

ATM	Frame Relay	GRE	IP	IPv6	L2TP	MPLS	VLAN
forward	forward	forward	forward	forward	forward	forward	forward
color	color	color	forward interface (input, secondary input, and output policies only)	forward next-hop (for input policies only)	color	color	color
–	–	–	exception for input and secondary input policies only (not supported on ES2 10G Uplink LM)	–	–	–	–
mark-clp (See mark-clp in the *JunosE Command Reference Guide* for platform support information.)	mark-de	mark	forward next-hop (for input policies only)	color	rate-limit-profile	rate-limit-profile	mark-user-priority
filter	filter	filter	color	rate-limit-profile	filter	mark-exp	filter
user-packet-class	user-packet-class	user-packet-class	rate-limit-profile	user-packet-class	user-packet-class	filter	user-packet-class
traffic-class	traffic-class	traffic-class	user-packet-class	traffic-class	traffic-class	user-packet-class	traffic-class
–	–	–	traffic-class	mark	–	traffic-class	–
–	–	–	mark	filter	–	–	–

Table 4: Policy Rule Commands and Precedence *(continued)*

ATM	Frame Relay	GRE	IP	IPv6	L2TP	MPLS	VLAN
–	–	–	filter	–	–	–	–
–	–	–	log (not supported on ES2 10G Uplink LM or ES2 10G LM)	–	–	–	–

NOTE: The commands listed in this section replace the Policy List Configuration mode versions of the commands. For example, the **color** command replaces the Policy List Configuration mode version of the **color** command. The original command may be removed completely in a future release.

Related Documentation

- Classifier Groups and Policy Rules Overview on page 31
- Monitoring Policy Management Overview on page 197
- color
- color-mark-profile
- filter
- forward
- forward interface
- forward next-hop
- green-mark
- log
- mark
- mark-clp
- mark-de
- mark-exp
- mark-user-priority
- next-hop
- next-interface
- rate-limit-profile
- red-mark
- reference-rate

- traffic-class
- user-packet-class
- yellow-mark

Using Policy Rules to Provide Routing Solutions

The next-interface, next-hop, filter, and forward rules provide routing solutions for traffic matching a classifier. A classifier can have only one action that provides a routing solution.

If you configure two routing solution rules, such as filter and forward, in the same classifier group, the router displays a warning message, and the rule configured last replaces the previous rule.

For IP policy lists, policy rules are available to enable you to make a forwarding decision that includes the next interface and next hop:

- Forward next interface—Causes an interface to forward all packets that satisfy the classification associated with that rule to the next interface specified
- Forward next hop—Causes an interface to forward all packets that satisfy the classification associated with that rule to the next-hop address specified

For example, you can route packets arriving at IP interface ATM 0/0.0 so that they area handled as indicated:

- Packets from source 1.1.1.1 are forwarded out of interface ATM 0/0.1.
- Packets from source 2.2.2.2 are forwarded out of interface ATM 2/1.1.
- All other packets are dropped.

To configure this routing policy, issue the following commands:

```
host1(config)#ip classifier-list claclA ip host 1.1.1.1 any
host1(config)#ip classifier-list claclB ip host 2.2.2.2 any
host1(config)#ip policy-list IpPolicy100
host1(config-policy-list)#classifier-group claclA
host1(config-policy-list-classifier-group)#forward interface atm 0/0.1
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group claclB
host1(config-policy-list-classifier-group)#forward interface atm 2/1.1
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group *
host1(config-policy-list-classifier-group)#filter
host1(config-policy-list-classifier-group)#exit
host1(config)#interface atm 0/0.0
host1(config-subif)#ip policy input IpPolicy100 statistics enabled
```

Related Documentation

- Classifier Groups and Policy Rules Overview on page 31
- classifier-group
- ip classifier-list

- ip policy-list

Configuring Policies to Provide Network Security

You can configure policy management to provide a level of network security by using policy rules that selectively forward or filter packet flows:

- Forward—Causes the packet flows that satisfy the classification associated with the rule to be routed by the virtual router
- Filter—Causes the interface to drop all packets of the packet flow that satisfy the classification associated with the rule

To stop a denial-of-service attack, you can use a policy with a filter rule. You need to construct the classifier list associated with the filter rule so that it isolates the attacker's traffic into a flow. To determine the criteria for this classifier list, you need to analyze the traffic received on an interface. "Monitoring Policy Management Overview" on page 197 describes how to capture packets into a log.

For example, you can route packets entering an IP interface (ATM 0/0.0) so that they are handled as indicated:

- Packets from source 1.1.1.1 are routed.
- TCP packets from source 2.2.2.2 with the IP fragmentation offset set to one are dropped.
- All other TCP packets are routed.
- All other packets are dropped.

To configure this policy, issue the following commands:

```
host1(config)#ip classifier-list claclA ip host 1.1.1.1 any
host1(config)#ip classifier-list claclB tcp host 2.2.2.2 any ip-frag-offset eq 1
host1(config)#ip classifier-list claclC tcp any any
host1(config)#ip policy-list IpPolicy100
host1(config-policy-list)#classifier-group claclA
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group claclB
host1(config-policy-list-classifier-group)#filter
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group claclC
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group *
host1(config-policy-list-classifier-group)#filter
host1(config-policy-list-classifier-group)#exit
host1(config)#interface atm 0/0.0
host1(config-subif)#ip policy input IpPolicy100 statistics enabled
```

Related Documentation

- Classifier Groups and Policy Rules Overview on page 31
- classifier-group

- ip classifier-list
- ip policy-list

Creating an Exception Rule within a Policy Classifier Group

To create the exception rule within an IP policy classifier group to specify the client application for the destination of packets rather than forwarding them by the forwarding controller (FC), use the **exception http-redirect** command. Doing this enables the application to then perform an application-dependent action on the content of the packet. The exception rule applies to input and secondary-input policies.

The guidelines for creating exception rules within an IPv6 policy classifier group are the same as those for creating exception rules within an IPv4 policy classifier group.

NOTE: The **exception http-redirect** command is not supported for the ES2 10G Uplink LM.

An exception rule in the input policy only takes effect if neither the input policy nor the secondary policy drops the packet. Packets dropped by input or secondary policies are not exceptioned to the SRP module. HTTP redirect is the only application that is available as a destination of the **exception** rule.

Because classifier groups can contain multiple actions, the following list describes how each rule interacts with the exception rule:

- **color**—Packets are colored and the exception rule is applied.
- **filter**—Packets are filtered and the exception rule is not applied. When the filter rule is present, other rules are not applied.
- **forward**—Forward rule is ignored and the exception rule is applied to packets.
- **log**—Packets are logged and the exception rule is applied.
- **mark**—Packets are marked and the exception rule is applied.
- **next-hop**—Next-hop rule is ignored and the exception rule is applied to packets.
- **next-interface**—Next-interface rule is ignored and the exception rule is applied to packets.
- **rate-limit-profile**—Rate limit is applied and the exception rule is applied to packets.
- **traffic-class**—Traffic class is set and the exception rule is applied to packets.
- **user-packet-class**—User packet class is set and the exception rule is applied to packets.
- **exception**—Exception rule is applied to packets.

Related Documentation

- Classifier Groups and Policy Rules Overview on page 31
- exception http-redirect

Defining Policy Rules for Forwarding

The **forward next-hop** command defines a rule that creates the forwarding solution for packets matching the current CLACL. The **forward** command can be used while the policy list is referenced by interfaces. The **suspend** version suspends the forward rule within the classifier group.

For IPv4 and IPv6 policy lists:

- You can use the **forward interface** command to specify multiple interfaces for IPv4 policies and the **forward next-hop** command to specify next-hop IPv4 or IPv6 addresses as possible forwarding solutions for IPv4 or IPv6 policies. If you define multiple forwarding solutions for a single CLACL, use the **order** keyword to specify the order in which the router chooses the solutions. The router uses the first reachable solution in the list, starting with the solution with the lowest order value. The default order value is 100.

  NOTE: The forward interface and **forward next-hop** commands replace the **nest-interface** and **next-hop** commands.

  The switch route processor (SRP) module Fast Ethernet port cannot be the destination of the **forward next-hop** and **forward next-interface** commands.

- If you specify a next-hop address as the forwarding solution, you can specify that the default route is not used as a routing solution for the next-hop address when selecting a reachable forward rule entry.
- IP interfaces referenced with this command can be tracked if they move. Policies attached to an interface also move if the interface moves. However, statistics are not maintained across the move.
- You can no longer use an interface specifier of **tunnel:mpls** with the **forward interface** command, because that usage requires IP interfaces on top of RSVP-TE tunnels. Such interfaces are no longer present in the redesigned MPLS architecture. However, you can configure a static route for an address that is not otherwise used to point to a tunnel, and then use the **forward next-hop** command in the policy:

```
host1(config)#ip route 10.10.10.10/32 tunnel mpls:foo
host1(config)#ip policy-list bar
host1(config-policy-list-classifier-group)#forward next-hop 10.10.10.10
```

Related Documentation

- Classifier Groups and Policy Rules Overview on page 31
- forward
- forward interface
- forward next-hop

Forwarding Based on Next-Hop Addresses for Input IPv4 and IPv6 Policies

You can define policies for incoming IPv4 and IPv6 traffic and apply the policy lists to the ingress of an interface to enable packet forwarding and routing operations to be performed based on the configured rules and actions. The forward rules that you define in classifier groups contained in a policy list define the forwarding mechanism for IPv4 and IPv6 packets that match the specified classifier access list (CLACL). You can use the **forward interface** command to specify multiple IPv4 interfaces for IPv4 policy lists and the **forward next-hop** command to specify next-hop addresses as possible forwarding solutions for IPv4 and IPv6 policy lists.

The next-hop and next-interface actions override the routing table lookup. In an environment in which Gigabit Ethernet uplink modules are connected to broadcast networks, you can use the next-hop actions for routing and forwarding of traffic. For IPv6 traffic, you cannot configure a forward rule to transmit packets that match a specific CLACL to a specific interface or multiple interfaces. However, you can configure a rule to forward packets that match a CLACL to multiple interfaces for IPv4 traffic.

You can specify multiple next-hop addresses or actions in a single forwarding policy rule. In such a case, packets are forwarded to the first available next-hop address that contains a route in the routing table. You can use the **order** keyword with the **forward next-hop** command in Classifier Group Configuration mode to specify the order of the group of forwarding solutions within a single forward rule.

To enable a forwarding solution to function by overriding the routing table lookup, you can configure policies with one or multiple next-hop addresses. Dynamic selection of the next-hop address is available. If a next-hop with the lowest order becomes reachable or is added freshly to a forward rule, the currently processed element is disregarded and the new next-hop entry is considered. If multiple next-hop addresses specified in the policy list have the same order, the selection is done based on the reachability and the first configured entry. You can specify a maximum of 20 forwarding solutions for a classifier. This limit encompasses the forward next-hop and the next-interface actions.

You can configure multiple next-hop elements in a forward rule for only the same virtual router. You cannot configure multiple forward next-hop rules in a policy that spans across different VRs. If only next-hop elements exist and you do not use the **virtual-router** option with the **forward next-hop** command, then the policy assumes the virtual router context of the CLI, making the policy specific to that VR. The policy can be attached only to interfaces that belong to that VR. You can use the **virtual-router** keyword with the **forward next-hop** command to specify a VR other than the default VR to enable the configuration of next-hop elements for that VR.

When a next-hop address is reachable, only if it has an entry in the routing table, this next-hop can be a default route in certain scenarios. In such cases, you can include the **ignore-default-route** keyword with the **forward next- hop** command to cause the default route to be not considered for the next-hop determination.

If next-hop selection changes dynamically, because of changes in the order of the action or changes in the reachability state of the next-hop, the statistics associated with the next-hop action are preserved, if collection of statistical details is enabled in the policy list. The statistical information is used per classifier rule that has a list of multiple next-hop actions.

Keep the following guidelines in mind while configuring forwarding rules based on next-hop addresses for input IPv6 policies:

- You can configure the rule to forward all packets that match a CLACL to a particular next-hop address only for input IPv6 policies on routers with ES2 4G LMs, ES2 10G LMs, and ES2 10G Uplink LMs (policies applied to ingress interfaces) or IPv6 policies on ES2 4G LMs, ES2 10G LMs, and ES2 10G Uplink LMs that function as access line modules (line modules with policies that receive traffic from low-speed circuits and route it to uplink modules).
- You cannot configure next-hop addresses as forwarding rules for IPv6 policies when the ES2 4G LMs, ES2 10G LMs, and ES2 10G Uplink LMs are core-facing, uplink modules. However, when the ES2 4G LMS, ES2 10G LMs, and ES2 10G Uplink LMs operate as access modules for forwarding rules for IPv6 policies, you can configure the core-facing modules as ES2 4G LMs, ES2 10G LMs, ES2 10G Uplink LMs, or ES2 10G ADV LMs.
- The performance of the policy manager application might be slightly impacted if you configure a significant number of IPv6 policies with forward rules and the reachability states of the configured next-hop addresses transition frequently.
- Forwarding of traffic based on next-hop addresses in input IPv6 policy lists is available only for ingress IPv6 interfaces that are configured over Ethernet or MPLS interfaces.
- You cannot configure forward rules based on next-hop addresses in policy lists for IPv6 interfaces over GRE tunnels.
- You can configure only indirect next-hop addresses while configuring forwarding rules based on next-hop addresses for input IPv6 policies.
- You cannot configure link-local, loopback, or multicast addresses for forwarding of traffic based on next-hop addresses in a classifier group in an IPv6 policy list. If you attempt to configure these types of addresses as next-hop addresses for forwarding of traffic using the **forward next-hop** command for IPv6 policy lists, an error message is displayed.

Related Documentation

- Defining Policy Rules for Forwarding on page 38
- Creating Multiple Forwarding Solutions with IP Policy Lists on page 42
- Creating Policy Lists for IP on page 23
- Creating Policy Lists for IPv6 on page 25

- forward
- forward next-hop

Assigning Values to the ATM CLP Bit

The **mark-clp** command assigns a value of 0 or 1 to the ATM CLP bit for packets conforming to the current classifier control list.

Modules on E Series routers support classifying and marking of the ATM CLP bit according to the following rules:

- Modules on E120 and E320 routers support classifying of the ATM CLP bit only for frame-based interfaces (ATM Adaptation Layer 5 [AAL5] encapsulation), but not for individual ATM cells (ATM Adaptation Layer 0 [AAL0] encapsulation). In this case, if the CLP bit in any cell in the frame has a value of 1, the router treats the reassembled AAL5 frame as if it also had a CLP value of 1.
- Modules on E120 and E320 routers support marking of the ATM CLP bit on frame-based interfaces. In this case, every cell of the segmented frame leaves the router with the same CLP value.
- Modules on ERX7xx models, ERX14xx models, and the ERX310 Broadband Services Router support classifying and marking of the ATM CLP bit for individual ATM cells (AAL0 encapsulation), but not for frame-based interfaces (AAL5 encapsulation).

Related Documentation

- mark-clp

Enabling ATM Cell Mode

When you configure a rate limit profile to account for ATM cell tax, the forwarding code calculates this information to determine the size of a frame instead of using only the frame size.

- Issue the **atm-cell-mode** command to account for the ATM cell tax in statistics and rate calculations:

  host1(config-policy-list)#**atm-cell-mode**

Use the **show rate-limit-profile** command to display the state of the mode.

Related Documentation

- Monitoring Policy Management Overview on page 197
- atm-cell-mode
- show rate-limit-profile

Enabling IP Options Filtering

You can filter packets with IP options on an interface:

- Issue the **ip filter-options** all command.

  host1(config-if)#**ip filter-options all**

When a packet arrives on an interface, the router checks to see if the packet contains IP options. If it does and if IP options filtering is enabled, that packet is dropped. IP options filtering is disabled by default.

Related Documentation

- Classifier Groups and Policy Rules Overview on page 31

- ip filter-options all

Packet Tagging Overview

You can use the traffic-class rule in policies to tag a packet flow so that the QoS application can provide traffic-class queuing. Policies can perform both in-band and out-of-band packet tagging:

- Policies perform in-band tagging by using their respective mark rule to modify a packet header field. For example, IP policies use the **mark** rule to modify an IP packet heard ToS field, and Frame Relay policies use the **mark-de** rule to modify the DE bit.
- Policies perform out-of-band tagging by using the traffic class or color rule. Explicit packet coloring lets you configure prioritized packet flows without having to configure a rate-limit profile. The router uses the color to queue packets for egress queue threshold dropping as described in "Creating Rate-Limit Profiles" on page 79.

For example, an Internet service provider (ISP) provides a Broadband Remote Access Server (B-RAS) service that has both video and data components, and the ISP wants to guarantee that the video traffic gets priority treatment relative to the data traffic. The ISP's users have a 1.5 Mbps virtual circuit (VC) terminating on a digital subscriber line access multiplexer (DSLAM). The ISP wants to allocate 800 Kbps of this link for video, if there is a video stream.

The ISP creates a classifier list to define a video packet flow, creates a policy to color the packets, and applies the policy to the interface:

```
host1(config)#ip classifier-list video ip any any dsfield 16
host1(config)#ip classifier-list data ip any any dsfield 32
host1(config)#ip policy-list colorVideoGreen
host1(config-policy-list)#classifier-group video
host1(config-policy-list-classifier-group)#color green
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group data
host1(config-policy-list-classifier-group)#color yellow
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#exit
host1(config)#interface atm 12/1.1
host1(config-if)#ip policy input colorVideoGreen statistics enabled
```

Related Documentation

- classifier-group
- color
- ip classifier-list
- ip policy-list

Creating Multiple Forwarding Solutions with IP Policy Lists

By default, the router uses a single route table lookup to determine the forwarding solution for packets. For IP policy lists only, the **forward** command enables you to configure one or more unique forwarding solutions (interfaces or next-hop addresses) that override the route table lookup. By creating a group of forwarding solutions, you can ensure that there is a reachable solution for the packets.

You can use the **order** keyword to specify the order of the group of forwarding solutions within a single forward rule. If no order value is specified, then the default order of 100 is assigned to a solution. The router evaluates the forwarding solutions in the group, starting at the solution with the lowest order value, and then uses the first reachable solution. To be considered a reachable solution, a solution must be a reachable interface or a next-hop address that has a route in the routing table. If no solutions are reachable, the traffic is dropped.

The following guidelines apply when you create a group of forwarding solutions in an IP policy list:

- You can specify a maximum of 20 forwarding solutions for a classifier.
- The interface and next-hop elements of a forwarding solution must exist within a single virtual router:
  - Next-interface elements are associated with the virtual router where that interface exists.
  - You can include an optional parameter to specify the virtual router when you define next-hop elements.
  - If only next-hop elements exist and you do not use the virtual router option, then the policy assumes the virtual router context of the command-line interface (CLI), making the policy specific to that VR. The policy can be attached only to interfaces that belong to that VR. However, the policy can still be displayed and modified from any VR. The output of the **show configuration** command displays the policy in the section of output related to that VR rather than in the section for the default VR. This behavior ensures that when you use that output for a configuration script, the policy is specific to the correct VR, and the original configuration is re-created.
- If you specify both an interface element and a next-hop address element, then they both must be reachable to be used. Also, the interface must be the correct interface for the next-hop address.
- If you specify a next-hop address, then you can optionally specify that the default route be ignored.
- If you delete the target (interface or next-hop address) referenced in a rule, that solution is replaced by the null interface but retains the same order number in the policy list. The null interface is always considered unreachable.
- When a forwarding solution with a lower order value than the currently active solution becomes reachable, the router switches to the lower-ordered solution.
- If two rules that have the same order value are reachable, then the rule that was created first is used.

NOTE: The **forward interface** and **forward next-hop** commands are replacing the **next-interface** and **next-hop** commands, which do not support multiple forwarding solutions in a single forward rule.

In the following sample classifier group of a policy list, the forwarding solution of ATM interface 0/0.1 has the lowest order value in the group, and would therefore be selected as the solution for the policy list. However, if this interface is not reachable, the router then attempts to use the solution with the next higher order; which would be ATM interface 12/0.1. If none of the solutions in the group is reachable, the traffic is dropped.

```
host1(config-policy-list)#classifier-group westfordClacl precedence 200
host1(config-policy-list-classifier-group)#forward interface atm 0/0.1 order 10
host1(config-policy-list-classifier-group)#forward interface atm 12/0.1 order 50
host1(config-policy-list-classifier-group)#forward interface atm 3/0.25 order 300
```

NOTE: You can use the **suspend** version of the command to suspend an individual entry in a group of forwarding solutions. The forward rule remains active as long as there is a reachable or active entry in the group of forwarding solutions. If you suspend all entries in the group, the status of the forward rule is changed to suspended.

Related Documentation

- Creating or Modifying Classifier Control Lists for IP Policy Lists on page 12
- Creating Policy Lists for IP on page 23

Creating a Classifier Group for a Policy List

To create a classifier group for a policy list and assigns precedence to the specific CLACL that is referenced in the group:

1. Create a classifier group.

   ```
   host1(config-policy-list)#classifier-group C1 parent-group IPG1
   ```

2. Assign a precedence to the CLACL.

   ```
   host1(config-policy-list)#classifier-group westfordClacl precedence 150
   ```

3. Create a hierarchical policy parameter list.

   ```
   host1(config)#policy-parameter A hierarchical
   host1(config)#parent-group EPG1
   host1(config-parent-group)#exit
   host1(config)#ip policy-list POL
   host1(config-policy-list)#classifier-group C1 external parent-group EPG1 parameter
     A
   host1(config-policy-list)#exit
   ```

   The **no** version removes the classifier group and its rules from a policy list. The **precedence** keyword specifies the order in which a classifier group is evaluated compared to other classifier groups. Classifier groups are evaluated from lowest to highest precedence value (for example, a classifier group with a precedence of 1 is used before a classifier group with a precedence of 2). Classifier groups with equal precedence are evaluated in the order of creation, with the group created first having precedence. A default value of 100 is used if no precedence is specified.

The **parent-group** keyword creates a parent group in a rate-limit hierarchy for IP, IPv6, L2TP, and MPLS. The **external parent-group** keyword creates an external parent group in a rate-limit hierarchy for IP, IPv6, L2TP, and MPLS. All packets matching the classifier are sent to the parent group for further processing, except for packets dropped by the classifier using the filter rule.

More than one classifier group can have the same parent group, which enables you to create hierarchies.

NOTE: Empty classifier groups have no effect on the router's classification of packets and are ignored by the router. You might inadvertently create empty classifier groups in a policy if you use both the newer CLI style and the older CLI style, which used the Policy List Configuration mode version of the classifier list commands.

Related Documentation

- Classifier Groups and Policy Rules Overview on page 31
- Creating Rate-Limit Profiles on page 79
- Monitoring Policy Management Overview on page 197

- aggregation-node
- classifier-group
- ip policy-parameter hierarchical
- ip policy-parameter reference-rate
- ipv6 policy-parameter hierarchical
- ipv6 policy-parameter reference-rate
- l2tp policy-parameter hierarchical
- l2tp policy-parameter reference-rate
- mpls policy-parameter hierarchical
- mpls policy-parameter reference-rate
- next-parent
- parent-group
- policy-parameter hierarchical

Applying Policy Lists to Interfaces and Profiles Overview

You can assign a policy list to supported interfaces and profiles. Policy lists are supported on Frame Relay, IP, IPv6, GRE tunnel, MPLS layer 2, and VLAN interfaces. You can also specify IP, IPv6, and L2TP policies in profiles to assign a policy list to an interface. In either case, you can enable or disable the recording of statistics for bytes and packets affected by the assigned policy.

You can also preserve statistics when you attach a new policy that has a classifier list that is the same for both the original and the new policy attachments.

You can use policy commands to assign an ATM, Frame Relay, GRE tunnel, IP, IPv6, MPLS, or VLAN policy list to an interface. Also, you can use them to specify an IP, IPv6, or L2TP policy list to a profile, which then assigns the policy to the interfaces to which the profile is attached

NOTE:

- The **mpls policy** command is used to attach policies to MPLS Layer 2 circuits only.
- The SRP module Fast Ethernet port does not support policy attachments, nor can the module be the destination for the forward next-hop, forward next-interface, next-hop, and next-interface commands

NOTE: Some of the VLAN subinterfaces on a line module that are in the dormant state are deleted even before the maximum number of VLAN subinterfaces supported on the line module is reached. Such a deletion of VLAN subinterfaces in the dormant state enables input and output policy attachments to the other VLAN subinterfaces that are in the active state to occur successfully. For example, a number of subscribers might be disconnected from VLAN subinterfaces and after the maximum number of supported VLAN subinterfaces is exceeded on a line module, a certain number of clients might be logged in again. In such cases, the deletion of some of the dormant VLAN subinterfaces enables successful attachment of input and output policies to the VLAN subinterfaces for the subscribers that newly logged in.

The Ethernet application on the interface controller starts a timer for 8 milliseconds and deletes the dormant VLAN subinterfaces within this period. The number of dormant Ethernet VLAN subinterfaces that are deleted varies depending on the processor load of the line module.

Use the input or output keyword to assign the policy list to the ingress or egress of the interface. For ATM, IP, and IPv6 policy lists, use the secondary-input keyword to assign the policy list, after route lookup, to data destined for local or remote destinations. For IP and IPv6 policy lists, use the secondary-input keyword to assign the policy list, after route lookup, to data destined to local or remote destinations. The router supports secondary input policies whose principal applications are:

- To defeat denial-of-service attacks directed at a router's local IP or IPv6 stack
- To protect a router from being overwhelmed by legitimate local traffic
- To apply policies on packets associated with the route class

NOTE: The local-input keyword for the ip policy and ipv6 policy commands is deprecated, and may be completely removed in a future release. We recommend you remove the keyword from scripts. Re-create any local input policies using the ip classifier-list local true command and attaching the policies using the ip policy secondary-input command.

You can enable or disable the recording of routing statistics for bytes and packets affected by the policy. If you enable statistics, you can enable or disable baselining of the statistics. The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline whenever baseline-relative statistics are retrieved. You must also enable baselining on the interface with the appropriate baseline command.

NOTE: The gre-tunnel policy command does not support the baseline keyword.

You can use the preserve keyword to save the existing statistics when you attach a policy to an interface that already has a policy attached. This keyword saves the statistics for any classifier-list that is the same for both the new and old policy attachments. Without the preserve keyword, all statistics are deleted when you attach the new policy.

For example, when you replace a policy attachment that references the original policy-list plOne with a new attachment referencing policy-list plTwo, the existing statistics for the classifier group referencing clOne and the default classifier group are saved.

Original Policy Attachment	New Policy Attachment	Comment
ip policy-list plOne	ip policy-list plTwo	-
ip classifier-list clOne	ip classifier-list clOne	statistics from plOne are saved
Forward	Forward	-
ip classifier-list clTwo	ip classifier-list clFour	-
Forward	Forward	-
ip classifier-list clThree	ip classifier-list clFive	-

Original Policy Attachment	New Policy Attachment	Comment
Forward	Forward	-
classifier-list *	classifier-list *	statistics from plOne are saved
Filter	Filter	-

You can use the merge keyword to enable merging of multiple policies to form a single policy.

host1(config)#vlan policy input VlanPolicy33 statistics enabled preserve

host1(config)#ipv6 policy secondary-input my-policy

To assign the policy list named routeForXYZCorp with statistics enabled to the ingress IP interface over an ATM subinterface:

```
host1(config)#interface atm 12/0.1
host1(config)#ip policy input routeForXYZCorp statistics enabled
```

To create an L2TP profile that applies the policy list routeForABCCorp to the egress of an interface:

```
host1(config)#profile bostonProfile
host1(config)#l2tp policy output routeForABCCorp
```

Related Documentation

- atm policy
- frame-relay policy
- gre-tunnel policy
- interface atm
- ip policy
- ipv6 policy
- l2tp policy
- mpls policy
- profile
- vlan policy

Using RADIUS to Create and Apply Policies Overview

E Series routers enable you to use RADIUS to create and apply policies on IPv4 and IPv6 interfaces. This feature supports the Ascend-Data-Filter attribute [242] through a RADIUS vendor-specific attribute (VSA) that specifies a hexadecimal field. The hexadecimal field is encoded with policy attachment, classification, and policy action information

The policy defined in the Ascend-Data-Filter attribute is applied when RADIUS receives a client authorization request and replies with an Access-Accept message.

When you use RADIUS to apply policies, a subset of the router's classification fields and actions is supported. The supported actions and classification fields are:

- Actions
  - Filter
  - Forward
  - Packet marking
  - Rate limit
  - Traffic class
- Classifiers
  - Destination address
  - Destination port
  - Protocol
  - Source address
  - Source port

NOTE: An E Series router dynamically assigns names to the new classifier list and policy list as described in "Ascend-Data-Filter Attribute for IPv4/IPv6 Subscribers in a Dual Stack" on page 52.

To create a policy, you use hexadecimal format to configure the Ascend-Data-Filter attribute on the RADIUS server. For example:

```
Ascend-Data-Filter="01000100 0A020100 00000000 18000000 00000000
00000000"
```

Table 5 on page 49 lists the fields in the order in which they are specified in the hexadecimal Ascend-Data-Filter attribute.

Table 5: Ascend-Data-Filter Fields

Action or Classifier	Format	Comments
Type	1 byte	1=IPv4 3=IPv6
Filter or forward	1 byte	0=filter 1=forward
Indirection	1 byte	0=egress 1=ingress

Table 5: Ascend-Data-Filter Fields *(continued)*

Action or Classifier	Format	Comments
Spare	1 byte	-
Source IP address	4 bytes for IPv4 16 bytes for IPv6	-
Destination IP address	4 bytes for IPv4 16 bytes for IPv6	-
Source IP prefix	1 byte	Type 1 = Number of leading zeros in the wildcard mask Type 3 = Higher-order contiguous bits of the address that comprise the network portion of the address
Destination IP prefix	1 byte	Type 1 = Number of leading zeros in the wildcard mask Type 3 = Higher-order contiguous bits of the address that comprise the network portion of the address
Protocol	1 byte	-
Established	1 byte	Non implemented
Source port	2 bytes	-
Destination port	2 bytes	-
Source port qualifier	1 byte	0= no compare 1= less than 2= equal to 3= greater than 4= not equal to
Destination port qualifier	1 byte	0= no compare 1= less than 2= equal to 3= greater than 4= not equal to
Reserved	2 bytes	-
Marking value	1 byte	Type of Service (ToS)—for IPv4 Differentiated Services Code Point (DSCP)—for IPv6

Table 5: Ascend-Data-Filter Fields *(continued)*

Action or Classifier	Format	Comments
Marking mask	1 byte	0= no packet marking
Traffic class	1–41 bytes	• 0= no traffic class (required if there is no profile) • First byte specifies the length of the ASCII name of the traffic class • Traffic class must be statically configured • Name can optionally be null terminated, which consumes 1 byte • Although the traffic class name field supports up to 41 bytes, you can create an Ascend-Data-Filter attribute with the traffic class name field set to a maximum of 32 bytes only (including null characters). This restriction occurs because the traffic class group configuration enables a traffic class name of up to 31 characters only.
Rate-limit profile	1–41 bytes	• 0= no rate limit (required if there is no profile) • First byte specifies the length of the ASCII, followed by the ASCII name of the profile • Profile must be statically configured • Name can optionally be null terminated, which consumes 1 byte

NOTE: **To create a rate-limit profile, traffic class, or marking rule, you must first configure the filter/forward field as forward.**

A single RADIUS record can contain two policies—one ingress policy and one egress policy. Each policy can have a maximum of 512 ascend-data filters. Each ascend data-filter creates a classifier group and the action associated with the classifier group.

Construction of IPv6 Classifiers from the Hexadecimal Ascend-Data-Filter Attribute

If both the source and destination IP prefixes are 128, the IPv6 classifier is created using the IPv6 host argument as follows:

```
IPv6 classifier-list testipv6 source-host 2001:db8:85a3::8a2e:370:7334 destination-host
  2001:db8::1428:57ab
```

If either the source or destination IP prefix is non-zero, but less than 128 bits, (for example, 64 bits), the IPv6 classifier is created using the IPv6 address argument as follows:

```
IPv6 classifier-list v6cl4 source-address 2001:db8:85a3::8a2e:370:7334/64
  destination-address 2001:db8::1428:57ab/64
```

NOTE: In JunosE Release 10.1.x and earlier, the maximum width of a CAM hardware classifier entry for IPv4 or IPv6 in a single policy was 128 bits. In JunosE Release 10.2.x and later, based on the size limit for a combined IPv6 classifier entry, a maximum of 336 bits of CAM entry is supported for full IPv6 classification with an additional 16 bits for rule set ID. However, OC48/STM16 line modules on ERX14xx models, ERX7xx models, and the ERX310 router support only 128-bit IPv6 classification. For more information on size limits for IP and IPv6 classifiers, see "Size Limit for IP and IPv6 CAM Hardware Classifiers" on page 168.

Ascend-Data-Filter Attribute for IPv4/IPv6 Subscribers in a Dual Stack

The PPP link between the customer premises equipment (CPE) and the provider edge (PE) device or E Series router equipment might require both IPv4 and IPv6 protocols for transmission of data. Such networks require that PE devices run a dual stack of IPv4 and IPv6 services. Dual-stack routers allow simultaneous support for both IPv4 and IPv6 applications. The following guidelines are used to create a policy defined in the Ascend-Data-Filter attribute when IPv4 and IPv6 subscribers are in a network:

- If a subscriber requires only IPv4 services, only the Type 1 action is used in the Access-Accept message returned from the RADIUS server in response to the client authentication request.
- If a subscriber requires only IPv6 services, only the Type 3 action is used in the Access-Accept message returned from the RADIUS server.
- If both IPv4 and IPv6 addresses are assigned to the subscriber interface, then either Type 1 or Type 3 or both the actions are used in the Access-Accept message.
- If the Type 1 action is used and the Indirection action field is set to 01 in the Ascend-Data-Filter attribute, one primary input policy is created and applied on the ingress IPv4 interface.
- If the Type 3 action is used and the Indirection action field is set to 01 in the Ascend-Data-Filter attribute, one primary input policy is created and applied on the ingress IPv6 interface.
- If the Type 1 action is used and the Indirection action field is set to 00 in the Ascend-Data-Filter attribute, one primary output policy is created and applied on the egress IPv4 interface.
- If the Type 3 action is used and the Indirection action field is set to 00 in the Ascend-Data-Filter attribute, one primary output policy is created and applied on the egress IPv6 interface.
- Ascend-Data-Filter attributes for both IPv4 and IPv6 interfaces are stored on the RADIUS server and the appropriate policies are created and applied to the corresponding interfaces when they come up, depending on the type of subscribers.

In lower-numbered releases, the formats of the input and output classifier list names and policy list names were as follows:

- clin_<InterfaceId>_<filterNum>
- clout_<InterfaceId>_<filterNum>
- plin_<InterfaceId>
- plout_<InterfaceId>

where:

- clin—Classifier list included in an input policy list
- clout—Classifier list included in an output policy list
- plin—Policy list applied to the ingress interface
- plout—Policy list applied to the egress interface
- InterfaceId—A unique identifier for the interface to which the policy is applied
- filterNum—A value that denotes the sequence of Ascend-Data-Filter attribute configured on the RADIUS server

In this release, the formats of the input and output classifier list names and policy list names are modified to support IPv6 subscribers. The following is the new format of the input and output classifier list and policy list:

- clin_<AuthId>_<filterNum>
- clout_<AuthId>_<filterNum>
- plin_<ip/ipv6>_<AuthId>
- plout_<ip/ipv6>_<AuthId>

where:

- AuthId—A unique identifier that is used during the authentication of the client with the RADIUS server
- ip/ipv6—Type of protocol used based on the Type action field

Related Documentation

- Examples: Using the Ascend-Data-Filter Attribute for IPv4 Subscribers on page 53
- Examples: Using the Ascend-Data-Filter Attribute for IPv6 Subscribers on page 59

Examples: Using the Ascend-Data-Filter Attribute for IPv4 Subscribers

This section provides examples showing the configuration of policies that use the Ascend-Data-Filter attribute for IPv4 subscribers.

In this example, the following Ascend-Data-Filter attribute creates a RADIUS record that configures an input policy. The policy filters all packets from network 10.2.1.0 with wildcard mask 0.0.0.255 to any destination.

```
Ascend-Data-Filter="01000100 0A020100 00000000 18000000 00000000
00000000"
```


Table 6 on page 54 lists the values specified in the Ascend-Data-Filter attribute.

Table 6: Ascend-Data-Filter Attribute for an Input Policy on an IPv4 Interface

Action or Classifier	Hex Value	Actual Value
Type	01	IPv4
Filter or Forward	00	Filter
Indirection	01	Ingress
Spare	00	None
Source IP address	0a020100	10.2.1.0
Destination IP address	00000000	Any
Source IP mask	18	24 (0.0.0.255)
Destination IP mask	00	0 (255,255,255,255)
Protocol	00	None
Established	00	None
Source port	0000	None
Destination port	0000	None
Source port qualifier	00	None
Destination port qualifier	00	None
Reserved	0000	None

Use the **show classifier-list** and **show policy-list** commands to view information about the policy:

```
host1#show classifier-list

                         Classifier Control List Table
                         ---------- ------- ---- -----

IP clin_1800020_00.1 ip 10.2.1.0 0.0.0.255 any

host1#show policy-list

                                  Policy Table
                                  ------ -----
IP Policy plin_ip_1800020
   Administrative state: enable
   Reference count:      1
   Classifier control list: clin_1800020_00, precedence 100
```

```
      filter

   Referenced by interface(s):
      ATM4/0.0  input policy, statistics enabled, virtual-router default

   Referenced by profile(s):
      No profile references
```

In this example, the Ascend-Data-Filter attribute is used to create RADIUS records that configure two policies. The first policy is an input policy that filters all TCP packets that come from a port greater than 9000 on host 10.2.1.1 and that go to any destination. The second policy is an output policy that filters all UDP packets from network 20.1.0.0 to host 10.2.1.1, port 3090.

Ascend-Data-Filter = "01000100 0A020101 00000000 20000600 23280000 03000000"
Ascend-Data-Filter = "01000000 14010000 0A020101 10201100 00000C12 00020000"

Using the **show classifier-list** and **show policy-list** commands produces the following information about the new policies:

host1#**show classifier-list**

```
                         Classifier Control List Table
                         ---------- ------- ---- -----
IP clin_1800021_00.1 tcp 10.2.1.1 gt 9000 any
IP clout_1800021_01.1 udp 20.1.0.0 0.0.255.255 10.2.1.1 eq 3090
```

host1#**show policy-list**

```
                                  Policy Table
                                  ------ -----
IP Policy plin_ip_1800021
   Administrative state: enable
   Reference count:      1
   Classifier control list: clin_1800021_00, precedence 100
      filter

   Referenced by interface(s):
      ATM4/0.0  input policy, statistics enabled, virtual-router default

   Referenced by profile(s):
      No profile references

IP Policy plout_ip_1800021
   Administrative state: enable
   Reference count:      1
   Classifier control list: clout_1800021_01, precedence 100
      filter

   Referenced by interface(s):
      ATM4/0.0  output policy, statistics enabled, virtual-router default

   Referenced by profile(s):
      No profile references
```

This example creates an input policy and an output policy, each with multiple rules. The rules for the two policies are shown in the following list:

- Input policy rules

- Forward all TCP packets from host 10.2.1.1 to destination 20.0.0.0 0.255.255.255
- Filter all TCP packets from host 10.2.1.1 to any destination.
- Forward all packets from host 10.2.1.1 to any destination.
- Filter all other traffic.

The rules for the input policy translate to the following VSAs. The VSAs must be specified in this order:

Ascend-Data-Filter = "01010100 0A020101 14000000 20080600 00000000 00000000"
Ascend-Data-Filter = "01000100 0A020101 00000000 20000600 00000000 00000000"
Ascend-Data-Filter = "01010100 0A020101 00000000 20000000 00000000 00000000"
Ascend-Data-Filter = "01000100 00000000 00000000 00000000 00000000 00000000"

- Output policy rules
  - Forward all TCP packets from 20.0.0.0 0.255.255.255 to host 10.2.1.1.
  - Filter all TCP packets from any source to host 10.2.1.1.
  - Forward all packets from any source to host 10.2.1.1.
  - Filter all other traffic.

The rules for the input policy translate to the following VSAs. The VSAs must be specified in this order:

Ascend-Data-Filter = "01010000 14000000 0A020101 08200600 00000000 00000000"
Ascend-Data-Filter = "01000000 00000000 0A020101 00200600 00000000 00000000"
Ascend-Data-Filter = "01010000 00000000 0A020101 00200000 00000000 00000000"
Ascend-Data-Filter = "01000000 00000000 00000000 00000000 00000000 00000000"

Using the **show classifier-list** and **show policy-list** commands produces the following information about the new policies:

```
host1#show classifier-list

                         Classifier Control List Table
                         ---------- ------- ---- -----
IP clin_1800022_00.1 tcp host 10.2.1.1 20.0.0.0 0.255.255.255
IP clin_1800022_01.1 tcp host 10.2.1.1 any
IP clin_1800022_02.1 ip host 10.2.1.1 any
IP clout_1800022_04.1 tcp 20.0.0.0 0.255.255.255 host 10.2.1.1
IP clout_1800022_05.1 tcp any host 10.2.1.1
IP clout_1800022_06.1 ip any host 10.2.1.1
```

```
host1#show policy-list
```



```
                                  Policy Table
                                  ------ -----
IP Policy plin_ip_1800022
   Administrative state: enable
   Reference count:      1
   Classifier control list: clin_1800022_00, precedence 100
      forward
   Classifier control list: clin_1800022_01, precedence 100
      filter
   Classifier control list: clin_1800022_02, precedence 100
      forward
   Classifier control list: *, precedence 100
      filter

   Referenced by interface(s):
      ATM4/0.0  input policy, statistics enabled, virtual-router default

   Referenced by profile(s):
      No profile references

IP Policy plout_ip_1800022
   Administrative state: enable
   Reference count:      1
   Classifier control list: clout_1800022_04, precedence 100
      forward
   Classifier control list: clout_1800022_05, precedence 100
      filter
   Classifier control list: clout_1800022_06, precedence 100
      forward
   Classifier control list: *, precedence 100
      filter

   Referenced by interface(s):
      ATM4/0.0  output policy, statistics enabled, virtual-router default

   Referenced by profile(s):
      No profile reference
```

In this example, the following Ascend-Data-Filter attribute creates a RADIUS record that configures an input policy on an IPv4 interface. The policy filters TCP packets from host address 10.2.1.2 to any destination. The policy marks the packets with a ToS byte of 5 and a mask of 170. The policy also applies a traffic class named someTcl and a rate-limit profile named someRlp.

Ascend-Data-Filter="01010100 0a020102 00000000 20000600 045708ae 02010000 05aa0773 6f6d6554 636c0773 6f6d6552 6c70"

Table 7 on page 57 lists the values specified in the Ascend-Data-Filter attribute.

Table 7: Ascend-Data-Filter Attribute Values for a RADIUS Record

Action or Classifier	Hex Value	Actual Value
Type	01	IPv4
Forward	01	Filter
Indirection	01	Ingress

Table 7: Ascend-Data-Filter Attribute Values for a RADIUS Record *(continued)*

Action or Classifier	Hex Value	Actual Value
Spare	00	None
Source IP address	0a020102	10.2.1.2
Destination IP address	00000000	Any
Source IP mask	20	32 (0.0.0.0)
Destination IP mask	00	0 (255,255,255,255)
Protocol	06	TCP
Established	00	None
Source port	0000	None
Destination port	0000	None
Source port qualifier	00	None
Destination port qualifier	00	None
Reserved	0000	None
Marking value	05	5
Marking mask	aa	170
Traffic class	0773 6f6d6554 636c	someTcl
Rate-limit profile	0773 6f6d6552 6c70	someRlp

```
host1#show classifier-list

                         Classifier Control List Table
                         ---------- ------- ---- -----
IP clin_1800023_00.1 tcp host 10.2.1.2

host1#show policy-list

                                  Policy Table
                                  ------ -----
IP Policy plin_ip_1800023
   Administrative state: enable
   Reference count:      1
   Classifier control list: clin_1800023_00, precedence 100
      mark 5 mask 170
      traffic-class someTcl
      rate-limit-profile someRlp
```

```
Referenced by interface(s):
   ATM11/0.0  input policy, statistics enabled, virtual-router default

Referenced by profile(s):
   No profile references
```

Related Documentation

- Examples: Using the Ascend-Data-Filter Attribute for IPv6 Subscribers on page 59
- Using RADIUS to Create and Apply Policies Overview on page 48

Examples: Using the Ascend-Data-Filter Attribute for IPv6 Subscribers

This section provides examples showing the configuration of policies that use the Ascend-Data-Filter attribute when there are IPv6 subscribers in a network.

In this example, the following two Ascend-Data-Filter attributes are used to create RADIUS records that configure two policies. The first policy is an output policy that filters all UDP packets from network 2001:82ab:1020:87ec::0/64 to host 2001:82ab:1020:87ec:1234:0917:3415:0012, port 3090. The second policy is an input policy that filters all TCP packets that come from a port greater than 9000 on host 2001:82ab:1020:87ec:1234:0917:3415:0012 and that go to any destination.

Ascend-Data-Filter1 = "03000000 300182ab 102087ec 00000000 00000000 200182ab 102087ec 12340917 34150012 40801100 00000C12 00020000"
Ascend-Data-Filter2 = "03000100 200182ab 102087ec 12340917 34150012 00000000 00000000 00000000 00000000 80000600 23280000 03000000"

Table 8 on page 59 lists the values specified in the Ascend-Data-Filter1 attribute that are used to create an output policy.

Table 8: Ascend-Data-Filter Attribute for an Output Policy on an IPv6 Interface

Action or Classifier	Hex Value	Actual Value
Type	03	IPv6
Forward	00	Filter
Indirection	00	Egress
Spare	00	None
Source IPv6 address	300182ab 102087ec 00000000 00000000	3001:82ab:1020:87ec: 0000:0000:0000:0000
Destination IPv6 address	200182ab 102087ec 12340917 34150012	2001:82ab:1020:87ec: 1234:0917:3415:0012
Source IPv6 prefix	40	64
Destination IPv6 prefix	80	128
Protocol	11	UDP

Table 8: Ascend-Data-Filter Attribute for an Output Policy on an IPv6 Interface *(continued)*

Action or Classifier	Hex Value	Actual Value
Established	00	None
Source port	0000	None
Destination port	0C12	3090
Source port qualifier	00	None
Destination port qualifier	02	Equal to
Reserved	0000	None

Table 9 on page 60 lists the values specified in the Ascend-Data-Filter2 attribute that are used to create an input policy.

Table 9: Ascend-Data-Filter Attribute for an Input Policy on an IPv6 Interface

Action or Classifier	Hex Value	Actual Value
Type	03	IPv6
Forward	00	Filter
Indirection	01	Ingress
Spare	00	None
Source IPv6 address	200182ab 102087ec 12340917 34150012	2001:82ab:1020:87ec:1234:0917:3415:0012
Destination IPv6 address	00000000 00000000 00000000 00000000	Any
Source IPv6 prefix	80	128
Destination IPv6 prefix	00	0
Protocol	06	TCP
Established	00	None
Source port	2328	9000
Destination port	0000	None
Source port qualifier	03	Greater than
Destination port qualifier	00	None

Table 9: Ascend-Data-Filter Attribute for an Input Policy on an IPv6 Interface *(continued)*

Action or Classifier	Hex Value	Actual Value
Reserved	0000	None

Use the **show classifier-list** and **show policy-list** commands to view information about the configured input and output policies:

host1#**show classifier-list**

```
                         Classifier Control List Table
                         ---------- ------- ---- -----

IPv6 clout_1800020_00.1 udp source-address 3001:82ab:1020:87ec::/64
destination-host
2001:82ab:1020:87ec:1234:917:3415:12 destination-port eq 3090
IPv6 clin_1800020_01.1 tcp source-host 2001:82ab:1020:87ec:1234:917:3415:12
source-port gt 9000
```

host1#**show policy-list**

```
                                  Policy Table
                                  ------ -----
IPv6 Policy plout_ipv6_1800020
   Administrative state: enable
   Reference count:      1
   Classifier control list: clout_1800020_00, precedence 100
      filter

   Referenced by interface(s):
      GigabitEthernet10/0.2  output policy, statistics enabled, virtual-router
default

   Referenced by profile(s):
      None

   Referenced by merged policies:
      None

IPv6 Policy plin_ipv6_1800020
   Administrative state: enable
   Reference count:      1
   Classifier control list: clin_1800020_01, precedence 100
      filter

   Referenced by interface(s):
      GigabitEthernet10/0.2  input policy, statistics enabled, virtual-router
default

   Referenced by profile(s):
      None

   Referenced by merged policies:
      None
```


Related Documentation

- Examples: Using the Ascend-Data-Filter Attribute for IPv4 Subscribers on page 53
- Using RADIUS to Create and Apply Policies Overview on page 48

CHAPTER 5

Creating Rate-Limit Profiles

This chapter provides information for configuring rate-limit policy management on E Series routers. For information on monitoring rate-limit profiles, see "Monitoring Rate-Limit Profiles" on page 229

This chapter discusses the following topics:

Rate Limits for Interfaces Overview

To configure rate limiting for interfaces, you first create a rate-limit profile, which is a set of bandwidth attributes and associated actions. Your router supports two types of rate-limit profiles—one-rate and two-rate—for IP, IPv6, LT2P, and MPLS Layer 2 transport traffic. You next create a policy list with a rule that has rate limit as the action and associate a rate-limit profile with this rule.

You configure rate limit profiles from Global Configuration Mode.

NOTE: **Commands that you issue in Rate Limit Profile Configuration mode do not take effect until you exit from that mode.**

When packets enter an interface that has a rate-limit profile applied, the router performs the following:

- Counts the number of bytes (packets) over time
- Categorizes each packet as committed, conformed, or exceeded
- Assigns a transmit, drop, or mark action

NOTE: **Mark actions and mask values are supported only on IP, IPv6, and MPLS rate-limit profiles. They are not supported on hierarchical rate limits, but are replaced by color-mark profiles.**

An additional function of rate limiting is to apply a color code to packets assigned to each category: green for committed, yellow for conformed, and red for exceeded. The system uses the color code internally to indicate drop preference when an outbound interface is congested.

Rate limiters are implemented using a dual token bucket scheme: a token bucket for conformed (yellow) packets and a token bucket for committed (green) packets. One token is synonymous with one byte. The capacity of the buckets is the maximum number of tokens that can be placed in each bucket.

You configure the bucket capacity with the peak burst parameter or the committed burst parameter. The burst parameters are in bytes (not bytes per second), which is the number of tokens in a full bucket. When a packet passes through a rate limiter, its size is compared to the contents of both buckets, the packet is categorized, and the rate-limiter action is taken on the packet.

Peak rate and committed rate determine the fill rate of their respective buckets. If you set the committed rate to 128,000 bps, tokens are added to the committed (green) bucket at a rate of 128,000 bps (16 K bytes per second), regardless of the traffic. If no traffic passes through the rate limiter, the bucket continues to fill until it reaches the committed burst setting.

Traffic passes through the rate limiter causing a draining of tokens. The drain rate is dependent on how large the packets are and how much time elapses between packets. At any given instant the level of tokens in each bucket is a function of the fill rate, size of packets, and elapsed time between packets.

When packets are received on an interface with a rate limiter applied, the level of tokens in each bucket dynamically changes in both of the following ways:

- Tokens are added every 100-ms sample period
- Tokens are removed based on the size and rate of incoming packets

Related Documentation

- Hierarchical Rate Limits Overview on page 65
- One-Rate Rate-Limit Profiles Overview on page 84
- Two-Rate Rate-Limits Overview on page 88

Hierarchical Rate Limits Overview

In another type of rate limiting, rate-limit hierarchies enable lower priority traffic to access unused bandwidth allocated for real-time traffic, such as voice or video, during times when no real-time traffic is flowing. IP subscribers receive multiple services, such as Web, video, and file transfer, that have a maximum bandwidth. A rate-limit hierarchy can apply a common rate limit to several classified flows, enabling them to share bandwidth according to the preferences set in the hierarchical rate limits.

You can also use rate-limit hierarchies in a layer 2 (ATM) access network for DSL where many routing gateways lead into one Broadband Access Server. The Broadband Access Server uses rate-limit hierarchies to allocate shareable bandwidth to each routing gateway, which enables unused bandwidth from one routing gateway to be used by others. The hierarchy in the rate limit represents the hierarchy in the access network.

Rate-limit hierarchies enable you to share unused bandwidth dynamically, taking unused preferred bandwidth. They also enable real-time traffic to use all guaranteed bandwidth at any time without violating the configured limit on the total interface bandwidth. While preferred traffic fluctuates, the interface rate limit adjusts, dropping non-preferred packets to keep the total flow through the interface under a configured maximum rate, because preferred packets cannot be dropped by the shared rate limits, only by their individual rate limits.

Shared rate limits in the hierarchy keep the combined traffic below a configured maximum without dropping preferred packets. Preferred packets always reduce tokens on these rate limits, making their token counts negative, if necessary. Later non-preferred packets are then dropped in greater volume, bringing the total traffic through the shared rate limit below its configured maximum.

Every packet passing through a rate limit hierarchy has an owner, which is the last rate limit that can modify the packet; for example, by changing its color or dropping it. Preferred packets are owned by their individual preferred rate limits, which do not transfer ownership of the packet while the packet traverses the hierarchy. Ownership of non-preferred packets is transferred while they move from one rate-limit to the next in the hierarchy, so shared rate limits can change the packet color or drop them.

Hierarchical Classifier Groups

Rate-limit hierarchies can be intra-interface, where different flows from classifier groups are in one policy attachment on an interface. Each time the policy is attached to another interface the rate-limit hierarchy is replicated, with no rate limits shared between attachments. Hierarchical rate-limits are only applied at forwarding interfaces because they provide the most accurate classification of packets.

You can configure rate-limit hierarchies by defining a hierarchy of policy classifier and parent groups, each with a rate limit. This hierarchy applies to the packet flow on one interface attachment for the policy. Each policy attachment creates its own copy of the rate-limit hierarchy. There are no shared rate limits across interface attachments.

A policy-based rate-limit hierarchy consists of classifier groups with an aggregate node policy object. Aggregate nodes create the interior nodes of a policy-based hierarchy; they are not classifier groups and the only policy rule applicable to them is the rate limit rule. Every classifier group or aggregate node can select another aggregate node as its parent. The policy manager ensures that these choices always result in a hierarchy. Not every classifier group with a parent aggregate node must have a rate limit rule; multiple classifier groups can share a common parent group, which may have a rate limit rule.

A policy imposes a limit of three parent groups that can be traversed from any classifier group. However, the total number of parent groups in one policy can be up to 512, but every packet must pass through no more than three parent groups at any point.

In a hierarchy of rate limits, a rate limit can be color-blind or color-aware; color-blind rate limits run the same algorithm for all packets, regardless of their color. Color-aware rate limits can change the algorithm used, depending on the color of the incoming packet (possibly set in the previous rate limit or an earlier policy, such as a VLAN policy on ingress or an IP policy). The color mark profile action changes the ToS field for the packet, depending on packet type (EXP for MPLS, DSCP or ToS for IPv4), and transmits the packet. If the mark action uses a color-mark profile, the ToS values marked can depend on the color of the packet.

Hierarchical Rate-Limit Profiles

Hierarchical rate-limit profiles are independent from interface types. You can apply the green, yellow, or red mark values to the rate-limit profile for every type of forwarding interface that accepts ToS marking for packets. The same rate limit can be reused for a different interface type. Hierarchical rate limits have two-rate or TCP-friendly rate types.

The value applied to the ToS field is configured in the CLACL group for green, yellow, or red packets but the coloring of the packet as green, yellow, or red depends on the entire rate-limit hierarchy.

- Preferred packets are transmitted unconditionally. Rate limits that process packets transmitted unconditionally always decrement their token count, if necessary, making it negative.
- Red packets cannot be transmitted unconditionally, to avoid cases where an aggregate rate limit is oversubscribed with transmit-unconditional rates.
- Color-aware uses the incoming packet color in its algorithm
- Not promoting packets means that if the packet enters the rate limit as yellow and the rate-limit then determines that it is green, the packet remains yellow. If the rate limit determines it is red, then the packet is colored red.

A rate-limit rule is an instance of a rate-limit profile. The same profile can be used to create many rate-limit rules in the same hierarchy or in different rate-limit hierarchies. The classifier group that defines the flow can use a mark rule with color-mark profile to set the packet ToS field based on the packet color. A rate-limit hierarchy invoked from the classifier group is one way of changing the packet color; the rate-limit hierarchy is invoked before the classifier group runs the mark rule to set the packet ToS.

Hierarchical Rate-Limit Actions

Every packet traversing a rate-limit hierarchy has an owner that is defined by the last rate limit that can apply its actions to the packet; this is a configuration option.

A rate limit in the hierarchy that does not own the packet only decrements its tokens, but cannot perform any of the following actions:

- Transfer ownership of the packet to the next rate limit.
- Retain ownership of the packet but consume tokens from the remaining rate limits in the hierarchy.
- Exit the rate-limit hierarchy, making that rate limit the final one for the packet.

These actions become the same action if the hierarchy has only one rate limit. Combining these actions with the additional choices to transmit or drop packets results in the following possible actions:

- Drop—Drops the packet at that rate limit in the hierarchy. The packet does not change the state of any rate limit further down the hierarchy.
- Transmit final—Sets the packet color and ends the packet's traversal of the rate-limit hierarchy at the current rate limit. The packet is forwarded and the rate limits further down the hierarchy are not affected. Because transmit final is based on the result of the rate limit, transmit is not an attribute of the node in the rate-limit hierarchy. Committed packets can exit the hierarchy while conformed and exceeded packets continue to the next rate limit.
- Transmit conditional—Sets the packet color to the result calculated by the rate limit and forwards the packet to the next rate limit for processing, also transferring ownership of the packet to the next rate limit. The next rate limit can then set the packet color

according to the state of its token buckets and apply its actions to the packet. The transmit conditional option is the same as connecting the two rate limits in series.

- Transmit unconditional—Sets the packet color to the result calculated by the rate limit, retains ownership of the packet, and forwards the packet to the next rate limit. Later rate limits only decrement their current token counts by the packet length but do not otherwise affect the packet, either by changing its color or applying their actions to it. Although the packet is not affected, the remaining rate limits change because the token counts are reduced, making them more likely to make other packets conformed or exceeded. Transmit unconditional is not allowed as an exceeded action.

  After the transmit-unconditional completes, the packet traverses to the end of the hierarchy. Because ownership of the packet has been retained, no rate limit further down can apply its actions to it. Some of the later rate limits might already have very low token counts, which must still be decremented when processing a transmit-unconditional packet (if necessary, by making the token count negative). Negative token counts enable the remaining rate limits to restrict the total traffic through them to their peak rate (over a large enough averaging interval, which is a function of rates and burst sizes only). Transmit unconditional packets traversing the rate-limit hierarchy reduce the number of tokens available for other packets.

A rate limit has one of the four preceding actions configured for each possible result: committed, conformed, and exceeded. (Transmit unconditional is not allowed as an exceeded action.) The action taken depends only on the result of that rate limit, its rates, burst sizes, and current token state. In addition, the rate limit assigns a color to the packet, depending on both the result of the rate limit and the packet's incoming color. The final color after a packet has finished traversing a rate-limit hierarchy is a function of all the rate limits that owned the packet.

Policy actions are processed in the following order:

1. log
2. filter
3. traffic class
4. user packet class
5. next hop
6. rate limit
7. color status
8. color action
9. parent group
10. mark

The mark action is the last action that occurs, after parent-group, so that the color-mark profile can mark the packet with the final color from the hierarchy.

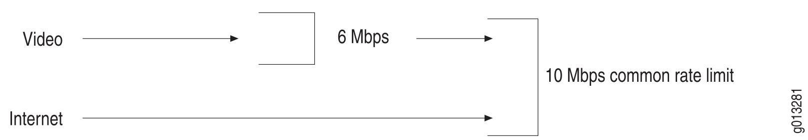
NOTE: To avoid saturation when using dual token buckets, the total amount of yellow transmit unconditional traffic should be less than the peak rate minus the committed rate; the green transmit unconditional traffic should be less than the committed rate.

Example: Multiple Flows Sharing Preferred Bandwidth Rate-Limiting Hierarchical Policy

Figure 2 on page 69 shows an interface with an attached policy that has a Video classifier that singles out a substream of the packets flowing on that interface. The Video classifier can be allocated 6 Mbps out of the 10 Mbps interface rate. All other packets on the interface are Internet. The common rate limit cannot drop Video packets, but must limit the total flow (Video and Internet) to under 10 Mbps. Internet traffic can use the Video bandwidth when there are no active Video calls, while avoiding hard partitioning of interface bandwidth.

NOTE: To avoid rate-limit saturation, we recommend that you set the rate limit profile to color-aware when the rate limit is set to receive transmit conditional.

Figure 2: Multiple Flows Sharing Preferred Bandwidth



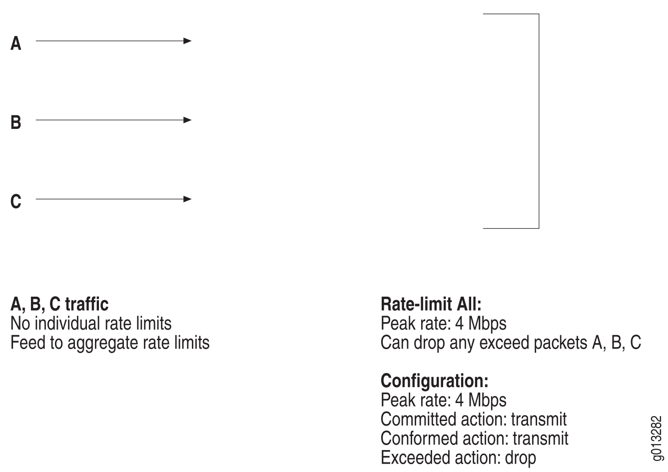

```
host1(config)#rate-limit-profile video two-rate hierarchical
host1(config-rate-limit-profile)#committed-action transmit unconditional
host1(config-rate-limit-profile)#conformed-action transmit unconditional
host1(config-rate-limit-profile)#exceeded-action drop
host1(config-rate-limit-profile)#peak-rate 60000000
host1(config-rate-limit-profile)#exit
host1(config)#rate-limit-profile common two-rate hierarchical
host1(config-rate-limit-profile)# color-aware
host1(config-rate-limit-profile)#committed-action transmit conditional
host1(config-rate-limit-profile)#conformed-action transmit conditional
host1(config-rate-limit-profile)#exceeded-action drop
host1(config-rate-limit-profile)#peak-rate 100000000
host1(config-rate-limit-profile)#exit
host1(config)#policy-list mycompany
host1(config-policy-list)#classifier-group video parent-group all
host1(config-policy-list-classifier-group)#rate-limit-profile video
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group * parent-group all
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#parent-group all
host1(config-policy-list-parent-group)#rate-limit-profile common
host1(config-policy-list-parent-group)#exit
```


In this example, the rate limit Common is color-aware, using the color of the incoming packets instead of setting them to Green. This causes the rate limit Preferred to send 6 Mbps of yellow, transmit unconditional packets. The rate limit Common counts the packets against the yellow token bucket, which has a rate of 10 Mbps. However, if the rate limit Common is color-blind, it treats all packets as Green so the green token bucket gets 6 Mbps of transmit unconditional traffic, which eventually causes all packets to be saturated and dropped.

Example: Multiple Flows Sharing a Rate Limit Hierarchical Policy

Figure 3 on page 70 shows an interface that has one rate limit and three classified flows, A, B, and C. The combined traffic for A, B, and C must be below a peak rate of 4 Mbps, but each individual flow can burst up to that amount. Statistics can be collected separately on A, B, and C, while limiting only the aggregate of all three. None of the flows has any preference in accessing the rate limit, and the rate limit is shared on a first-come first-serve basis.

Figure 3: Multiple Packet Flows Sharing a Rate Limit



This example uses committed and conformed actions for a preferred rate limit profile so that the common rate limit drops only exceeded packets (those packets that raise the traffic load above 4 Mbps); packets below 4 Mbps are transmitted. By specifying **classifier-group * parent-group all**, all packets are sent to the parent group. There is no individual rate limit so that those packet use any available, unused bandwidth in the parent group rate limit.

```
host1(config)#rate-limit-profile All two-rate hierarchical
host1(config-rate-limit-profile)#committed-action transmit conditional
host1(config-rate-limit-profile)#conformed-action transmit conditional
host1(config-rate-limit-profile)#exceeded-action drop
host1(config-rate-limit-profile)#peak-rate 40000000
host1(config-rate-limit-profile)#exit
host1(config)#policy-list rlpshare
host1(config-policy-list)#classifier-group A parent-group All
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group B parent-group All
host1(config-policy-list-classifier-group)#forward
```



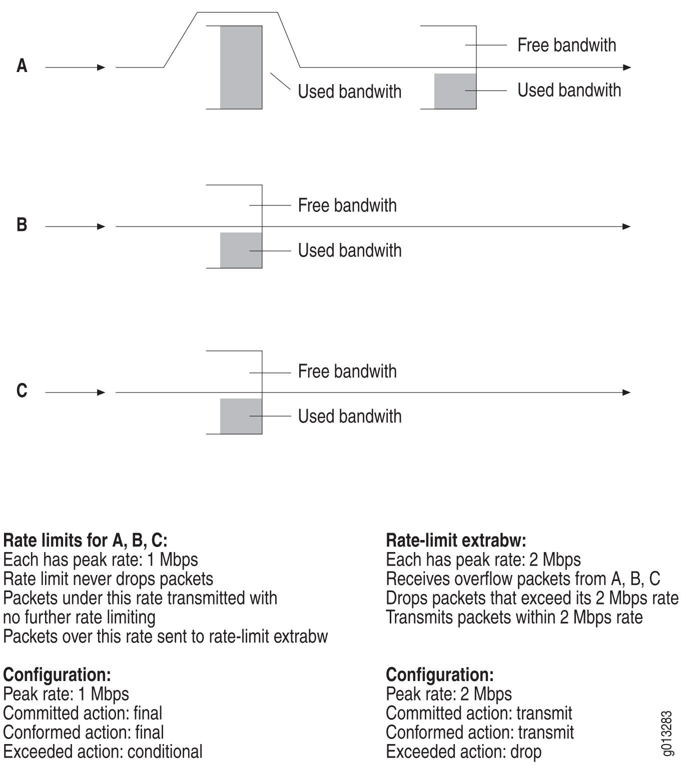
```
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group C parent-group All
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#parent-group All
host1(config-policy-list-parent-group)#rate-limit-profile All
host1(config-policy-list-parent-group)#exit
```

Example: Shared Pool of Additional Bandwidth with Select Flows Rate-Limiting Hierarchical Policy

Figure 4 on page 71 shows three classified flows, A, B, and C, each of which has an individual rate limit with a peak rate of 1 Mbps. If flow A is exceeding its peak rate, rather than drop the packet, the flow tries to use any bandwidth left in a shared rate limit (extrabw) of peak rate of 2 Mbps. The packet is dropped only if both the individual and the shared rate limit have no bandwidth left.

The total flow is limited to 5 Mbps, which is the sum of all the individual peak rates plus the peak rate of the shared rate limit. Individual flows A, B, and C are limited to a maximum of 3 Mbps (1 Mbps from its individual rate limit and up to 2 Mbps if it can consume the entire shared pool); however, it cannot go below a 1 Mbps rate because of the other flows. A shared rate limit enables many flows to share the extra bandwidth dynamically.

Figure 4: Shared Pool of Additional Bandwidth with Select Flows



This example uses **transmit final** so that those packets do not pass through the common rate limit. Transmit final also indicates that there is no shared maximum. If the packets are committed or conformed, they do not need to borrow extra bandwidth or subtract tokens from it. The example uses **exceeded action transmit conditional** so that packets above the individual rate-limit maximum are not dropped but sent to the next rate limit in the hierarchy. Because this is **transmit conditional**, ownership of the packet also transfers so the common rate limit can drop these packets if it has no bandwidth left.

```
host1(config)#rate-limit-profile indiv two-rate hierarchical
host1(config-rate-limit-profile)#committed-action transmit final
host1(config-rate-limit-profile)#conformed-action transmit final
host1(config-rate-limit-profile)#exceeded-action transmit conditional
host1(config-rate-limit-profile)#peak-rate 10000000
host1(config-rate-limit-profile)#exit
host1(config)#rate-limit-profile extrabw two-rate hierarchical
host1(config-rate-limit-profile)#committed-action transmit conditional
host1(config-rate-limit-profile)#conformed-action transmit conditional
host1(config-rate-limit-profile)#exceeded-action drop
host1(config-rate-limit-profile)#peak-rate 20000000
host1(config-rate-limit-profile)#exit
host1(config)#policy-list mypolicy
host1(config-policy-list)#classifier-group A parent-group extrabw
host1(config-policy-list-classifier-group)#rate-limit-profile indiv
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group B parent-group extrabw
host1(config-policy-list-classifier-group)#rate-limit-profile indiv
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group C parent-group extrabw
host1(config-policy-list-classifier-group)#rate-limit-profile indiv
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#parent-group extrabw
host1(config-policy-list-parent-group)#rate-limit-profile extrabw
host1(config-policy-list-parent-group)#exit
```

Example: Aggregate Marking with Oversubscription Rate-Limiting Hierarchical Policy

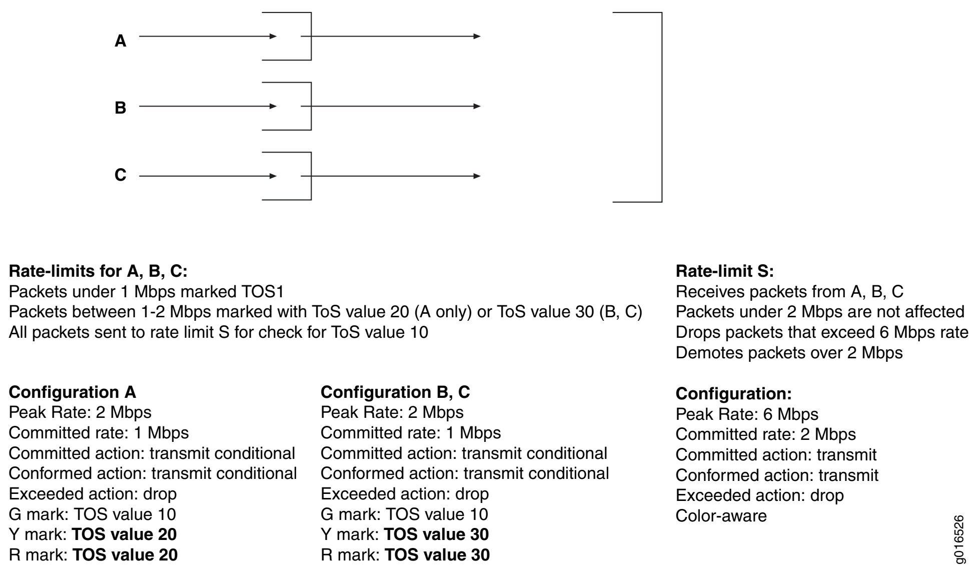
Figure 5 on page 73 shows an aggregate rate limit that enables up to 2 Mbps of traffic to be sent with a ToS value marked as 10. Traffic above that rate is sent with a ToS value marked as 20 or 30 (depending on packet type) and traffic above 6 Mbps is dropped. The 2 Mbps of traffic with the ToS value of 10 is oversubscribed among individual flows A, B, and C, each of which can have up to 1 Mbps of traffic with the ToS value of 10. An individual flow can mark a packet with the ToS value of 10, but if there is insufficient bandwidth at the shared rate limit because of oversubscription, the packet is demoted and remarked.

The demoted packets from flow A are marked with the ToS value of 20 but the demoted packets from flows B and C are marked with the ToS value of 30. The shared rate limit determines whether to demote the packet, in which case each individual rate limit selects the new ToS marking. Individual flows are not required to mark demoted packets with the same value.

The committed and conformed actions are transmit conditional so that all packets also go through rate limit S, because rate limit S imposes the limit of 2 Mbps of traffic with a ToS value of 10 (total across A, B, and C).

Committed packets are transmitted conditionally to rate limit S, which has a peak rate of 6 Mbps and a committed rate of 2 Mbps; these packets can be demoted by S to Y (yellow), in which case they are remarked with the ToS value of 20 or 30. If S leaves them as G (green), they are marked with the ToS value of 10. All conformed packets from A, B, and C are also transmitted conditionally to S but arrive as Y because rate limits do not promote packets in color. S is color-aware so these Y packets do not take away G tokens, leaving them reserved only for the G packets coming from A, B, and C.

Figure 5: Aggregate Marking with Oversubscription




```
host1(config)#rate-limit-profile indiv two-rate hierarchical
host1(config-rate-limit-profile)#committed-action transmit conditional
host1(config-rate-limit-profile)#conformed-action transmit conditional
host1(config-rate-limit-profile)#exceeded-action drop
host1(config-rate-limit-profile)#committed-rate 10000000
host1(config-rate-limit-profile)#peak-rate 20000000
host1(config-rate-limit-profile)#exit
host1(config)#rate-limit-profile S two-rate hierarchical
host1(config-rate-limit-profile)#committed-action transmit conditional
host1(config-rate-limit-profile)#conformed-action transmit conditional
host1(config-rate-limit-profile)#exceeded-action drop
host1(config-rate-limit-profile)#committed-rate 20000000
host1(config-rate-limit-profile)#peak-rate 60000000
host1(config-rate-limit-profile)#color-aware
host1(config-rate-limit-profile)#exit
host1(config)#ip color-mark-profile A
host1(config-color-mark-profile)#green-mark 10
host1(config-color-mark-profile)#yellow-mark 20
host1(config-color-mark-profile)#red-mark 20
host1(config-color-mark-profile)#exit
host1(config)#ip color-mark-profile BC
host1(config-color-mark-profile)# green-mark 10
host1(config-color-mark-profile)# yellow-mark 30
host1(config-color-mark-profile)# red-mark 30
host1(config-color-mark-profile)# exit
host1(config)#policy-list ToS_value_10_oversubsribed
host1(config-policy-list)#classifier-group A parent-group S
host1(config-policy-list-classifier-group)#rate-limit-profile indiv
```

```
host1(config-policy-list-classifier-group)#mark profile A
host1(config-classifier-group)#exit
host1(config-policy-list)#classifier-group B parent-group S
host1(config-policy-list-classifier-group)#rate-limit-profile indiv
host1(config-policy-list-classifier-group)#mark profile BC
host1(config-classifier-group)#exit
host1(config-policy-list)#classifier-group C parent-group S
host1(config-policy-list-classifier-group)#rate-limit-profile indiv
host1(config-policy-list-classifier-group)#mark profile BC
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#parent-group S
host1(config-policy-list-parent-group)#rate-limit-profile S
host1(config-policy-list-parent-group)#exit
```

Color-Aware Configuration for Rate-Limiting Hierarchical Policy

Common to many rate-limit hierarchies is a large aggregate rate limit that receives packets from many smaller individual rate limits. An individual rate limit can mark a packet yellow but, if few individual flows are active, the aggregate rate limit is likely to try to promote it to green, overriding the individual rate limit. For this reason, rate limits never promote packets in color; color-aware rate limits use the incoming color in their algorithm, but the final result is always equal to or less than the initial packet color.

Rate-limit profiles for rate-limit hierarchies include a non-default configuration option for color-aware. For two-rate rate limits this option enables the color-aware algorithm. If hierarchical, TCP-friendly one-rate rate limits have a color-aware algorithm defined.

In the following color-aware example, the non-preferred packets do not take any green tokens from rate-limit A, leaving them all for preferred packets. Preferred packets may take green and also take yellow tokens (which reduces the flow of non-preferred). In this way the non-preferred packets do not reduce the number of green preferred packets, only the number of yellow preferred packets; preferred packets are then marked from a color-mark profile.

```
class non-preferred parent A
color yellow
class preferred parent A
mark profile cm
parent A
rate-limit A !! a color-aware rate limit
```

The color-mark profile translates the packet color, which is independent of its type, to a type-dependent mark for ToS or EXP and applies it to a packet after it has exited the rate-limit hierarchy. If no translation is configured for a color, then packets of that color are not changed.

Transmit-unconditional packets entering a color-aware rate limit uses the color on the packet for the rate-limit algorithm. Doing this ensures that the color-aware rate limit depletes tokens from the token buckets to account for these packets.

Every packet sent through a rate-limit hierarchy is either dropped inside the hierarchy or emerges with a green, yellow, or red color assigned to it by the rate-limit hierarchy. The color depends on the last rate limit in the hierarchy that owned the packet and all prior rate limits. The green, yellow, or red classification applies to packets of any type and is not interface-type dependent.

A packet that has traversed the hierarchy either has been dropped or emerges with a color (green, yellow or red). This final color can be used by a mark rule with a color-mark profile to select the ToS marking for the packet. Because this operation is interface-type dependent, the actual value is configured where the packet entered the hierarchy; however, the color is set by the entire rate-limit hierarchy.

We recommend that all rate-limit profiles that receive transmit unconditional packets should be color-aware. If not color-aware, yellow transmit unconditional packets are processed through both the green and yellow token buckets; if the green rate is low, this causes an oversubscription of transmit unconditional packets and leads to saturation. By making the rate limit color-aware, the yellow transmit unconditional packets are counted only against the yellow token bucket.

Related Documentation

- color
- color-aware
- color-mark-profile
- green-mark
- red-mark
- yellow-mark

Percent-Based Rates for Rate-Limit Profiles Overview

Percent-based rate-limit profiles enable you to divide the reference rate as percentages instead of specific values. You can specify the reference rate on each interface and specify these rates in terms of percentage of this reference rate within the rate-limit profile to derive the appropriate rate. This enables you to define rate-limit profiles with rates in terms of percentage and bursts in terms of milliseconds.

You can use percent-based rate-limit profiles to:

- Configure rates in rate-limit profiles based on a percentage of a parameter. You can assign values to these parameters at the time of attachment, which enables you to use the same policy for multiple interfaces with different parameter values.
- Specify burst sizes in milliseconds when you configure percent-based rate-limits.
- Provide a generic way to configure and use policy parameters. You can use parameter names when you create policy objects and defer assigning values to these parameters until policy attachment. This enables you to share policy objects by attaching the same policy at multiple interfaces with different parameter values. You do not have to specify values each time you attach a policy; if you do not specify interface-specific, the system uses the global value.

Policy Parameter Reference-Rate

You can use a policy parameter reference-rate to derive the rates in rate-limit profiles. You can configure rate-limit profiles as a percentage of this parameter. The system calculates the rate at the time of attachment using the value assigned to this parameter for that interface.

If you do not specify a value for this parameter in Interface Configuration mode, then the Global configuration value is used.

You can modify the value of this parameter in Global Configuration mode or Interface Configuration mode. In Interface Configuration mode, you can change the value using the increase keyword.

If you use the no version of the command in Interface Configuration mode, the parameter value is set to the global default value. The no version of the command with the increase keyword decrements the value. The parameter value cannot have a negative value. The no version of the command in Global Configuration mode deletes the parameter if it is not used anywhere else.

Modified values affect the rates in the rate-limit profiles that are using the reference-rate parameter.

NOTE: Beginning with JunosE Release 10.3.x, you cannot modify the policy reference-rate parameter in Interface Configuration mode, if the percent-based rate-limit profile is used in external parent groups. If you attempt to change the reference-rate parameter that is referenced by multiple classification operations using the **policy-parameter reference-rate** command at the interface level, an error message is displayed in the CLI interface. This restriction exists because multiple interfaces might refer to the same external parent group resource and to prevent such interfaces from losing their reference to the older external parent group resources. However, you can modify the percent-based rate-limit profiles for external parent groups at the hierarchical rate-limit profile level.

Specifying Rates Within Rate-Limit Profiles

Within a rate-limit profile you can specify the rate either as a percentage or a specific value. In two-rate rate-limit profiles, you can select committed rate and peak rate. You can specify one rate in terms of percentage and another as a specific value. Also, one rate can be a percentage of one parameter and another rate can be a percentage of another parameter.

If the rate in a rate-limit profile is x percent, then the actual rate can be calculated from a parameter value as:

Actual rate (in bits per second) = (parameter value *x)/100

The committed rate can be in the range 0—100 percent of the parameter value. The peak rate can be in the range 0—1000 percent of the parameter value.

The parameter value derives the appropriate rate within the rate-limit profile using a percentage. There are no validations to make the total rate less than or equal to the parameter value.

Specifying Burst Sizes

Within a rate-limit profile you can specify the burst size in milliseconds or bytes. Because rate-limit profiles have multiple rates and no restrictions, you can specify one burst in terms of milliseconds and another as bytes whether or not the corresponding rate is a percentage.

If the burst size is m milliseconds, it is calculated as:

Burst size in bytes = (rate in bps * m) / (8*1000)

In this example, the burst size can be in the range 0—10000 ms (10 seconds).

The maximum burst size is 4294967295 bytes (32 bit).

If you do not set the burst size, the system sets the default committed burst and peak burst to 100 ms. If the default burst size is less than 8192, the system changes it to 8192.

Using Service Manager with Merged Policies

When you use the Service Manager, you can attach multiple policies to the same interface point with the **merge** keyword and these policies are then merged into a new policy. The **increase** keyword enables you to change the parameter value for the profile.

If you activate the service without the increase keyword, the interface-specific value of the parameter is set to the value specified in the profile. However, if you activate the service with the **increase** keyword, the interface-specific value of the parameter increases by the value specified in the profile. If there was no interface-specific value at the time of activation of the profile with the **increase** keyword, then it increases from 0.

If you deactivate the service that used the **increase** keyword, the value of the parameter decreases. But if the profile did not use the **increase** keyword, deactivation does not change the current interface-specific value for that parameter. The interface-specific parameter remains until the interface is deleted.

Policy Parameter Configuration Considerations

The following list describes the rules for using policy parameters:

- Policy parameter names must be unique regardless of its type. If you configure a policy parameter with a reference-rate type, then you cannot configure it with another type until it is deleted.
- You can create policy parameters in Global Configuration mode and in Interface Configuration mode in any order.

- In Global Configuration mode, you can assign a parameter type to a parameter name and assign a default value for this parameter.
- If a parameter is configured in Global Configuration mode, but you do not assign a default value, then the system assigns a default value to the parameter. The system default value for any parameter of type reference-rate is 64K (65536).
- In Interface Configuration mode, you assign a parameter type and value for an interface. Policy parameters configured in Interface Configuration mode that have interface-type IP, IPv6, or L2TP specified with the command associate the command with the respective interface in the stack.
- If a parameter is configured in Interface Configuration mode without configuring it in Global Configuration mode, a global configuration is automatically created for this parameter with the type specified in interface configuration and a system-specified default value.
- A parameter value specified in Interface Configuration mode overrides the value specified in Global Configuration mode.
- If the parameter is not configured in Interface Configuration mode, the value from the global configuration is used. If the global value satisfies most of the interfaces, then you do not have to configure parameters for each interface separately, which reduces the number of configuration steps you need to take.
- When you delete an interface, the interface-specific configuration of the parameter is deleted. However, the global configuration remains until you delete it whether it was created explicitly in Global Configuration mode or automatically created in Interface Configuration mode.

  For example, you can configure policy parameter param1 of type reference-rate in Global Configuration mode with a default value of 100,000 and then configure it as 200,000 in Interface Configuration mode for inf1. If you configure a policy parameter as 500,000 in Interface Configuration mode for interface inf1, the system automatically creates parameter param2 with a 64K (65536) global default value. When you delete interface inf1, the system deletes the interface-specific configuration for param1 and param2, but the global configuration values of 100,000 and 64K (65536) remain until you explicitly delete them.
- You must create policy parameters in either Global Configuration mode or Interface Configuration mode before they can be used or referenced as policy objects. For example. before you define a rate in a rate-limit profile in terms of percentage of a policy parameter param1, you must configure param1 as parameter type reference-rate.
- You can configure multiple policy parameters; there are no restrictions on the number of parameters.
- If you modify a policy parameter value in Interface Configuration mode, it affects all policies attached to that interface. If a parameter value is changed for an interface, only the input, secondary-input, and output policies attached to that interface are affected by this change.
- If you modify a policy parameter value in Global Configuration mode, it affects all policies attached to all interfaces that use the global values. For example, if parameter param1 is used in policies attached to two interfaces, but param1 is only configured for

interface i1, when you modify the default value for param1 in Global Configuration mode, it affects only the attachment on the second interface i2.

- You can specify a rate within a rate-limit profile as a percentage of the parameter and burst size in milliseconds. You can use this rate-limit profile in a policy. You can assign values to these parameters for an interface. The actual rate and burst size are calculated at the time of attachment. You can attach the same policy to multiple interfaces with different parameter values.
- A hierarchical rate-limit profile that contains percentage-based parameters can be used in an external parent-group and the global default values can be changed for each external parent group instance. The following restrictions apply:
  - When you attach policies that reference such same external parent group instances to each interface, you must specify the same reference-rate policy-parameter value.
  - You cannot change the reference-rate policy-parameter values at interface level when a policy attachment to the interface exists.

Related Documentation

- Creating Rate-Limit Profiles on page 79
- Policy Parameter Quick Configuration on page 79

Policy Parameter Quick Configuration

To configure policing, use the following steps:

1. Configure a policy parameter in Global Configuration mode.
2. Assign the parameter type and global default value to a parameter.
3. Use this policy parameter in policy objects, create a generic policy, and attach it to multiple interfaces.
4. Adjust the policy parameter value for a specific interface by configuring it in Interface Configuration mode for any interface.

Related Documentation

- Percent-Based Rates for Rate-Limit Profiles Overview on page 75
- Example: Configuring Hierarchical Policy Parameters on page 132

Creating Rate-Limit Profiles

Create rate-limit profiles with a rate based on percentage and a burst in milliseconds. The system creates a policy using these rate-limit profiles and then attaches them to different interfaces using different parameter values.

1. Create policy parameter refRlpRate.

```
host1(config)#policy-parameter refRlpRate reference-rate
host1(config-policy-param-reference-rate)#reference-rate 100000
host1(config-policy-param-reference-rate)#exit
```

2. Create rate-limit profile rlpData.

```
host1(config)#ip rate-limit-profile rlpData
host1(config-rate-limit-profile)#committed-rate refRlpRate percentage 10
host1(config-rate-limit-profile)#committed-burst millisecond 100
host1(config-rate-limit-profile)#peak-rate refRlpRate percentage 100
host1(config-rate-limit-profile)#peak-burst millisecond 150
host1(config-rate-limit-profile)#exit
```

3. Create rate-limit profile rlpVoice.

```
host1(config)#ip rate-limit-profile rlpVoice
host1(config-rate-limit-profile)#committed-rate 64000
host1(config-rate-limit-profile)#committed-burst 100000
host1(config-rate-limit-profile)#peak-rate refRlpRate percentage 100
host1(config-rate-limit-profile)#peak-burst millisecond 150
host1(config-rate-limit-profile)#exit
```

4. Create rate-limit profile rlpVideo.

```
host1(config)#ip rate-limit-profile rlpVideo
host1(config-rate-limit-profile)#committed-rate refRlpRate percentage 70
host1(config-rate-limit-profile)#committed-burst millisecond 100
host1(config-rate-limit-profile)#peak-rate refRlpRate percentage 100
host1(config-rate-limit-profile)#peak-burst millisecond 150
host1(config-rate-limit-profile)#exit
```

5. Create the policy.

```
host1(config)#ip policy-list P
host1(config-policy)#classifier-group data
host1(config-policy-classifier-group)#rate-limit-profile rlpData
host1(config-policy-classifier-group)#exit
host1(config-policy)#classifier-group voice
host1(config-policy-classifier-group)#rate-limit-profile rlpVoice
host1(config-policy-classifier-group)#exit
host1(config-policy)#classifier-group video
host1(config-policy-classifier-group)#rate-limit-profile rlpVideo
host1(config-policy-classifier-group)#exit
host1(config-policy)#exit
```

6. Attach IP Policy P at interface atm5/0.1.

```
host1(config)#interface atm 5/0.1
host1(config-if)#ip policy-parameter reference-rate refRlpRate 1000000
host1(config-if)#ip policy input P
```

7. Attach IP Policy P at interface atm5/0.2 with merge.

```
host1(config)#interface atm 5/0.2
host1(config-if)#ip policy input P stats enabled merge
```

8. Display the policy list.

```
host1#show policy-list

Policy Table
------ -----
IP Policy P
   Administrative state: enable
   Reference count:      1
   Classifier control list: data, precedence 100
```



```
      rate-limit-profile rlpData
   Classifier control list: voice, precedence 100
      rate-limit-profile rlpVoice
   Classifier control list: video, precedence 100
      rate-limit-profile rlpVideo

   Referenced by interfaces:
      ATM5/0.1 input policy, statistics disabled, virtual-router default
      ATM5/0.2 input policy, statistics enabled, virtual-router default

   Referenced by profiles:
      None

   Referenced by merge policies:
      None
```

9. Display the rate-limit profiles.

   host1#**show rate-limit-profile**

```
Rate Limit Profile Table
---- ----- ------- -----
IP Rate-Limit-Profile: rlpData
   Profile Type:                two-rate
   Reference count:             1
   Committed rate:              refRlpRate % 10
   Committed burst:             100 milliseconds
   Peak rate:                   refRlpRate % 100
   Peak burst:                  150 milliseconds
   Mask:                        255
   Committed rate action:       transmit
   Conformed rate action:       transmit
   Exceeded rate action:        drop

IP Rate-Limit-Profile: rlpVoice
   Profile Type:                two-rate
   Reference count:             1
   Committed rate:              64000
   Committed burst:             100000
   Peak rate:                   refRlpRate % 100
   Peak burst:                  150 milliseconds
   Mask:                        255
   Committed rate action:       transmit
   Conformed rate action:       transmit
   Exceeded rate action:        drop

IP Rate-Limit-Profile: rlpVideo
   Profile Type:                two-rate
   Reference count:             1
   Committed rate:              refRlpRate % 70
   Committed burst:             100 milliseconds
   Peak rate:                   refRlpRate % 100
   Peak burst:                  150 milliseconds
   Mask:                        255
   Committed rate action:       transmit
   Conformed rate action:       transmit
   Exceeded rate action:        drop
```

10. Display policy parameters. If a rate-limit profile uses this parameter twice then it increases the reference count by 2.

    host1#**show policy-parameter brief**

```
Reference-rate refRlpRate: 100000, 6 references
```

```
       Display policy parameters
host1#show policy-parameter
Policy Parameter refRlpRate
    Type: reference-rate
    Rate: 100000
    Reference count: 6
    Referenced by interfaces: 1 references
      IP interface ATM5/0.1: 1000000

    Referenced by rate-limit profiles: 5 references
      rlpData
      rlpVoice
      rlpVideo
```

11. Display interface atm5/0.1.

```
host1#show ip interface atm 5/0.1
ATM5/0.1 line protocol Atm1483 is down, ip is down (ready)
  Network Protocols: IP
  Internet address is 1.1.1.1/255.255.255.255
  Broadcast address is 255.255.255.255
  Operational MTU = 0  Administrative MTU = 0
  Operational speed = 100000000 Administrative speed = 0
  Discontinuity Time = 0
  Router advertisement = disabled
  Proxy Arp = disabled
  Network Address Translation is disabled
  TCP MSS Adjustment = disabled
  Administrative debounce-time = disabled
  Operational debounce-time    = disabled
  Access routing = disabled
  Multipath mode = hashed
  Auto Configure = disabled
  Auto Detect = disabled
  Inactivity Timer = disabled

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Policed Packets 0, Bytes 0
  In Error Packets 0
  In Invalid Source Address Packets 0
  In Discarded Packets 0
  Out Forwarded Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Routed Packets 0, Bytes 0
  Out Scheduler Dropped Packets 0, Bytes 0
  Out Policed Packets 0, Bytes 0
  Out Discarded Packets 0

  IP policy input P
    Statistics are disabled
```

12. Display interface atm5/0.2.

```
host1#show ip interface atm 5/0.2
ATM5/0.2 line protocol Atm1483 is down, ip is down (ready)
  Network Protocols: IP
  Internet address is 2.2.2.2/255.255.255.255
  Broadcast address is 255.255.255.255
  Operational MTU = 0  Administrative MTU = 0
  Operational speed = 100000000  Administrative speed = 0
  Discontinuity Time = 0
```

```
Router advertisement = disabled
Proxy Arp = disabled
Network Address Translation is disabled
TCP MSS Adjustment = disabled
Administrative debounce-time = disabled
Operational debounce-time    = disabled
Access routing = disabled
Multipath mode = hashed
Auto Configure = disabled
Auto Detect = disabled
Inactivity Timer = disabled

In Received Packets 0, Bytes 0
  Unicast Packets 0, Bytes 0
  Multicast Packets 0, Bytes 0
In Policed Packets 0, Bytes 0
In Error Packets 0
In Invalid Source Address Packets 0
In Discarded Packets 0
Out Forwarded Packets 0, Bytes 0
  Unicast Packets 0, Bytes 0
  Multicast Routed Packets 0, Bytes 0
Out Scheduler Dropped Packets 0, Bytes 0
Out Policed Packets 0, Bytes 0
Out Discarded Packets 0

IP policy input P
  classifier-group data entry 1
    0 packets, 0 bytes
    rate-limit-profile rlpData
         committed rate: 10000 bps, committed burst: 125 bytes
         peak Rate: 100000 bps, peak burst: 1875 bytes
      committed: 0 packets, 0 bytes, action: transmit
      conformed: 0 packets, 0 bytes, action: transmit
      exceeded:  0 packets, 0 bytes, action: drop
  classifier-group voice entry 1
    0 packets, 0 bytes
    rate-limit-profile rlpVoice
         committed rate: 64000 bps, committed burst: 100000 bytes
         peak Rate: 100000 bps, peak burst: 1875 bytes
      committed: 0 packets, 0 bytes, action: transmit
      conformed: 0 packets, 0 bytes, action: transmit
      exceeded:  0 packets, 0 bytes, action: drop
  classifier-group video entry 1
    0 packets, 0 bytes
    rate-limit-profile rlpVideo
         committed rate: 70000 bps, committed burst: 875 bytes
         peak Rate: 100000 bps, peak burst: 1875 bytes
      committed: 0 packets, 0 bytes, action: transmit
      conformed: 0 packets, 0 bytes, action: transmit
      exceeded:  0 packets, 0 bytes, action: drop
```

To configure a policy-parameter at an interface with the **increase** keyword:

1. Create policy list P2.

   host1(config)#**ip policy-list P2**
   host1(config-policy)#**classifier-group data2**
   host1(config-policy-classifier-group)#**rate-limit-profile rlpData**
   host1(config-policy-classifier-group)#**exit**
   host1(config-policy)#**exit**

2. Attach IP Policy P2 at interface atm5/0.2 with the **merge** keyword.

   host1(config)#**interface atm 5/0.2**
   host1(config-If)#**ip policy-parameter reference-rate refRlpRate 100000**

   This increases from 0.

   host1(config)#**ip policy-parameter reference-rate refRlpRate increase 100000**

   This increases from the existing 100000.

   host1(config)#**ip policy input P2 merge**

3. Verify the configuration.

```
host1#show policy-parameter
Policy Parameter refRlpRate
    Type: reference-rate
    Rate: 100000
    Reference count: 7
    Referenced by interfaces: 2 references
      IP interface ATM5/0.1: 1000000
      IP interface ATM5/0.2: 200000

    Referenced by rate-limit profiles: 5 references
      rlpData
      rlpVoice
      rlpVideo
```

Related Documentation

- Creating a One-Rate Rate-Limit Profile on page 85
- Creating a Two-Rate Rate-Limit Profile on page 91
- Percent-Based Rates for Rate-Limit Profiles Overview on page 75
- rate-limit-profile
- show rate-limit-profile

One-Rate Rate-Limit Profiles Overview

E Series routers implement a single-rate rate limiter, which you can configure to provide more efficient service to TCP applications. With the single-rate rate limiter, when the committed rate is exceeded, the rate limiter drops a single packet and then resumes transmission up to a configurable burst window. The single, unacknowledged packet causes TCP to cut its transmission rate in half rather than falling back to its initial window size.

NOTE: Commands that you issue in Rate Limit Profile Configuration mode do not take effect until you exit from that mode.

The one-rate rate-limit profile attributes are:

- Color aware—Color-aware rate action (only for hierarchical rate limits)
- Committed rate—Target rate for a packet flow

- Committed burst—Amount of bandwidth allocated to accommodate bursty traffic in excess of the rate
- Excess burst—Amount of bandwidth allocated to accommodate a packet in progress when the rate is in excess of the burst
- Committed action—Drop, transmit, mark (IP and IPv6), or mark-exp (MPLS) when traffic flow does not exceed the rate; the mark value is not supported for hierarchical rate limits and the transmit values conditional, unconditional, or final are only supported on hierarchical rate limits
- Conformed action—Drop, transmit, mark (IP and IPv6), or mark-exp (MPLS) when traffic flow exceeds the rate but not the excess burst; the mark value is not supported for hierarchical rate limits and the transmit values conditional, unconditional, or final are only supported on hierarchical rate limits
- Exceeded action—Drop, transmit, mark (IP and IPv6), or mark-exp (MPLS) when traffic flow exceeds the rate; the mark value is not supported for hierarchical rate limits and the transmit values conditional, unconditional, or final are only supported on hierarchical rate limits
- Mask value—Mask to be applied with mark values for the ToS byte; applicable only to IP and IPv6 rate-limit profiles; not supported on hierarchical rate limits
- EXP mask value—Mask to be applied with mark-exp values; applicable only to MPLS rate-limit profiles; not supported on hierarchical rate limits

Related Documentation

- Creating a One-Rate Rate-Limit Profile on page 85
- Creating Rate-Limit Profiles on page 79

Creating a One-Rate Rate-Limit Profile

To create or modify a one-rate rate-limit profile, use the following commands with the **one-rate** keyword:

- **ip** rate-limit-profile command
- **ipv6** rate-limit-profile command
- **l2tp** rate-limit-profile command
- **mpls** rate-limit-profile command

The following example creates a rate-limit profile named tcpFriendly8Mb. This rate-limit profile, when included as part of a rule in a policy list, sets a TCP-friendly rate for a specified flow:

```
host1(config)#ip rate-limit-profile tcpFriendly8Mb one-rate
host1(config-rate-limit-profile)#committed-rate 8000000
host1(config-rate-limit-profile)#committed-burst 1500000
host1(config-rate-limit-profile)#excess-burst 3000000
host1(config-rate-limit-profile)#committed-action transmit
host1(config-rate-limit-profile)#conformed-action transmit
host1(config-rate-limit-profile)#exceeded-action drop
```



```
host1(config-rate-limit-profile)#mask-val 255
```

To configure a single-rate hard limit, set the committed rate and burst rate to the desired values, the committed action to transmit, the conformed action to drop, and the exceeded action to drop. The peak rate must be set to zero.

NOTE: **You can also achieve the characteristics of the single-rate hard limit by configuring a one-rate rate-limit profile with the extended burst rate set to zero.**

Related Documentation

- One-Rate Rate-Limit Profiles Overview on page 84
- Monitoring Policy Management Overview on page 197
- Rate Limits for Interfaces Overview on page 64
- Setting a One-Rate Rate-Limit Profile on page 96

- rate-limit-profile

Configuring a TCP-Friendly One-Rate Rate-Limit Profile

You can configure a committed rate, committed burst, and excess burst for the token bucket. For example, to configure a rate-limit process with hard tail dropping of packets when tokens are unavailable, set the committed rate and committed burst to a nonzero value, and set the excess burst to zero. Setting the excess burst to a nonzero value causes the router to drop packets in a more friendly way.

The configuration values for the preceding attributes determine the degree of friendliness of the rate-limit process. Instead of tail dropping packets that arrive outside the committed and burst rate envelope, the TCP-friendly bucket enables more tokens to be borrowed, up to a limit determined by the excess burst size. The next packet that borrows tokens in excess of the excess burst size is deemed excessive and is dropped if the exceeded action is set to drop.

The rate-limit algorithm is designed to avoid consecutive packet drops in the initial stages of congestion when the packet flow rate exceeds the committed rate of the token bucket. The intention is that just a few packet drops are sufficient for TCP's congestion control algorithm to drastically scale back its sending rate. Eventually, the packet flow rate falls below the committed rate, which enables the token bucket to replenish faster because of the reduced load.

If the packet flow rate exceeds the committed rate for an extended period of time, the rate-limit algorithm tends toward hard tail dropping. In a properly configured scenario, the rate limiter is consistently driven to borrow tokens because of TCP's aggressive nature, but it replenishes the tokens as TCP backs off, resulting in a delivered rate that is very close to the rate configured in the rate-limit profile.

The recommended burst sizes for TCP-friendly behavior are:

- Committed burst—0.2 to 2.0 seconds of the committed rate

- Excess burst—1.0 to 2.0 seconds of the committed rate, plus the committed burst

For example, if the committed rate is 1,000,000 bps, the recommended burst sizes are as follows:

- Committed burst is 1,000,000 x 1.0 x 1/8 = 125,000 bytes

  Multiplying the committed rate by 1.0 seconds converts the rate to bits, then multiplying the number of bits by 1/8 converts the value to bytes.

- Excess burst is 1,000,000 x 1.5 x 1/8 + 125,000 = 312,500 bytes

  Multiplying the committed rate by 1.5 converts the rate to bits, then multiplying the number of bits by 1/8 converts the value to bytes.

TCP-friendly rate limits have only one token bucket, but they also maintain a cumulative debt counter that represents how much traffic above the committed rate has recently been seen. This cumulative debt increases until it reaches the extended burst value; at that point the cumulative debt is reset to 0, but the offending packet is marked red. The cumulative debt increases faster than just by the packet size, so if the TCP source does not respond to TCP flow control and more of its packets are dropped.

Table 10 on page 87 presents equations that can also represent the algorithm for the TCP-friendly one-rate rate limit profile when using hierarchical rate limiting, where:

- B = size of packet in bytes
- CD = cumulative debt
- t = time
- T(t) = number of tokens in token bucket at time t

Table 10: TCP-Friendly One-Rate Rate-Limit Profile Algorithms

Step	Result
If not color aware, use green as the incoming packet color, otherwise use the actual packet color	–
If incoming packet color is green	–
If T(t) > = B	• Packet is colored • T(t) is decremented by B
If T(t) < B and CD is incremented by B – T(t)	–
If CD < Extended Burst and T(t) < B	• Packet is colored • T(t) is decremented by B (allow T(t) < 0, if necessary)
If CD > = Extended Burst B and T(t) < B	• Packet is colored • CD is reset to 0

Table 10: TCP-Friendly One-Rate Rate-Limit Profile Algorithms *(continued)*

Step	Result
If incoming packet color is (only occurs in color-aware operation)	–
If T(t) < B and CD is incremented by B– T(t)	–
If CD < Extended Burst	• Packet is colored yellow • T(t) is decremented by B (allow T(t) < 0, if necessary)
If CD >= Extended Burst	• Packet is colored • CD is reset to 0
If incoming packet color is (only occurs in color-aware operation)	• Packet is colored red

Related Documentation

- One-Rate Rate-Limit Profiles Overview on page 84
- Creating Rate-Limit Profiles on page 79

Two-Rate Rate-Limits Overview

The two-rate rate limiter enables you to build tiered rate-limit services and to specify different treatments for packets at different rates.

Token buckets control how many packets per second are accepted at each of the configured rates and provide flexibility in dealing with the bursty nature of data traffic. At the beginning of each sample period, the two buckets are filled with tokens based on the configured burst sizes and rates. Traffic is metered to measure its volume. When traffic is received, if tokens remain in both buckets, one token is removed from each bucket for every byte of data processed. As long as tokens are still in the committed burst bucket, the traffic is treated as committed.

When the committed burst token bucket is empty but tokens remain in the peak burst bucket, traffic is treated as conformed. When the peak burst token bucket is empty, traffic is treated as exceeded.

In color-blind mode, if the committed token bucket has enough tokens when a packet is received, the packet is green and tokens are subtracted from both the committed and the peak token buckets. If the peak bucket does not have enough tokens left, it is allowed to go negative. Green packets are the committed traffic.

If the committed bucket does not have enough tokens for the packet, the peak bucket is tested (and the committed bucket is not changed). If there are enough tokens in the peak bucket, it is decremented and the packet is yellow. Yellow packets are the conformed traffic. If the peak bucket does not have enough tokens either (because the committed bucket did not have enough tokens), the packet is red. Red packets are the exceeded traffic.

The two-rate rate-limit profile attributes are:

- ATM cell mode—ATM cell tax accounted for in statistics and rate calculations
- Color-aware—Color-aware rate action (only for hierarchical rate limits)
- Committed rate—Target rate for a packet flow
- Committed burst—Amount of bandwidth allocated to accommodate bursty traffic in excess of the committed rate
- Peak rate—Amount of bandwidth allocated to accommodate excess traffic flow over the committed rate
- Peak burst—Amount of bandwidth allocated to accommodate bursty traffic in excess of the peak rate
- Committed action—Drop, transmit, conditional, unconditional, final, mark (IP and IPv6), or mark-exp (MPLS) when traffic flow does not exceed the committed rate; the mark value is not supported for hierarchical rate limits and the transmit values conditional, unconditional, or final are only supported on hierarchical rate limits
- Conformed action—Drop, transmit, mark (IP and IPv6), or mark-exp (MPLS) when traffic flow exceeds the committed rate but remains below the peak rate; the mark value is not supported for hierarchical rate limits and the transmit values conditional, unconditional, or final are only supported on hierarchical rate limits
- Exceeded action—Drop, transmit, mark (IP and IPv6), or mark-exp (MPLS) when traffic flow exceeds the peak rate; the mark value is not supported for hierarchical rate limits and the transmit values conditional, unconditional, or final are only supported on hierarchical rate limits
- Mask value—Mask to be applied with mark values for the ToS byte; applicable only to IP and IPv6 rate-limit profiles; not supported on hierarchical rate limits
- EXP mask value—Mask to be applied with mark-exp values; applicable only to MPLS rate-limit profiles; not supported on hierarchical rate limits

Table 11 on page 89 indicates the interaction between the rate settings and the actual traffic rate to determine the action taken by a rate-limit rule in a policy when applied to a traffic flow. This implementation is known as a two-rate, three-color marking mechanism.

Table 11: Policy Action Applied Based on Rate Settings and Traffic Rate

Peak Rate	Committed Rate = 0	Committed Rate Not 0
Peak rate = 0	• All traffic assigned the exceeded action	• Traffic <= committed rate assigned the committed action • Traffic > committed rate assigned the exceeded action

Table 11: Policy Action Applied Based on Rate Settings and Traffic Rate *(continued)*

Peak Rate	Committed Rate = 0	Committed Rate Not 0
Peak rate not 0	• Traffic <= peak rate assigned the conformed action • Traffic > peak rate assigned the exceeded action	• Traffic <= committed rate assigned the committed action • Committed rate < Traffic < peak rate assigned the conformed action • Traffic > peak rate assigned the exceeded action

Table 12 on page 90 presents equations that can represent the algorithm for the two-rate rate-limit profile, where:

- B = size of packet in bytes
- Tp = size of peak token bucket in bytes (maximum size of this bucket is the configured peak burst)
- Tc = size of the committed token bucket in bytes (maximum size of this bucket is the configured committed burst)
- t = time

Table 12: Two-Rate Rate-Limit Profile Algorithms

Step	Result
If not color-aware, use green as the incoming packet color, otherwise use the actual packet color	–
If incoming packet color is green:	–
If Tc(t) >= B	• Packet is marked as green • Tc(t) is decremented by B • Tp(t) is decremented by B (allow Tp(t) < 0 if necessary)
If Tp(t) >= B and Tc(t) < B	• Packet is marked as yellow • Tp(t) is decremented by B
If Tp(t) < B and Tc(t) < B	• Packet is marked as red
If incoming packet color is (only occurs in color-aware operation)	–
If Tp(t) >= B	• Packet is marked as yellow • Tp(t) is decremented by B
If Tp(t) < B	• Packet is marked as red
If incoming packet color is red (only occurs in color aware operation)	• Packet is marked as red

Related Documentation

- Creating a Two-Rate Rate-Limit Profile on page 91
- Creating Rate-Limit Profiles on page 79

Creating a Two-Rate Rate-Limit Profile

To create or modify a two-rate rate-limit profile, use the following commands with the **two-rate** keyword:

- **ip** rate-limit-profile command
- **ipv6** rate-limit-profile command
- **l2tp** rate-limit-profile command
- **mpls** rate-limit-profile command

The following example creates a rate-limit profile named hardlimit9Mb. This rate-limit profile, when included as part of a rule in a policy list, sets a hard limit on the specified committed rate with no peak rate or peak burst ability:

```
host1(config)#ip rate-limit-profile hardlimit9Mb two-rate
host1(config-rate-limit-profile)#committed-rate 9000000
host1(config-rate-limit-profile)#committed-burst 20000
host1(config-rate-limit-profile)#committed-action transmit
host1(config-rate-limit-profile)#conformed-action drop
host1(config-rate-limit-profile)#exceeded-action drop
host1(config-rate-limit-profile)#mask-val 255
```

The following example modifies the rate-limit profile named hardlimit9Mb to include an exceeded action that marks the packets that exceed the peak rate. This marking action sets the DS field in the ToS byte (the six most significant bits) to the decimal value of 7 using a mask value of 0xFC:

```
host1(config)#ip rate-limit-profile hardlimit9Mb two-rate
host1(config-rate-limit-profile)#exceeded-action mark 7
host1(config-rate-limit-profile)#mask-val 252
```

To set IP precedence in the ToS byte, use the mask value of 0xE0, for visibility into the three most significant bits.

Related Documentation

- Creating Rate-Limit Profiles on page 79
- Monitoring Policy Management Overview on page 197
- Rate Limits for Interfaces Overview on page 64
- Setting a Two-Rate Rate-Limit-Profile on page 97
- Two-Rate Rate-Limits Overview on page 88
- rate-limit-profile

Setting the Committed Action for a Rate-Limit Profile

You can use the **committed-action** command to set the committed action for a rate-limit profile. Packets are colored green. For IP and IPv6 rate-limit profiles, mark the packet by setting the ToS byte (IP) or traffic class field (IPv6) to the specified 8-bit value, and transmit the packet. The mark value is masked with the default 255 unless it is overridden by the **mask-val** command to specify a different mask; not supported on hierarchical rate limits. For MPLS rate-limit profiles, set the EXP bits of MPLS packets to the specified value in the range 0–7, and transmit the packet. The mark EXP value is masked with the default 7 unless you use the **exp-mask** command to specify a different mask; not supported on hierarchical rate limits. The **no** version restores the default value, **transmit**

NOTE: If a rate-limit rule and a mark or mark-exp rule have the same classifier, the marking set by a rate-limit rule takes precedence when packets are processed by that classifier. The rate-limit rule marking overwrites any marking set by the mark rule (to the ToS byte) or by the mark-exp rule (to the EXP bits). These actions do not apply to hierarchical rate limits.

To configure the committed action, enter Rate Limit Profile Configuration mode.

- Issue the **committed-action** command:

  host1(config-rate-limit-profile)#**committed-action transmit**

Related Documentation

- committed-action

Setting the Committed Burst for a Rate-Limit Profile

You can use the **committed-burst** command to set the committed burst in bytes; range is 1–4294967295. You can use the **committed-burst** command to set the committed burst in milliseconds for a rate-limit profile; range is 1–10000. The no version restores the default value, 8192 bytes if the rate is in bytes per second; 100 milliseconds if the rate is in milliseconds.

When you specify a nonzero value for the rate, the burst size is automatically calculated for a 100-ms burst as described for the **committed-rate** command. If the calculated burst size is less than the default value of 8 KB, the default value (8192 bytes) is used.

NOTE: We recommend that you do not configure a committed or peak burst size smaller than the MTU of the interface. Doing so causes large packets to be dropped even when they are transmitted at a very low rate.

When you specify a nonzero value for the committed rate, the committed burst size is calculated based on a 100-ms burst as follows:

committed burst in bytes = (committed rate in bps x 100 ms) ÷ 8 bits per byte

The router displays committed rate in bits per second and committed burst in bytes. For example, if the rate is 8 Mbps, the burst size is 100 ms x 8 Mbps = 800,000 bits or 100,000 bytes:

committed burst = (8,000,000 bps x 100 ms) ÷ 8 = 100,000 bytes

For this example, displaying the rate-limit profile shows:

committed-rate 8000000

committed-burst 100000

If the calculated burst value is less than the default burst size of 8 KB, the default burst size is used. For most configurations this value probably is sufficient, making it optional for you to configure a value for the associated committed burst size.

To configure the committed burst, enter Rate Limit Profile Configuration mode.

- Issue the **committed-burst** command:

  host1(config-rate-limit-profile)#**committed-burst 20000**

Related Documentation

- committed-burst

Setting the Committed Rate for a Rate-Limit Profile

You can set the committed rate as a percentage of a reference rate defined in the specified policy parameter.

- Issue the **committed-rate** command from Rate Limit Profile Configuration mode to set the committed rate in bits per second for a rate-limit profile:

  host1(config-rate-limit-profile)#**committed-rate refRlpRate percentage 10**

Related Documentation

- committed-rate

Setting the Conformed Action for a Rate-Limit Profile

You can use the **conformed-action** command. Packets are colored yellow. For IP and IPv6 rate-limit profiles, mark the packet by setting the ToS byte (IP) or traffic class field (IPv6) to the specified 8-bit value, and transmit the packet. The mark value is masked with the default 255 unless it is overridden by the **mask-val** command to specify a different mask; not supported on hierarchical rate limits. For MPLS rate-limit profiles, set the EXP bits of MPLS packets to the specified value in the range 0–7, and transmit the packet. The mark EXP value is masked with the default 7 unless you use the **exp-mask** command to specify a different mask; not supported on hierarchical rate limits. To set the conformed action for a rate-limit profile:

- Issue the **conformed-action** command from Rate Limit Profile Configuration mode:

  host1(config-rate-limit-profile)#**conformed-action transmit**

Related Documentation

- conformed-action

Setting the Exceeded Action for a Rate-Limit Profile

You can use the **exceeded-action** command to set the exceeded action for a rate-limit profile: Packets are colored red. For IP and IPv6 rate-limit profiles, mark the packet by setting the ToS byte (IP) or traffic class field (IPv6) to the specified 8-bit value, and transmit the packet. The mark value is masked with the default 255 unless it is overridden by the **mask-val** command to specify a different mask; not supported on hierarchical rate limits. For MPLS rate-limit profiles, set the EXP bits of MPLS packets to the specified value in the range 0–7, and transmit the packet. The mark EXP value is masked with the default 7 unless you use the **exp-mask** command to specify a different mask; not supported on hierarchical rate limits. The **no** version restores the default value, **drop.**

- Issue the **exceeded-action** command from Rate Limit Profile Configuration mode:

  host1(config-rate-limit-profile)#**exceeded-action drop**

Related Documentation

- exceeded-action

Setting the Excess Burst for a Rate-Limit Profile

For one-rate rate-limit profiles only, use the **excess-burst** command to set the excess burst in bytes for a rate-limit profile; range is 0–4294967295. Use the **excess-burst** command to set the excess burst in milliseconds for a rate-limit profile; range is 1–10000. The **no** version restores the default value, 0.

- Issue the excess-burst command from Rate Limit Profile Configuration mode:

  host1(config-rate-limit-profile)#**excess-burst millisecond 1000**

Related Documentation

- excess-burst

Setting the Mask Value for MPLS Rate-Limit Profiles

You can use the **exp-mask** command to set the mask value used for MPLS rate-limit profiles, in the range 1–255. The **no** version restores the default value, 7. This command is associated with the **committed-action, conformed-action, and exceeded-action commands.**

- Issue the **exp-mask** command from Rate Limit Profile Configuration mode.

  host1(config-rate-limit-profile)#**exp-mask 5**

Related Documentation

- exp-mask

Setting the Mask Value for IP and IPv6 Rate-Limit Profiles

You can use the **mask-val** command to set the mask value used for IP and IPv6 rate-limit profiles.5.2.0b1 ID-2170 Use the mask values to set the appropriate bits in the ToS field of the IP packet header or in the traffic class field of the IPv6 packet header. The **no** version restores the default value, 255. This command is associated with the **committed-action, conformed-action. and exceeded-action commands.**

- Issue the **mask-val** command from Rate Limit Profile Configuration mode:

  host1(config-rate-limit-profile)#**mask-val 0xFC**

Related Documentation

- mask-val

Setting the Peak Burst for Two-Rate Rate-Limit Profiles

For two-rate rate-limit profiles only, you can use the **peak-burst** command to set the peak burst in bytes for a rate-limit profile; range is 1–4294967295. Use to set the peak burst in milliseconds for a rate-limit profile; range is 1–10000. The no version restore the default value, 100 ms or 8192 bytes (whichever is more).

When you specify a nonzero value for the peak rate, the peak burst size is automatically calculated for a 100-ms burst as described for the **peak-rate** command. If the calculated peak burst size is less than the default value of 8192 bytes, the default value is used.

NOTE: **We recommend that you do not configure a committed or peak burst size smaller than the MTU of the interface. Doing so causes large packets to be dropped even when they are transmitted at a very low rate.**

- Issue the **peak-burst** command in Rate Limit Profile Configuration mode to set the peak burst in bytes:

  host1(config-rate-limit-profile)#**peak-burst 96256**

  To set the peak burst in milliseconds:

  host1(config-rate-limit-profile)#**peak-burst millisecond 1000**

Related Documentation

- peak-burst

Setting the Peak Rate for Rate-Limit Profiles

For two-rate rate-limit profiles only, you can use the **peak-rate** command to set the peak rate in bits per second for a rate-limit profile; range is 1–4294967295. Use to set the peak rate as a percentage value; range is 0–100. During a software upgrade, the peak rate in a rate-limit profile is automatically set to 0 if it was nonzero but less than the committed rate before the upgrade. The **no** version to restores the default value, 0.

When you specify a nonzero value for the peak rate, the peak burst size is calculated based on a 100-ms burst as follows:

peak burst in bytes = (peak rate in bps x 100 ms) ÷ 8 bits per byte

The CLI displays peak rate in bits per second and peak burst in bytes. For example, if the rate is 8 Mbps, the burst size is 100 ms x 8 Mbps = 800,000 bits or 100,000 bytes:

peak burst = (8,000,000 bps x 100 ms) ÷ 8 = 100,000 bytes

For this example, displaying the rate-limit profile shows:

```
peak-rate 8000000
peak-burst 100000
```

If the calculated peak burst value is less than the default peak burst size of 8 KB, the default burst size is used. For most configurations this value is probably sufficient, making it optional to configure the associated peak burst size.

- Issue the **peak-rate** command in Rate Limit Profile Configuration mode to set the peak rate:

  host1(config-rate-limit-profile)#**peak-rate refRlpRate percentage 100**

Related Documentation

- peak-rate

Setting a One-Rate Rate-Limit Profile

You can use the **rate-limit-profile one-rate** command to create a rate-limit profile and enter Rate Limit Profile Configuration mode, from which you can configure attributes for the rate-limit profile. See Table 11 on page 89.

NOTE: The JunosE Software includes the layer 2 headers in the calculations it uses to enforce the rates that you specify in rate-limit profiles.

Use one of the **ip**, **ipv6**, **l2tp**, or **mpls** keywords in front of the command to specify the type of rate-limit profile you want to create or modify. If you do not include one of the keywords, the router creates an IP rate-limit profile by default.

For hierarchical rate limits, do not specify the interface type, but add the **hierarchical** keyword at the end. The **color-aware** keyword is only supported on hierarchical rate limits.

If you do not include a **one-rate** or **two-rate** keyword, the default is a two-rate rate-limit profile. If you enter a **rate-limit-profile** command with the **one-rate** keyword and then type **exit**, the router creates a rate-limit profile with the default values listed in Table 13 on page 97.

Table 13: One-Rate Rate-Limit-Profile Defaults

Policy Attribute	Default Value
type	one-rate
committed-rate	0
committed-burst	8192
excess-burst	0
committed-action	transmit
conformed-action	transmit
exceeded-action	drop
mask (IP and IPv6 rate-limit profiles)	255
exp-mask (MPLS rate-limit profiles)	7

NOTE: We recommend that you do not configure a committed or peak burst size smaller than the MTU of the interface. Doing so causes large packets to be dropped even when they are transmitted at a very low rate.

- Issue the **ip rate-limit-profile** command in Global Configuration mode:

  host1(config)#**ip rate-limit-profile tcpFriendly10Mb one-rate**

  NOTE: Commands that you issue in Rate Limit Profile Configuration mode do not take effect until you exit from that mode.

Related Documentation

- Creating a One-Rate Rate-Limit Profile on page 85
- rate-limit-profile

Setting a Two-Rate Rate-Limit-Profile

You can use the **rate-limit-profile two-rate** command to create a rate-limit profile and enter Rate Limit Profile Configuration mode, from which you can configure attributes for the rate-limit profile. See Table 11 on page 89.

NOTE: The JunosE Software includes the layer 2 headers in the calculations it uses to enforce the rates that you specify in rate-limit profiles

Use one of the **ip**, **ipv6**, **l2tp**, or **mpls** keywords in front of the command to specify the type of rate-limit profile you want to create or modify. If you do not include one of the keywords, the router creates an IP rate-limit profile by default.

For hierarchical rate limits, do not specify the interface type, but add the **hierarchical** keyword at the end. In Parent Group Configuration Mode, associates a rate limit for a parent group. The **color-aware** keyword is only supported on hierarchical rate limits.

If you do not include a **one-rate** or **two-rate** keyword, the default is a two-rate rate-limit profile. If you enter a **rate-limit-profile** command and then type **exit**, the router creates a rate-limit profile with the default values listed in Table 14 on page 98:

Table 14: Two-Rate Rate-Limit-Profile Defaults

Policy Attribute	Default Value
type	two-rate
committed-rate	0
committed-burst	8192
peak-rate	0
peak-burst	8192
committed-action	transmit
conformed-action	transmit
exceeded-action	drop
mask (IP and IPv6 rate-limit profiles)	255
exp-mask (MPLS rate-limit profiles)	7

During a software upgrade, certain values are set as follows:

- Committed burst size—Set to 8192 if it was less than that value before the upgrade
- Peak burst size—Set to 8192 if it was less than that value before the upgrade
- Peak rate—Set to 0 if it was nonzero but less than the committed rate before the upgrade

NOTE: We recommend that you do not configure a committed or peak burst size smaller than the MTU of the interface. Doing so causes large packets to be dropped even when they are transmitted at a very low rate.

- Issue the **ip rate-limit-profile** command in Global Configuration mode:

host1(config)#**ip rate-limit-profile hardlimit9Mb two-rate**

NOTE: Commands that you issue in Rate Limit Profile Configuration mode do not take effect until you exit from that mode.

Related Documentation

- Creating a Two-Rate Rate-Limit Profile on page 91
- exp-mask
- rate-limit-profile

Bandwidth Management Overview

When you configure the rate-limit profile, packets are tagged with a drop preference. The color-coded tag is added automatically when the committed and peak burst values for an interface's rate-limit profile are exceeded. The egress forwarding controller uses the drop preference to determine which packets are dropped when there is contention for outbound queuing resources within the E Series router.

The queuing system uses drop eligibility to select packets for dropping when congestion exists on an egress interface. This method is called dynamic color-based threshold dropping. The 2-bit tag assigns a color code to the packet: red, yellow, or green. Each packet queue has two color-based thresholds as well as a queue limit:

- Red packets are dropped when congestion causes the queue to fill above the red threshold.
- Yellow packets are dropped when the yellow threshold is reached.
- Green packets are dropped when the queue limit is reached.

This internal tagging is done automatically when a rate-limit profile is applied to an interface and does not necessarily reflect the operation of the policy on an interface.

Having a committed rate and a peak rate enables you to configure two different fill rates for the token buckets. For example, you can configure the fill rate on the peak token bucket to be faster than the fill rate on the committed bucket. This configuration enables you to accommodate bursts of traffic, but, through coloring, it enables you to identify which packets are committed and which ones are not.

To enforce ingress data rates below the physical line rate of a port, you can rate limit a classified packet flow at ingress. A rate-limit profile with a policy rate-limit profile rule provides this capability. The rate-limit profile defines the attributes of the desired rate.

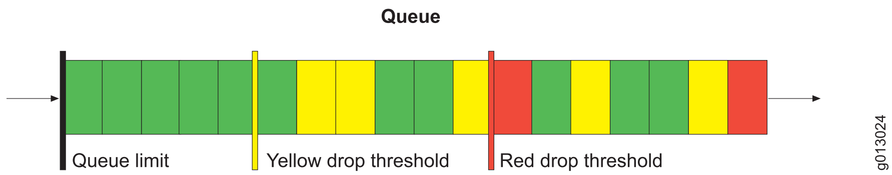
You can set an action based on one rate or two rates. These actions include drop, transmit, or mark. The default is to transmit committed and conformed packets, and to drop exceeded packets.

A color-coded tag is added automatically to each packet based on the following categories:

- Committed—Green
- Conformed—Yellow
- Exceeded—Red

Figure 6 on page 100 illustrates congestion management.

Figure 6: Congestion Management



Examples: One-Rate Rate-Limit Profile

A one-rate rate-limit profile can be configured for hard tail drop rate-limit or TCP-friendly behavior. Packets can be categorized as committed, conformed, or exceeded.

You can configure a one-rate rate-limit profile to hard limit a packet flow to a specified rate. To rate limit the traffic on an interface from source IP address 1.1.1.1 to 1 Mbps, issue the following commands:

```
host1#configure terminal
host1(config)#ip rate-limit-profile oneMegRlp one-rate
host1(config-rate-limit-profile)#committed-rate 1000000
host1(config-rate-limit-profile)#exit
host1(config)#ip classifier-list claclA ip host 1.1.1.1 any
host1(config)#ip policy-list testPolicy
host1(config-policy-list)#classifier-group claclA
host1(config-policy-list-classifier-group)#rate-limit-profile oneMegRlp
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#exit
host1(config)#interface atm 0/0.0
host1(config-subif)#ip policy input testPolicy statistics enabled
```

You can also configure a one-rate rate-limit profile to provide a TCP-friendly rate limiter. To configure a rate limiter with TCP-friendly characteristics, we recommend that you set the committed burst to allow for 1 second of data at the specified rate, and the excess burst to allow 1.5 seconds of data at the specified committed rate plus the committed burst. For example:

```
host1(config)#ip rate-limit-profile tcpFriendly8MB one-rate
host1(config-rate-limit-profile)#committed-rate 8000000
host1(config-rate-limit-profile)#committed-burst 1000000
host1(config-rate-limit-profile)#excess-burst 2500000
host1(config-rate-limit-profile)#committed-action transmit
host1(config-rate-limit-profile)#exceeded-action drop
```

Examples: Two-Rate Rate-Limit Profile

You can configure a two-rate rate-limit profile for two different rates, committed and peak, that are used to define a two-rate, three-color marking mechanism. You can categorize packets as committed, conformed, or exceeded:

- Up to the committed rate, packets are considered to be committed.
- From the committed to peak rate, packets are considered to be conformed.
- After the peak rate, packets are considered to be exceeded.

This configuration is implemented with token buckets. See RFC 2698 for more details.

The following example rate limits traffic on an interface from source IP address 1.1.1.1 so that traffic at a rate up to 1 Mbps is colored green and transmitted, traffic at a rate from 1 Mbps to 2 Mbps is colored yellow and transmitted, and traffic at a rate above 2 Mbps is dropped.

```
host1(config)#ip rate-limit-profile 1MbRLP
host1(config-rate-limit-profile)#committed-rate 1000000
host1(config-rate-limit-profile)#peak-rate 2000000
host1(config-rate-limit-profile)# committed-action transmit
host1(config-rate-limit-profile)#conformed-action transmit
host1(config-rate-limit-profile)#exceeded-action drop
host1(config-rate-limit-profile)#exit
host1(config)#ip classifier-list claclA ip host 1.1.1.1 any
host1(config)#ip policy-list testPolicy
host1(config-policy-list)#classifier-group claclA
host1(config-policy-list-classifier-group)#rate-limit-profile 1MbRLP
host1(config-policy-list-classifier-group)# exit
host1(config-policy-list)#exit
host1(config-policy-list)#interface atm 0/0.0
host1(config-subif)#ip policy input testPolicy statistics enabled
```

Examples: Rate-Limiting Individual or Aggregate Packet Flows

You can construct policies to provide rate limiting for individual packet flows or for the aggregate of multiple packet flows. For example, if you have traffic from multiple sources, you can either rate limit each traffic flow individually, or you can rate limit the aggregate flow for the traffic from all sources.

- To rate limit individual packet flows, use a separate classifier list to classify each flow.
- To rate limit the aggregate of multiple traffic flows, use a single classifier list for the multiple entries.

In the following example, interface ATM 3/1.1 classifies on three traffic flows from different sources. Each traffic flow is rate limited to 1MB (which is defined by the rate-limit profile rl1Meg).

```
host1(config)#ip classifier-list clFlow1 ip host 10.1.1.1 any
host1(config)#ip classifier-list clFlow2 ip host 10.1.1.2 any
host1(config)#ip classifier-list clFlow3 ip host 10.1.1.3 any
host1(config)#ip policy-list plRateLimit
host1(config-policy-list)#classifier-group clFlow1
host1(config-policy-list-classifier-group)#rate-limit-profile rl1Meg
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group clFlow2
host1(config-policy-list-classifier-group)#rate-limit-profile rl1Meg
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group clFlow3
```

```
host1(config-policy-list-classifier-group)#rate-limit-profile rl1Meg
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#exit
host1(config)#interface atm 3/1.1
host1(config-subif)#ip policy input plRateLimit statistics enabled
host1(config-subif)#exit
```

In the following example, interface ATM 3/1.1 again classifies on three traffic flows; however, this policy rate limits the aggregate of the three flows to 1 MB.

```
host1(config)#ip classifier-list clFlowAll ip host 10.1.1.1 any
host1(config)#ip classifier-list clFlowAll ip host 10.1.1.2 any
host1(config)#ip classifier-list clFlowAll ip host 10.1.1.3 any
host1(config)#ip policy-list plRateLimit
host1(config-policy-list)#classifier-group clFlowAll
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#exit
host1(config)#interface atm 3/1.1
host1(config-subif)#ip policy input plRateLimit statistics enabled
host1(config-subif)#exit
```

Related Documentation

- Configuring a TCP-Friendly One-Rate Rate-Limit Profile on page 86
- Packet Tagging Overview on page 42
- Setting the Committed Burst for a Rate-Limit Profile on page 92
- Setting the Committed Rate for a Rate-Limit Profile on page 93
- Setting the Excess Burst for a Rate-Limit Profile on page 94
- Setting the Peak Rate for Rate-Limit Profiles on page 95

Rate-Limiting Traffic Flows

You can rate limit traffic flows destined for an SRP module by implementing a token bucket policer. The configured rate limits are stored in NVS and persist across reboots.

Related Documentation

- Rate Limits for Interfaces Overview on page 64
- Monitoring Policy Management Overview on page 197

- control-plane
- policer

CHAPTER 6

Merging Policies

This chapter provides information about merging policies on E Series routers. The chapter discusses the following topics:

Merging Policies Overview

Merging policies enables you to create multiple policy attachments at an attachment point, resulting in a merged policy that is created and attached at this interface. Executing more than one policy attachment command with the same attachment type at an interface triggers a policy merge through the CLI.

In Profile Configuration mode, policy interface commands for IP and L2TP allow attachments to be merged into any existing merge-capable attachment at an attachment point. Service Manager can request that multiple interface profiles be applied or removed at an interface as part of service activation or deactivation. Service Manager also specifies whether or not the attachments created from these interface profiles are persistent on subsequent reloads.

An interface and an attachment type identify an attachment point. The policies referenced by the component attachments merge into a new policy, which then attaches at the attachment point. The set of component policies are ordered alphabetically by name.

This order determines how any merge conflicts are resolved, with the most recently executed command taking precedence.

With policy merging, a set of policies is combined to form a single new policy, which is a union of all the component policies. Classifier groups and policy rules from each component combine to create the merged policy as in the following example:

```
host1(config)#interface atm 5/0.1
host1(config-subif)#ip policy input p1 statistics enable merge
host1(config-subif)#ip policy input p2 statistics enable merge
host1(config-subif)#ip policy input p3 statistics enable merge
host1(config-subif)#ip policy output p4 statistics enable merge
host1(config-subif)#ip policy output p5 statistics enable merge
host1(config-subif)#exit
```

The example internally results in the following, where policies p1 + p2 + p3 = mpl_10 and policies p4 + p5 = mpl_11.

```
interface atm 5/0.1
ip policy input mpl_10 statistics enable merge
ip policy output mpl_11 statistics enable merge
exit
```

The classifier list referenced by the classifier group is neither split or merged. If a merged policy already exists for a set of component policies, then the merged policy is used for the attachment. An attachment enables a merged policy to have one or more attachments.

The CLI and the Service Manager applications are the only clients of policy management that can request merging of policy attachments. With policy merging, classifier groups and policy rules from each component policy combine into the merged policy.

Policy merging follows these rules:

- The Classifier list referenced by the classifier group cannot be split or merged.
- Policy merging combines classifier groups from all component policies into the merged policy. In the previous example, policies p1, p2, and p3 are the component policies and mpl_10 is the merged policy. The merge policy is created as if all CLI commands for each component policy are run in the context of the merged policy. The merged policy result is the sum of all commands executed in the respective component policies CLI context in a predetermined merge order.
- If a merged policy already exists for a set of component policies, the merged policy is used for the attachment instead of creating a new one. This functionality allows a merged policy to have one or more attachments. A merge policy is automatically deleted when the last reference is removed.

The following restrictions apply to policy merging:

- Classifier lists cannot be merged.
- Secure policies cannot be merged.
- Policies created using ascend-data-filters cannot be merged.

- Existing policy VSAs in RADIUS are not changed; attachments created by this method cannot be merged. Ascend data filter policies can be attached at input and output attachment points.
- SNMP support for polling statistics based on component policy attachments is not available.
- The merge policy naming convention is not configurable.

Related Documentation

- Error Conditions for Merged Policies on page 110
- Merging Policies on page 111
- Merged Policy Naming Conventions on page 108
- Resolving Policy Merge Conflicts on page 105

Resolving Policy Merge Conflicts

The set of component policies are first ordered by their name to form the final merged policy. For example, if the component policies sets contain cp_1, cp_3, cp_9, cp_2, the order in which these policies are merged is cp_1, cp_2, cp_3, and cp_9. The merge order is important for resolving merge conflicts.

Various conflicting combinations of component policies can result in a merged policy that is not a perfect union of the component policies. These conflicts are resolved as they currently are in policy CLI context, where, in any conflict, the most recently executed command takes precedence.

More than one component policy can contain the same classifier group. If the precedence does not match, the precedence of the classifier group defined in the last component policy becomes the final precedence for this classifier group in the merged policy, as in the following example:

```
host1(config)#ip policy-list p1
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#forward
host1(config-classifier-group)#exit
host1(config)#ip policy-list p2
host1(config-policy)#classifier-group C1 precedence 100
host1(config-classifier-group)#forward
host1(config-classifier-group)#exit
host1(config)#ip policy-list p3
host1(config-policy)#classifier-group C1 precedence 130
host1(config-classifier-group)#forward
host1(config-classifier-group)#exit
```

If you combine p1, p2, and p3, you get the following with p1, p2, p3 as the merge order for the set of component policies.

```
ip policy-list mpl_10
classifier-group C1 precedence 130
forward
exit
```

For IP, the forward, filter, next-hop, and next-interface rules are mutually exclusive within a classifier group. For all other types, filter and forward rules are mutually exclusive.

A conflict arises when more than one component policy has the same classifier group and when the rule sets defined in these classifier groups conflict. To resolve the merge conflict, the last command entered replaces any previous conflicting commands for a classifier group, as in the following example:

```
host1(config)#ip policy-list p1
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#forward
host1(config-classifier-group)#exit
host1(config)#ip policy-list p2
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#next-hop 1.1.1.1
host1(config-classifier-group)#exit
host1(config)#ip policy-list p3
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#filter
host1(config-classifier-group)#exit
```

Combining p1 and p2 internally results in:

```
ip policy-list mpl_20
classifier-group C1 precedence 90
next-hop 1.1.1.1
exit
```

Combining p2 and p3 internally results in:

```
ip policy-list mpl_21
classifier-group C1 precedence 90
filter
exit
```

Combining p1, p2, and p3 internally results in:

ip policy-list mpl_22

```
classifier-group C1 precedence 90
filter
exit
```

If you have the same policy rule with different parameters, the parameter of the last rule entered with the same type is used, with the exception of IP forward rule, to resolve the conflict, as in the following example:

```
host1(config)#ip policy-list p1
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#color red
host1(config-classifier-group)#exit
host1(config)#ip policy-list p2
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#color yellow
host1(config-classifier-group)#exit
```

Combining p1 and p2 internally results in:


```
ip policy-list mpl_20
classifier-group C1 precedence 90
color yellow
exit
```

With the IP policy forward rule, when more forward rules are added to an existing classifier group, the list of forward rules is created. This is also true during merging, as in the following example:

```
host1(config)#ip policy-list p1
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#forward next-hop 1.1.1.1
host1(config-classifier-group)#exit
host1(config)#ip policy-list p2
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#forward next-interface atm 5/0.1
host1(config-classifier-group)#exit
host1(config)#ip policy-list p2
host1(config-policy)#classifier-group C1 precedence 90
host1(config-classifier-group)#forward next-interface fastEthernet 4/0.1
  next-hop 1.1.1.2
host1(config-classifier-group)#exit
```

Combining p1, p2, and p3, internally results in the following:

```
ip policy-list mpl_10
classifier-group C1 precedence 90
forward next-hop 1.1.1.1
forward next-interface atm 5/0.1
forward next-interface fastEthernet 4/0.1 next-hop 1.1.1.2
exit
```

Policy management enables multiple policy attachments at the same attachment point, which results in a merged policy that is created and attached at the specified attachment point. The logical OR of the **statistics** and **baseline** keywords of all attachments are used as the **statistics** and **baseline** keyword for the merged policy attachment, as in the following example:

```
host1(config)#interface atm 5/0.1
host1(config-subif)#ip policy input p1 statistics enable baseline enable merge
host1(config-subif)#ip policy input p2 merge
host1(config-subif)#ip policy input p3 statistics enable merge
host1(config-subif)#exit
```

Results in the following:

```
interface atm 5/0.1
ip policy input mpl_5 statistics enable baseline enable merge
exit
```

Related Documentation

- Merging Policies Overview on page 103

Merged Policy Naming Conventions

Merged policies are dynamically created. The naming convention is mpl_hex_of internally_generated_policy ID, such as mpl_10. If the newly generated name already exists, then a sequence number is appended to the new name to make it unique. The sequence number starts at 1 and increments until the name is unique, such as mpl_10_2.

Related Documentation

- Merging Policies Overview on page 103

Reference Counting for Merged Policies

The reference counts in all containers referenced within a merged policy are incremented by the number of times they are referenced within the merged policy. Also, the reference counts of all component policies of a merged policy are incremented because of the association of the component policies with the merged policy. This means you cannot delete a component policy while a merged policy is still associated with it.

Related Documentation

- Merging Policies Overview on page 103

Persistent Configuration Differences for Merged Policies Through Service Manager

Service Manager can specify whether a component policy attachment is nonvolatile. If the interface where the component policy is attached is volatile, then policy management makes the attachment volatile even when the Service Manager specifies otherwise. A nonvolatile interface can have both volatile and nonvolatile component policy attachments. The merged policy that is created is the merge of all component policies attached at a given attachment point regardless of their volatility. The merged policy and its attachments are always volatile and reconstructed on each reload operation.

Related Documentation

- Merging Policies Overview on page 103
- Policy Attachment Sequence at Login Through Service Manager on page 108

Policy Attachment Sequence at Login Through Service Manager

During a user login, you can specify policy attachments through Service Manager, RADIUS, and Interface Profile. The order that is used to select the policy attachment source is Service Manager, RADIUS, and Interface Profile.

For example, if you configure Ingress-Policy-Name VSA for a user in RADIUS and also have a profile with an input policy reference applied to this user's interface column, when the user logs in, the RADIUS VSA is selected as the source for the input policy attachment. If you also have service profiles applied to the user's interface column, the service profiles override both RADIUS VSA and the policy name specified in the interface profile.

NOTE: Policy merging is not supported with ascend data filter policies.

Policy management does not reselect the source if the policy attachment fails for the selected source. If the policy attachment via service profiles fails, policy management does not reselect RADIUS VSA as the next source. This means the interface does not have any input policy attachment.

Related Documentation

- Persistent Configuration Differences for Merged Policies Through Service Manager on page 108
- Policy Attachment Rules for Merged Policies on page 109

Policy Attachment Rules for Merged Policies

The attributes of a policy attachment are as follows:

- Policy name—Name of policy to be attached.
- Attachment type—Type of attachment.
- Statistics enable/disable—Enable or disable statistics for the attachment.
- Baseline enable/disable—Enable or disable baselining for the attachment.
- Merge or Replace—Allow an attachment to become merge-capable and merge with any other attachments that are merge-capable. If the **merge** keyword is not specified, then it replaces any existing attachments with the new attachment. Merging always preserves statistics.
- Preserve—Preserve statistics from earlier attachment when replacing an attachment. This keyword is mutually exclusive with **merge** keyword.

Various possibilities result from a policy attachment at an interface due to the presence or absence of these keywords. The same rules apply while attaching policies based on interface profiles provided by Service Manager except as noted.

Attachments made through Interface Configuration mode follow these rules:

- If an attachment is issued with the **merge** keyword specified:
  - Any existing attachment of the same type at the interface without the **merge** keyword is replaced by the new attachment, which then becomes merge-capable.
  - An attachment is merged with any existing attachments of the same type that have the **merge** keyword set. If a merged policy already exists for the set of component policies, then this merged policy is used or a new merged policy is created dynamically and attached. The statistics for common classifier groups are preserved when replacing the existing merged attachment.
- If an attachment is issued when no **merge** or **preserve** keyword is set, then it replaces all other attachments with the same type at the interface. This attachment is not

merge-capable for future use and statistics from previous attachments are not preserved.

- If an attachment is issued when the **merge** keyword is not set, but the **preserve** keyword is set, it replaces all other attachments with the same type at the interface. This attachment is not merge-capable for future use. Statistics from existing attachments are preserved for all the common classifier-groups.
- You cannot have multiple attachments of the same policy on a single attachment point. Only Service Manager executes multiple attachments of the same policy at the same attachment point.
- A detachment based on the policy name removes all attachments for that policy at the specified attachment point in a single command regardless of creation source. A detachment based on attachment type detaches all attachments at that attachment point regardless of creation source. Service Manager can delete only one attachment at a time through service deactivation.
- The **statistics** and **baseline** keywords for the merged policy attachment are computed as a logical OR for all attachments at the specified attachment point.
- If you delete an attachment:
  - The merged policy is recomputed with the remaining attachments of the same type that have the **merge** keyword set. The statistics for common classifier groups are preserved when replacing the existing merged attachment.
  - The **statistics** and **baseline** keywords for the merged policy attachment are recomputed to be a logical OR of all remaining attachments at the specified attachment point.

Related Documentation

- Policy Attachment Sequence at Login Through Service Manager on page 108

Error Conditions for Merged Policies

Most errors, such as mismatched interface types while merging attachments, are caught during configuration. If merging fails, the attachment at the given interface is not modified.

You can modify component policies manually. Although you might want to do this for debugging purposes, we highly discourage you doing this because it can affect synchronization with the Service Manager application. You cannot manually attach a final merged policy to any interfaces. Instead, attach the set of component policies that constitute this merged policy. If you want to modify the final merged policy, use existing policy merging or component policy modification to achieve this.

Related Documentation

- Resolving Policy Merge Conflicts on page 105

Merging Policies

In the following example IP policy p1 and IP policy p2 are attached at interface atm5/0.1 as input attachments. Subsequently, policy p3 is attached at the same point. Then policies p1 and p2 are attached as output at atm 5/0.2.

1. Create IP policy p1.

   host1(config)#**ip classifier-list C1 tcp host 1.1.1.1 any eq 80**
   host1(config)#**ip classifier-list C2 icmp any any 8 0**
   host1(config)#**ip policy-list p1**
   host1(config-policy)#**classifier-group C1 precedence 90**
   host1(config-policy-classifier-group)#**forward next-hop 10.1.1.1**
   host1(config-policy-classifier-group)#**exit**
   host1(config-policy)#**classifier-group C2 precedence 10**
   host1(config-policy-classifier-group)#**filter**
   host1(config-policy-classifier-group)#**exit**

2. Create IP policy p2.

   host1(config)#**ip classifier-list C1 tcp host 1.1.1.1 any eq 80**
   host1(config)#**ip classifier-list C3 ip any host 2.2.2.2**
   host1(config)#**ip policy-list p2**
   host1(config-policy)#**classifier-group C1 precedence 90**
   host1(config-policy-classifier-group)#**forward next-hop 20.1.1.1**
   host1(config-policy-classifier-group)#**exit**
   host1(config-policy)#**classifier-group C3 precedence 10**
   host1(config-policy-classifier-group)#**filter**
   host1(config-policy-classifier-group)#**exit**
   host1(config-policy)#**classifier-group * precedence 1000**
   host1(config-policy-classifier-group)#**forward**
   host1(config-policy-classifier-group)#**exit**

3. Attach IP policy p1 as input at interface atm5/0.1.

   host1(config)#**Interface atm 5/0.1**
   host1(config-subif)#**ip policy input p1 statistics enable merge**
   host1(config-subif)#**exit**

4. Attach IP policy p2 as input at interface atm 5/0.1. A merged policy is created.

   host1(config)#**Interface atm 5/0.1**
   host1(config-subif)#**ip policy input p2 statistics enable merge**
   host1(config-subif)#**exit**

5. Display the policy lists.

   host1#**show policy-list**

```
                                  Policy Table
                                  ------ -----
IP Policy p1
   Administrative state: enable
   Reference count:      1
   Classifier control list: C2, precedence 10
      filter
   Classifier control list: C1, precedence 90
      forward
         Virtual-router: default
```

```
        List:
          next-hop 10.1.1.1, order 100, rule 2 (active)

  Referenced by interfaces:
     None

  Referenced by profiles:
     None

  Referenced by merge policies:
     mpl_5

IP Policy p2
  Administrative state: enable
  Reference count:      1
  Classifier control list: C3, precedence 10
     filter
  Classifier control list: C1, precedence 90
     forward
        Virtual-router: default
        List:
          next-hop 20.1.1.1, order 100, rule 3 (active)
  Classifier control list: *, precedence 1000
     forward

  Referenced by interfaces:
     None

  Referenced by profiles:
     None

  Referenced by merge policies:
     mpl_5

IP Policy mpl_5
  Administrative state: enable
  Reference count:      1
  Classifier control list: C2, precedence 10
     filter
  Classifier control list: C3, precedence 10
     filter
  Classifier control list: C1, precedence 90
     forward
        Virtual-router: default
        List:
          next-hop 10.1.1.1, order 100, rule 2 (active)
          next-hop 20.1.1.1, order 100, rule 3 (reachable)
  Classifier control list: *, precedence 1000
     forward

  Referenced by interfaces:
     ATM5/0.1  input policy, statistics enabled, virtual-router default

  Referenced by profiles:
     None

  Component policies:
     p1
     p2
```

6. Show configuration.

   host1#**show conf**

```
! Configuration script being generated on TUE APR 26 2005 17:33:01 UTC
! Juniper Edge Routing Switch ERX1440
! Version: 9.9.9 development-4.0 (April 4, 2005  15:39)
```



```
! Copyright (c) 1999-2005 Juniper Networks, Inc.  All rights reserved.
!
! Commands displayed are limited to those available at privilege level 15
!
…
interface atm 5/0.1
 ip policy input p1 statistics enabled merge
 ip policy input p2 statistics enabled merge
 exit
…
…
ip policy-list p1
 classifier-group C2 precedence 10
  filter
 classifier-group C1 precedence 90
  forward next-hop 10.1.1.1
!
ip policy-list p2
 classifier-group C3 precedence 10
  filter
 classifier-group C1 precedence 90
  forward next-hop 20.1.1.1
 classifier-group * precedence 1000
  forward
!
…
…
! End of generated configuration script.
```

7. Display interface statistics.

   host1#**show ip interface atm 5/0.1**

```
ATM5/0.1 line protocol Atm1483 is up, ip is up
  Network Protocols: IP
  Internet address is 99.99.99.2/255.255.255.0
  Broadcast address is 255.255.255.255
  Operational MTU = 9180  Administrative MTU = 0
  Operational speed = 155520000  Administrative speed = 0
  Discontinuity Time = 721112
  Router advertisement = disabled
  Proxy Arp = disabled
  Network Address Translation is disabled
  TCP MSS Adjustment = disabled
  Administrative debounce-time = disabled
  Operational debounce-time    = disabled
  Access routing = disabled
  Multipath mode = hashed
  Auto Configure = disabled
  Auto Detect = disabled
  Inactivity Timer = disabled

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Policed Packets 0, Bytes 0
  In Error Packets 0
  In Invalid Source Address Packets 0
  In Discarded Packets 0
  Out Forwarded Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Routed Packets 0, Bytes 0
  Out Scheduler Dropped Packets 0, Bytes 0
```

```
Out Policed Packets 0, Bytes 0
Out Discarded Packets 0

IP policy input mpl_5
  classifier-group C2 entry 1
    0 packets, 0 bytes
    filter
  classifier-group C3 entry 1
    0 packets, 0 bytes
    filter
  classifier-group C1 entry 1
    0 packets, 0 bytes
    forward
  classifier-group *
    0 packets, 0 bytes
    forward
queue 0: traffic class best-effort, bound to ip ATM5/0.1
  Queue length 0 bytes
  Forwarded packets 0, bytes 0
  Dropped committed packets 0, bytes 0
  Dropped conformed packets 0, bytes 0
  Dropped exceeded packets 0, bytes 0
```

8. Attach IP policy p1 at atm 5/0.2 as output.

   host1(config)#**interface atm 5/0.2**
   host1(config-subif)#**ip policy output p1 statistics enable merge**
   host1(config-subif)#**exit**

9. Attach IP policy p2 at atm 5/0.2 as output. Merge policy mpl_5 is now attached.

   host1(config)#**interface atm 5/0.2**
   host1(config-subif)#**ip policy output p2 merge**
   host1(config-subif)#**exit**

10. Display policies to verify that mpl_5 is created.

    host1#**show policy-list**

```
                                  Policy Table
                                  ------ -----
IP Policy p1
   Administrative state: enable
   Reference count:      1
   Classifier control list: C2, precedence 10
      filter
   Classifier control list: C1, precedence 90
      forward
         Virtual-router: default
         List:
           next-hop 10.1.1.1, order 100, rule 2 (active)

   Referenced by interfaces:
      None

   Referenced by profiles:
      None

   Referenced by merge policies:
      mpl_5

IP Policy p2
   Administrative state: enable
   Reference count:      1
   Classifier control list: C3, precedence 10
```



```
      filter
   Classifier control list: C1, precedence 90
      forward
         Virtual-router: default
         List:
           next-hop 20.1.1.1, order 100, rule 3 (active)
   Classifier control list: *, precedence 1000
      forward

   Referenced by interfaces:
      None

   Referenced by profiles:
      None

   Referenced by merge policies:
      mpl_5

IP Policy mpl_5
   Administrative state: enable
   Reference count:      2
   Classifier control list: C2, precedence 10
      filter
   Classifier control list: C3, precedence 10
      filter
   Classifier control list: C1, precedence 90
      forward
         Virtual-router: default
         List:
           next-hop 10.1.1.1, order 100, rule 2 (active)
           next-hop 20.1.1.1, order 100, rule 3 (reachable)
   Classifier control list: *, precedence 1000
      forward

   Referenced by interfaces:
      ATM5/0.1  input policy, statistics enabled, virtual-router default
      ATM5/0.2  output policy, statistics enabled, virtual-router default

   Referenced by profiles:
      None

   Component policies:
      p1
      p2
```

11. Create and attach IP policy p3 at atm 5/0.1. A new merge policy mpl_7 is created, which is a combination of p1, p2, and p3. The previous merge policy attachment is removed.

    host1(config)#**ip classifier-list C4 udp host 1.1.1.1 any eq 900**
    host1(config)#**ip policy-list p3**
    host1(config-policy)#**classifier-group C4 precedence 900**
    host1(config-policy-classifier-group)#**color red**
    host1(config-policy-classifier-group)#**exit**
    host1(config-policy)#**classifier-group C1 precedence 80**
    host1(config-policy-classifier-group)#**color yellow**
    host1(config-policy-classifier-group)#**exit**
    host1(config-policy)#**exit**
    host1(config)#**interface atm 5/0.1**
    host1(config-subif)#**ip policy input p3 statistics enable merge**
    host1(config-subif)#**exit**

12. Display policies to verify that mpl_5 and mpl_7 have been created.

host1#**show policy-list**

```
                                  Policy Table
                                  ------ -----
IP Policy p1
   Administrative state: enable
   Reference count:      2
   Classifier control list: C2, precedence 10
      filter
   Classifier control list: C1, precedence 90
      forward
         Virtual-router: default
         List:
           next-hop 10.1.1.1, order 100, rule 2 (active)

   Referenced by interfaces:
      None

   Referenced by profiles:
      None

   Referenced by merge policies:
      mpl_5
      mpl_7

IP Policy p2
   Administrative state: enable
   Reference count:      2
   Classifier control list: C3, precedence 10
      filter
   Classifier control list: C1, precedence 90
      forward
         Virtual-router: default
         List:
           next-hop 20.1.1.1, order 100, rule 3 (active)
   Classifier control list: *, precedence 1000
      forward

   Referenced by interfaces:
      None

   Referenced by profiles:
      None

   Referenced by merge policies:
      mpl_5
      mpl_7

IP Policy p3
   Administrative state: enable
   Reference count:      1
   Classifier control list: C1, precedence 80
      color yellow
   Classifier control list: C4, precedence 900
      color red

   Referenced by interfaces:
      None

   Referenced by profiles:
      None

   Referenced by merge policies:
      mpl_7
```

```
IP Policy mpl_5
   Administrative state: enable
   Reference count:      1
   Classifier control list: C2, precedence 10
      filter
   Classifier control list: C3, precedence 10
      filter
   Classifier control list: C1, precedence 90
      forward
         Virtual-router: default
         List:
           next-hop 10.1.1.1, order 100, rule 2 (active)
           next-hop 20.1.1.1, order 100, rule 3 (reachable)
   Classifier control list: *, precedence 1000
      forward

   Referenced by interfaces:
      ATM5/0.2  output policy, statistics enabled, virtual-router default

   Referenced by profiles:
      None

   Component policies:
      p1
      p2

IP Policy mpl_7
   Administrative state: enable
   Reference count:      1
   Classifier control list: C2, precedence 10
      filter
   Classifier control list: C3, precedence 10
      filter
   Classifier control list: C1, precedence 80
      forward
         Virtual-router: default
         List:
           next-hop 10.1.1.1, order 100, rule 2 (active)
           next-hop 20.1.1.1, order 100, rule 3 (reachable)
      color yellow
   Classifier control list: C4, precedence 900
      color red
   Classifier control list: *, precedence 1000
      forward

   Referenced by interfaces:
      ATM5/0.1  input policy, statistics enabled, virtual-router default

   Referenced by profiles:
      None

   Component policies:
      p1
      p2
      p3
```

13. Detach p2 from atm 5/0.1. A new merge policy mpl_8 is created, which is a combination of p1 and p3. The previous merge policy mpl_7 is detached and, because this policy has no attachments, it is deleted.

    host1(config)#**interface atm 5/0.1**
    host1(config-subif)#**no ip policy input p2**
    host1(config-subif)#**exit**

14. Display policies to verify that the mpl_7 is removed and the new merge policy mpl_8 is created.

    host1#**show policy-list**

```
                                  Policy Table
                                  ------ -----
IP Policy p1
   Administrative state: enable
   Reference count:      2
   Classifier control list: C2, precedence 10
      filter
   Classifier control list: C1, precedence 90
      forward
         Virtual-router: default
         List:
           next-hop 10.1.1.1, order 100, rule 2 (active)

   Referenced by interfaces:
      None

   Referenced by profiles:
      None

   Referenced by merge policies:
      mpl_5
      mpl_8

IP Policy p2
   Administrative state: enable
   Reference count:      1
   Classifier control list: C3, precedence 10
      filter
   Classifier control list: C1, precedence 90
      forward
         Virtual-router: default
         List:
           next-hop 20.1.1.1, order 100, rule 3 (active)
   Classifier control list: *, precedence 1000
      forward

   Referenced by interfaces:
      None

   Referenced by profiles:
      None

   Referenced by merge policies:
      mpl_5

IP Policy p3
   Administrative state: enable
   Reference count:      1
   Classifier control list: C1, precedence 80
      color yellow
   Classifier control list: C4, precedence 900
      color red

   Referenced by interfaces:
      None

   Referenced by profiles:
      None

   Referenced by merge policies:
      mpl_8
```



```
IP Policy mpl_5
   Administrative state: enable
   Reference count:      1
   Classifier control list: C2, precedence 10
      filter
   Classifier control list: C3, precedence 10
      filter
   Classifier control list: C1, precedence 90
      forward
         Virtual-router: default
         List:
           next-hop 10.1.1.1, order 100, rule 2 (active)
           next-hop 20.1.1.1, order 100, rule 3 (reachable)
   Classifier control list: *, precedence 1000
      forward

   Referenced by interfaces:
      ATM5/0.2  output policy, statistics enabled, virtual-router default

   Referenced by profiles:
      None

   Component policies:
      p1
      p2

IP Policy mpl_8
   Administrative state: enable
   Reference count:      1
   Classifier control list: C2, precedence 10
      filter
   Classifier control list: C1, precedence 80
      forward
         Virtual-router: default
         List:
           next-hop 10.1.1.1, order 100, rule 2 (active)
           next-hop 20.1.1.1, order 100, rule 3 (reachable)
      color yellow
   Classifier control list: C4, precedence 900
      color red

   Referenced by interfaces:
      ATM5/0.1  input policy, statistics enabled, virtual-router default

   Referenced by profiles:
      None

   Component policies:
      p1
      p3
```

15. Detach p1 from atm 5/0.1. Merge policy mpl_8 is detached and deleted, and only p3 is attached to this interface.

    host1(config)#**interface atm 5/0.1**
    host1(config-subif)#**no ip policy input p1**
    host1(config-subif)#**exit**

16. Display policies to verify that p3 is attached to atm 5/0.1 and mpl_8 is removed.

    host1#**show policy-list**

```
                                  Policy Table
                                  ------ -----
IP Policy p1
```

```
   Administrative state: enable
   Reference count:      1
   Classifier control list: C2, precedence 10
      filter
   Classifier control list: C1, precedence 90
      forward
         Virtual-router: default
         List:
           next-hop 10.1.1.1, order 100, rule 2 (active)

   Referenced by interfaces:
      None

   Referenced by profiles:
      None

   Referenced by merge policies:
      mpl_5

IP Policy p2
   Administrative state: enable
   Reference count:      1
   Classifier control list: C3, precedence 10
      filter
   Classifier control list: C1, precedence 90
      forward
         Virtual-router: default
         List:
           next-hop 20.1.1.1, order 100, rule 3 (active)
   Classifier control list: *, precedence 1000
      forward

   Referenced by interfaces:
      None

   Referenced by profiles:
      None

   Referenced by merge policies:
      mpl_5

IP Policy p3
   Administrative state: enable
   Reference count:      1
   Classifier control list: C1, precedence 80
      color yellow
   Classifier control list: C4, precedence 900
      color red

   Referenced by interfaces:
      ATM5/0.1  input policy, statistics disabled, virtual-router default

   Referenced by profiles:
      None

   Referenced by merge policies:
      None

IP Policy mpl_5
   Administrative state: enable
   Reference count:      1
   Classifier control list: C2, precedence 10
      filter
   Classifier control list: C3, precedence 10
      filter
   Classifier control list: C1, precedence 90
```



```
      forward
         Virtual-router: default
         List:
            next-hop 10.1.1.1, order 100, rule 2 (active)
            next-hop 20.1.1.1, order 100, rule 3 (reachable)
   Classifier control list: *, precedence 1000
      forward

   Referenced by interfaces:
      ATM5/0.2  output policy, statistics enabled, virtual-router default

   Referenced by profiles:
      None

   Component policies:
      p1
      p2
```

17. Detach p3 from atm 5/0.1.

    host1(config)#**interface atm 5/0.1**
    host1(config-subif)#**no ip policy input p3**
    host1(config-subif)#**exit**

18. Detach p1 from atm 5/0.2. Merge policy mpl_5 is detached and deleted and only p2 is now attached.

    host1(config)#**interface atm 5/0.2**
    host1(config-subif)#**no ip policy output p1**
    host1(config-subif)#**exit**

19. Detach p2 from atm 5/0.2.

    host1(config)#**interface atm 5/0.2**
    host1(config-subif)#**no ip policy output p2**
    host1(config-subif)#**exit**

20. Display policies to verify that no merge policies exist and that all other policies have a 0 reference count because they are not attached anywhere.

    host1#**show policy-list**

```
                                  Policy Table
                                  ------ -----
IP Policy p1
   Administrative state: enable
   Reference count:      0
   Classifier control list: C2, precedence 10
      filter
   Classifier control list: C1, precedence 90
      forward
         Virtual-router: default
         List:
            next-hop 10.1.1.1, order 100, rule 2 (active)

IP Policy p2
   Administrative state: enable
   Reference count:      0
   Classifier control list: C3, precedence 10
      filter
   Classifier control list: C1, precedence 90
      forward
         Virtual-router: default
         List:
```



```
            next-hop 20.1.1.1, order 100, rule 3 (active)
    Classifier control list: *, precedence 1000
       forward

 IP Policy p3
    Administrative state: enable
    Reference count:      0
    Classifier control list: C1, precedence 80
       color yellow
    Classifier control list: C4, precedence 900
       color red
```

Related Documentation

- Merging Policies Overview on page 103
- Monitoring Policy Management Overview on page 197
- atm classifier-list
- classifier-group
- color
- filter
- forward next-hop
- interface atm
- show ip interface
- show policy-list

Parent Group Merge Algorithm

The parent group merge algorithm enables the system to merge policies that contain references to parent groups and create an internal parent group for each internal parent group in a component policy in the final merged policy. There is a one-to-one correspondence between an internal parent group in the merged policy and an internal parent group in a component policy.

NOTE: **The naive parent group merging algorithm is not compatible with this parent group merge algorithm. If you have service definitions that used the naive parent group algorithm, you need to modify those service definitions to work with this algorithm.**

- If there is no existing internal parent group with the same name in the merged policy, the system creates a corresponding internal parent group with the same name.
- If an internal parent group with the same name already exists, the system uses a name built by appending an internally generated sequence number to the name of the internal parent group in the component policy.
- If the length of the name exceeds the maximum length allowed, the policy merge fails.

- If a classifier group in a component policy refers to an internal parent group, the same classifier group in the merged policy corresponds to the internal parent group in the merged policy.
- If a classifier group in a component policy refers to an external parent group, the same classifier group in the merged policy refers to the same external parent group.
- If there is a conflict where two or more component policies contain the same classifier group referring to an internal parent group in a corresponding component policy or to an external parent group, then last one is used.

In the following example, component policies P1 and P2 create the merged policy mpl_88000001.

host1#**show policy-list P1**

```
                                  Policy Table
                                  ------ -----
IP Policy P1
   Administrative state: enable
   Reference count:      1
   Classifier control list: *, precedence 100, parent-group Z
   forward
   Classifier control list: A, precedence 100, parent-group X
forward
   Classifier control list: B, precedence 100, parent-group X
forward
   Classifier control list: C, precedence 100, external parent-group EPG1
parameter foo
forward
  Classifier control list: D, precedence 100, external parent-group EPG1 parameter
 foo
forward

   Parent group: X, parent-group Z
      rate-limit-profile R1
   Parent group: Z
      rate-limit-profile R2
```

host1#**show policy-list P2**

```
                                  Policy Table
                                  ------ -----
IP Policy P2
   Administrative state: enable
   Reference count:      1
   Classifier control list: B, precedence 100, parent-group X
forward
   Classifier control list: C, precedence 100, parent-group Y
       forward
  Classifier control list: D, precedence 100, external parent-group EPG2 parameter
 abcd
forward

   Parent group: X, parent-group Y
      rate-limit-profile R3
   Parent group: Y
      rate-limit-profile R4
```

```
host1#show policy-list mpl_88000001

                                  Policy Table
                                  ------ -----

IP Policy mpl_88000001
Administrative state: enable
Reference count:      1
   Classifier control list: *, precedence 100, parent-group Z
forward
   Classifier control list: A, precedence 100, parent-group X
forward
   Classifier control list: B, precedence 100, parent-group X_1
forward
   Classifier control list: C, precedence 100, parent-group Y
forward
  Classifier control list: D, precedence 100, external parent-group EPG2 parameter
 abcd
forward

   Parent group: X, parent-group Z
      rate-limit-profile R1
   Parent group: Z
      rate-limit-profile R2
   Parent group: X_1, parent-group P2_Y
      rate-limit-profile R3
   Parent group: Y
      rate-limit-profile R4

   Referenced by interfaces:
      ATM5/0.1  input policy, statistics enabled, virtual-router default

   Referenced by profiles:
      None

   Component policies:
      P1
      P2
```

Related Documentation

- External Parent Groups on page 132

Overlapping Classification for IP Input Policy

IP auxiliary input policy can be used with IP input policy to provide overlapping classification. Two policies, each with a set of independent rules and actions, run in sequence so that each policy can independently produce a set of actions in sequence. A packet that matches both the input policies and auxiliary input policies is subject to both sets of policy actions.

E Series routers allow four input and two output policies per IP interface:

- One secure input policy
- Three nonsecure input policies
- One secure output policy
- One nonsecure output policy

Each classifier-group has a set of associated actions that is taken if it is the highest priority match. The system performs only one set of actions per policy attachment. By using an input and secondary-input policy, you can have overlapping classification with multiple policy actions on ingress. Overlapping classification on egress is not supported.

An additional policy attachment point enables overlapping classification within the input classification stage, between the input and secondary-input stages. There are five attachment points for IP policies that are executed in series:

- input
- secondary-input
- secure-input
- output
- secure-output

An explicit filter action, a forward action with a null next-interface, or a rate-limit action can cause an immediate packet discard at any stage. Other actions, such are marking and coloring can be done at each stage, with the last of each of these actions taking precedence over the others.

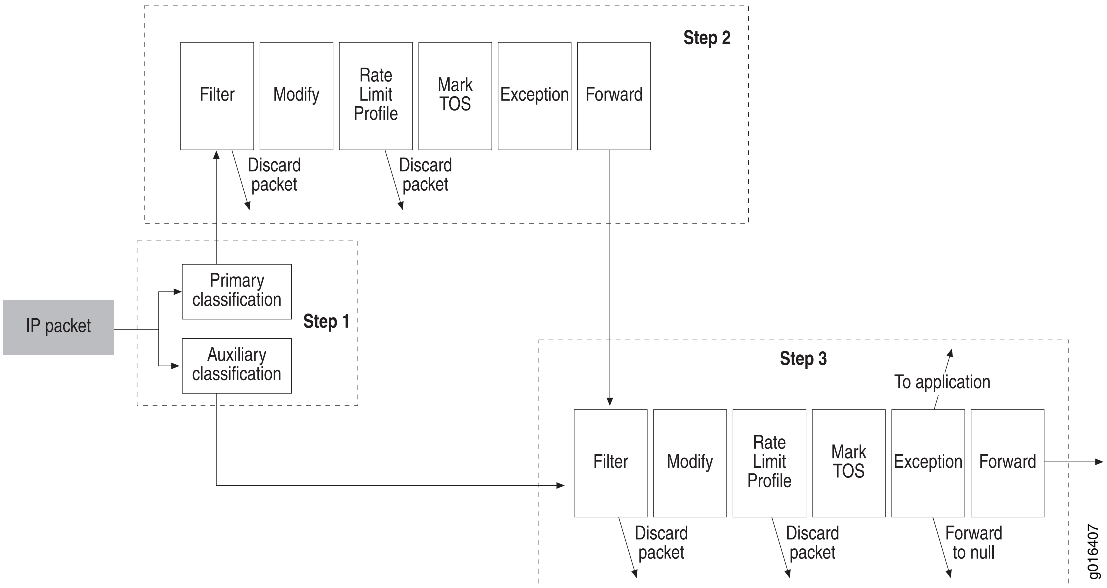
For example, unique policies can be attached at each stage, all of which mark the IP TOS field differently. The packet then exits the router with the TOS value that was set in the output policy stage. However, if TOS is also used as a classification (input) term for each of these policies, three different TOS values are presented to the classifier:

- Original TOS received
- TOS modified by the input policy
- TOS value modified by the secondary-input policy

Figure 7 on page 126 shows the input policy stage after the addition of the auxiliary substage. It is divided into three steps:

1. Apply classification for both substages.
2. Perform policy actions (if any) for the primary attachment.
3. Perform policy actions (if any) for the auxiliary attachment.

Figure 7: Input Policy with Primary Stage and Auxiliary Substage



The order of policy action execution for each attachment is:

1. Filter
2. Modify (includes setting of color, traffic class, user packet class) and Log
3. Rate limit profile/color
4. Mark TOS
5. Exception
6. Forward

Starting Policy Processing

Input and auxiliary-input classification operations, specified by the details of each policy, are performed in parallel. Classifier inputs for both policies are determined concurrently using the initial values of the classification terms. Policy attachments within a stage cannot communicate between the input and auxiliary-input classification operations. For example, any changes made by the input attachment to traffic-class, color, TOS, or user packet class are not visible in the auxiliary-input policy classification. If this communication is needed, it can only be done between different policy stages, rather than within a single stage.

The results of the input policy actions are passed forward to the auxiliary-input policy action processing. This means that a color-aware rate limit profile action in the auxiliary substage recognizes any change in color caused by primary policy actions.

Processing the Classifier Result

The classifier result of the input policy attachment is processed and a set of actions is identified. When you configure filter, it is the first action taken and immediately discards the packet. This is followed by any modification, such as mark or logging. If a rate limit profile is configured, the packet is dropped or colored. If the packet is not dropped, it is sent to the exception path (if configured). If the packet is not exceptioned, any configured forward action is saved in the packet for use later (unless overridden in Step 3). (See Figure 7 on page 126.)

Some information generated by the action processing in Step 2 is forwarded to Step 3, where it may affect the action processing for the auxiliary-input attachment. This information can include color, exception information, and forwarding information. The color can affect a rate-limit in the auxiliary-input attachment. Step 3 acts on the exception and forwarding information, if it is not overridden by similar actions from the auxiliary-input attachment.

The transmit information (transmit conditional, transmit unconditional, transmit final) generated with hierarchical policies does not carry forward from input to auxiliary-input action processing.

Processing the Auxiliary-Input Policy Attachment

If the packet is not filtered or exceptioned in policy Step 2, the classifier result of the auxiliary policy attachment is processed and a set of actions identified. The packet can be filtered or exceptioned at this time. These operations, if configured, are performed regardless of whether a forward action was performed in Step 2. If the packet is not discarded, either by a filter action or a rate limit, it can be exceptioned (if configured). If the packet is not filtered, rate-limited, or exceptioned, any configured forward action is applied and overrides any forward action from Step 2. If no forward action is configured, any forward action from Step 2 applies.

Policy Actions

The set of actions in the following list specified by the input and auxiliary-input policy attachments are executed in the order: input, auxiliary-input.

- Color packet action—Explicitly sets the packet color. Each policy attachment can set the color and the final value persists. A rate limit profile action can also set the color, which overrides the value of the color packet action.
- Mark action—Each attachment can set the TIP TOS, TOS precedence, and DS fields. The cumulative result of all configured mark actions determines the resulting value of these fields.
- Mirror action—Executes in the order: secure input policy follows secondary input policy, secure output policy follows output policy. Mirror is the only supported action for secure policies.
- Rate-limit profile action—Can be specified by any nonsecure input policy attachment. This enables the application of multiple rate limits either within a policy stage or across policy stages. These rate limits run serially; if the rate limit imposed in the primary substage causes the packet to drop, the auxiliary rate limit does not run and the associated token buckets are not affected. If you configure more than a single rate limit per interface, it significantly impacts forwarding performance. Attaching two

policies with rate limit profiles in the same policy stage is equivalent to having two policies attached in the same order, but in separate stages.

- Traffic class action—If both the input and auxiliary-input attachments need this action, the value configured in the auxiliary policy overwrites that of the primary policy.
- User packet class action—Can be set twice per stage, with the second value overriding the first.
- The filter, next-hop, forward interface, and forward next-hop actions are mutually exclusive within a classifier group. However, two policies in series can result in conflicting actions, which are resolved using the following precedence rules:
  - The filter action has highest priority. A filter action in input or auxiliary-input policy always prevails.
  - The exception action takes precedence over forward actions.
  - If multiple exception actions are required by the policy attachments, the last one takes precedence.
  - If forward operations are required by both input and auxiliary-input policy attachments, the auxiliary-input forward action takes precedence.

Input policy attachments depend on the **local** keyword in classifier list entries. Using the **local false** keyword or using no local keyword (default) treats both local and non-local traffic equally and ignores the local true classifier list entries.

Secondary input policies affect both local and non-local traffic and are processed by policies attached with the **secondary-input** keyword. Secondary input policies are controlled by the **local** keyword in the classifier list entries as follows:

- **local true** keyword only affects local traffic
- **local false** keyword only affects non-local traffic
- no **local** keyword (default) affects both local and non-local traffic

In Table 15 on page 128, the filter action for the input policy takes precedence over the others so that if a filter action is configured for either policy, the packet is filtered. If neither policy has a filter action, but both policies specify a forward action, the action specified by the auxiliary policy takes precedence. If only one policy specifies a forwarding action, that action is executed. The next-hop rule is inoperative for auxiliary-input policies, just as it is for secondary input policies. This policy rule has been superseded (but not replaced) by the forward next-hop rule, which is operative for auxiliary-input policies.

Table 15: Input Action and Secondary Input Actions

Input Action	Secondary Input Action					
	None	Exception	Filter	Next-hop	Forward Interface	Forward Next-hop

Table 15: Input Action and Secondary Input Actions *(continued)*

Input Action	Secondary Input Action					
None	None	Exception Auxiliary	Filter	None	Forward Interface Auxiliary	Forward Next-hop Auxiliary
Exception	Exception Primary	Exception Auxiliary	Filter	Exception	Exception Primary	Exception Primary
Filter	Filter	Filter	Filter	Filter	Filter	Filter
Next-hop	Next-hop Primary	Exception Auxiliary	Filter	Next-hop Primary	Forward Interface Auxiliary	Forward Next-hop Auxiliary
Fwd Interface	Forward Interface Primary	Exception Auxiliary	Filter	Forward Interface Primary	Forward Interface Auxiliary	Forward Next-hop Auxiliary
Fwd next-hop	Forward Next-hop Primary	Exception Auxiliary	Filter	Forward Next-hop Primary	Forward Interface Auxiliary	Forward Next-hop Auxiliary

Related Documentation

- Policy Attachment Rules for Merged Policies on page 109

CHAPTER 7

Creating Hierarchical Policies for Interface Groups

This chapter provides information for configuring policy-based routing management on E Series routers.

This chapter discusses the following topics:

Hierarchical Policies for Interface Groups Overview

Hierarchical policies allow classifier groups and parent groups within a policy to point to line module global parent groups. The line module global parent groups (external parent groups) can point to other external parent groups. Full intra-interface policy hierarchies for all forwarding layer policies allow classified flows within a policy attachment to share bandwidth. Bandwidth-sharing between interfaces uses line module global parent group definitions and interface grouping. However, if you need to share bandwidth between two or more interfaces, rate-limits must be chained beyond a single attachment.

Policies for interface groups include external parent groups that are implicitly instantiated during policy attachment based on each unique interface group encountered.

Related Documentation

- Hierarchical Policy Configuration Considerations on page 136

External Parent Groups

Parent groups act as nonleaf nodes in a hierarchical policy. You can build a hierarchy of policers using classifier groups as leaf nodes and parent groups as parent nodes within a policy list. Each classifier group (with or without a rate limit) can point to a single parent group and that parent group can point to another parent group. To avoid undefined hierarchies, each node can only point to one other node.

The inter-interface hierarchical model includes references to parent groups that are defined externally from a policy list. This enables you to define hierarchical nodes outside the scope of a policy-list attachment. In Global Configuration mode, each external parent group can have a rate-limit profile defined and have a reference to another external parent group.

The classifier groups and parent groups within a policy list can point to external parent groups for all policies that implement hierarchical policies. Each external parent group reference must also have a policy parameter name.

External parent group names are global. Internal parent group names are local to each policy configuration. Because both of these name spaces are different, you can configure overlapping names.

Related Documentation

- Parent Group Merge Algorithm on page 122
- Monitoring External Parent Groups on page 221

Example: Configuring Hierarchical Policy Parameters

You configure policy parameters in Global Configuration mode. Only hierarchical policy parameters can have external parent group references. Each parameter has a single value, depending on the type of parameter. The hierarchical policy parameter can have a single numeric value or a keyword.

In Interface Configuration mode, you can override the value for a policy parameter for each interface. The value for a parameter configured in Interface Configuration mode supersedes the value configured for the parameter in Global Configuration mode. However, if a parameter is not configured in Interface Configuration mode, the value configured in Global Configuration mode is used.

Each reference to a policy parameter in a policy is substituted with its value for all attachments of this policy at the interface. The value can come from the interface or global configuration for the parameter. Therefore, the value configured for the parameters referenced in policies can be different for attachments at different interfaces. This enables you to have an attachment-specific configuration in a policy list that is deferred until the policy is attached.

There are two types of values that a hierarchical policy parameter can take: numeric and keyword. Keywords are resolved to numeric values during configuration of a policy parameter at the interface.

The following example assigns a value of 10 to policy parameter A in Global Configuration mode.

```
host1(config)#policy-parameter A hierarchical
host1(config-policy-parameter)#aggregation-node 10
host1(config-policy-parameter)#exit
```

The following example assigns value 1 to policy parameter A and value 2 to policy parameter B in Interface Configuration mode. Also, the value configured for parameter A in interface fast3/0.1 overrides the value configured in the previous example.

```
host1(config)#interface fastEthernet 3/0.1
host1(config-interface)#ip policy-parameter hierarchical A 1
host1(config-interface)#ip policy-parameter hierarchical B 2
host1(config-interface)#exit
```

The following example assigns keyword **vlan** to parameter C in Global Configuration mode.

```
host1(config)#policy-parameter C hierarchical
host1(config-policy-parameter)#aggregation-node vlan
host1(config-policy-parameter)#exit
```

The following example assigns keyword **atm-vc** to parameter C in Interface Configuration mode. Policy parameter C is assigned with interface type atm-vc for IP interface at atm3/0.1. The keyword **atm-vc** is resolved to the identifier of the ATM minor interface on which the IP interface atm3/0.1 is stacked.

```
host1(config)#interface atm 3/0.1
host1(config-interface)#ip policy-parameter hierarchical C atm-vc
host1(config-interface)#exit
```

The following keywords are supported: **atm-vc**, **atm-vp**, **atm**, **ethernet**, **vlan**, **svlan**, **fr-vc**, **forwarding**, and **ppp-interface**. Table 16 on page 133 indicates the mapping of shorthand notation to actual value that are used internally.

Table 16: Shorthand Notation Mapping

Shorthand number	Shorthand	Value	Supported in
1	ATM-VP vpi	Identifier constructed from slot, adapter, port, ATM VP id.	IP, IPv6, L2TP, and MPLS policies

Table 16: Shorthand Notation Mapping *(continued)*

Shorthand number	Shorthand	Value	Supported in
2	ATM-VC	Unique identifier of the ATM minor interface	IP, IPv6, and MPLS policies
3	Ethernet	Unique identifier of Ethernet major interface	IP, IPv6, and MPLS policies
4	VLAN	Unique identifier of VLAN interface	IP, IPv6, and MPLS policies
5	SVLAN	Identifier constructed from slot, adapter, port, SVLAN ID.	IP, IPv6, L2TP, and MPLS policies
6	FR-VC	Unique identifier of frame relay minor interface	IP, IPv6, and MPLS policies
7	ATM	Unique identifier of ATM major interface	IP, IPv6, and MPLS policies
8	Forwarding	Unique identifier of the forwarding interface where the parameter is configured.	IP, IPv6, L2TP, and MPLS policies
9	ppp-interface	Identifier constructed from PPP interface.	IP, IPv6 policies

Hierarchical Aggregation Nodes

An internal parent group configured within a policy defines a hierarchical aggregation node template. An attachment of this policy creates an aggregation node for each internal parent group in a policy. Aggregation nodes are scoped within a single attachment and cannot be shared beyond a single attachment. An aggregation node stores a single rate-limit instance and statistics for this rate-limit. Aggregate nodes can be shared between two or more classified flows within a single attachment using the classifier group and parent group association.

Rate-limit aggregation nodes extend beyond a single attachment so classified flows across two or more attachments can reference the same aggregation node to share a single rate-limit instance. You can use external parent groups and policy parameters for sharing aggregate nodes across policy attachments. Each external parent group reference in a policy is accompanied by a parameter that is resolved during the attachment of the policy to an interface. An external rate-limit aggregation node can be defined by the 4-tuple (slot, direction, external parent group name, parameter value). The slot is the logical number of the line module location and the direction can be ingress or egress at the line module.

When you use hierarchical aggregation nodes, be aware of the following:

- VR/VRF—The hierarchical aggregate nodes based on external parent groups are not virtual router sensitive. The configuration allows interfaces from different virtual routers to have the same parameter name to value mapping, in which case both interfaces could share the same aggregate node created by an external parent group.

- Direction of Traffic—Hierarchical aggregate nodes are direction sensitive. The configuration does not allow input and output traffic at an interface to share the same rate-limit instance. Even when the input and output policy attachments refer to the same external parent group and parameter value, two separate aggregate nodes are created for each direction.
- Line Module—You should use hierarchical aggregate nodes. Rate limits cannot be shared across different line modules or service modules. Even when you configure the same parameter name to the same value for an external parent group, different rate-limit instances are instantiated if the interfaces are on different line modules.

Related Documentation

- Policy Attachment Rules for Merged Policies on page 109

RADIUS and Profile Configuration for Hierarchical Policies

You can use profiles to configure policy parameters. There is currently no RADIUS VSA support for policy parameters. Each reference to an external parent group and the chain of references from that group to other parent groups in a series requires one parent group resource for each reference and each attachment of the policy containing these references.

The rule that applies to external parent group resource count is: one resource per (interface, policy attachment type, policy name, external parent group name, parameter name) tuple; interface is the interface where the policy is attached and policy attachment type is the type of policy attachment.

A rate-limit instance for the external parent groups is created for each hierarchical aggregation node, which is a combination of (slot, direction, parent group name, parameter value) tuple; where slot is the slot number, direction is ingress or egress. A rate-limit resource will be consumed for each instance created.

If at least one policy attachment that uses an external parent group reference has statistics enabled, then statistics for the rate-limit configured within the external parent group is enabled. Each hierarchical aggregation node requires five statistics resources.

Related Documentation

- Hierarchical Policies for Interface Groups Overview on page 131

Interface Profiles for Service Manager Overview

Applying a profile to the interface where the subscriber sends and receives traffic activates service for a subscriber. Similarly, to deactivate a service, you reapply the respective profile with a negate flag.

You can use a profile to apply the policy parameters configuration for an interface. When you apply a profile containing relevant policy parameter commands to an interface, the parameter configuration is uniquely maintained for each dynamic interface created using this profile. The policy parameters are not deactivated when the corresponding service containing them is deactivated and can only be modified or created by service activations.

If you write service manager macros, you should define the rate-limit hierarchy when you create the policies and profiles associated with the services to be deployed.

Related Documentation

- Hierarchical Policies for Interface Groups Overview on page 131

Hierarchical Policy Configuration Considerations

When you configure hierarchical policies for interface groups, be aware of the following considerations:

- Loops—The system performs basic checks to prevent formation of loops when external parent groups refer to other external parent groups. Also, you cannot chain together more than four rate-limits in a hierarchy.
- Asynchronous Policy Parameter Configuration—You can individually configure the policy parameter configuration in an interface and the policy attachments. If a policy parameter is not configured in the interface before a policy is attached, the value configured in Global Configuration mode for this parameter is used. You can later change the parameter value for the interface.
- Asynchronous Parent Group Rate Limit Configuration—You can configure an external parent group without a rate-limit-profile reference. In this case, the system does not invoke a rate-limit for the external parent group (even if other nodes point to it) and calls the next node in the hierarchy.
- Parent Group Reference—The configuration fails if you do not first create an external parent group before it is referenced elsewhere.

Related Documentation

- Hierarchical Policies for Interface Groups Overview on page 131

Example: Hierarchical Policy Quick Configuration

To configure hierarchical policies for interface groups, use the following steps:

1. In Global Configuration mode, create rate limit profiles of the type hierarchical.
2. In Global Configuration mode, create policy parameters of the type hierarchical.
3. In Global Configuration mode, create external parent groups.
4. In Global Configuration mode, create a policy list and use the external parent groups and policy parameters to create a hierarchy of rate limits.
5. In Interface Configuration mode, attach the policy list to the interface.
6. (Optional) In Interface Configuration mode, specify values for the hierarchical policy parameters used by the policy list.

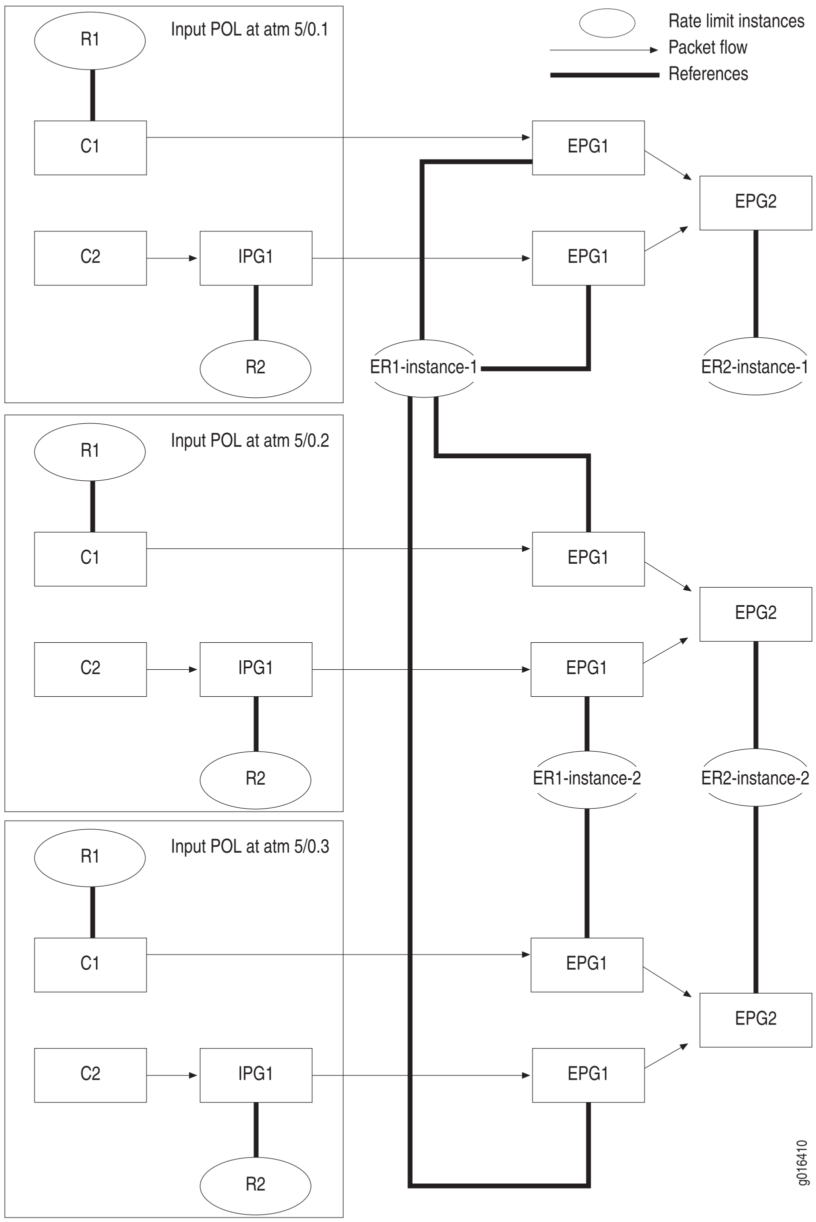
Related Documentation

- Example: Configuring Hierarchical Policies on page 137

Example: Configuring Hierarchical Policies

The configuration in Figure 8 on page 137 requires four parent group resources for each atm5/0.1, atm5/0.2, and atm5/0.3 attachment. The rate-limit instance R1 is referenced by C1 and packet flows from C1 to EPG1 to EPG2.

Figure 8: Configuration Process



This procedure uses the following designations:

- EPG1 and EPG2 are external parent groups.
- IP1 and IP2 are internal parent groups.

- ER1, ER2, R1, and R2 are rate-limit profiles.
- POL is the name of the IP policy.
- C1 and C2 are classified flows.
- A, B, and C are policy parameters.

1. Configure two external parent groups EPG1 and EPG2. Create policy-parameter C and two external parent groups: EPG1 and EPG2.

   host1(config)#**policy-parameter C hierarchical**
   host1(config-policy-parameter)#**exit**

   host1(config)#**parent-group EPG2**
   host1(config-parent-group)#**rate-limit-profile ER2**
   host1(config-parent-group)#**exit**

   host1(config)#**parent-group EPG1**
   host1(config-parent-group)#**next-parent EPG2 parameter C**
   host1(config-parent-group)#**rate-limit-profile ER1**
   host1(config-parent-group)#**exit**

   EPG1 contains a rate-limit profile ER1 and points to EPG2 as the next parent group in series. The EPG2 reference is associated with policy parameter C. When you later use the **policy-parameter** command in Interface Configuration mode, actual values are substituted for the names. EPG2 contains a reference to rate-limit-profile ER2.

2. Configure IP policy list POL.

   host1(config)#**ip policy-list POL**
   host1(config-policy-list)#**classifier-group C1 external parent-group EPG1 parameter A**
   host1(config-policy-list-classifier-group)#**rate-limit-profile R1**
   host1(config-policy-list-classifier-group)#**exit**

   host1(config-policy-list)#**classifier-group C2 parent-group IPG1**
   host1(config-policy-list-classifier-group)#**forward**
   host1(config-policy-list-classifier-group)#**exit**

   host1(config-policy-list)#**parent-group IPG1 external parent-group EPG1 parameter B**
   host1(config-parent-group)#**rate-limit-profile R2**
   host1(config-policy-list-parent-group)#**exit**

   A classified flow C1 references EPG1 as the next parent group to call in the hierarchy. This is an external parent group that is associated with policy parameter A. The C2 classified flow points to internal parent group IPG1, which contains rate-limit-profile R2 and points to EPG1 as the next parent group to call in the hierarchy. The EPG1 reference is associated with policy parameter B. When you later use the **policy-parameter** command in Interface Configuration mode, the policy parameters are given numeric values.

3. Attach POL to atm5/0.1 as an input policy.

   host1(config)#**interface atm 5/0.1**

host1(config-interface)#**ip policy-parameter hierarchical A 1**
host1(config-interface)#**ip policy-parameter hierarchical B 1**
host1(config-interface)#**ip policy-parameter hierarchical C 1**
host1(config-interface)#**ip policy input POL statistics enabled**
host1(config-interface)#**exit**

Policy list POL contains three parameter names that must be substituted with actual values. This attachment contains two internal rate-limit instances, one for R1 and one for R2. This attachment also contains one parent group instance for IPG1, one parent-group instance for (EPG1, parameter A) tuple, one for (EPG1, parameter B) tuple, and one for (EPG2, parameter C) tuple. Value number 1 is substituted for parameters A, B, and C when you use the **policy-parameter** command. Because of this policy attachment and the **policy-parameter** command, the following aggregation nodes are created: (slot 5, ingress, EPG1, 1), (slot 5, ingress, EPG2, 1). The system creates a rate-limit instance for each aggregation node: ER1-instance-1 and ER2-instance-1, respectively. ER1-instance-1 is referenced in parent-group instances (EPG1, parameter A) and (EPG1, parameter B). ER2-instance-1 is referenced in the parent group instance (EPG2, parameter C).

4. Attach POL to atm5/0.2 as input policy.

   host1(config)#**interface atm 5/0.2**
   host1(config-interface)#**ip policy-parameter hierarchical A 1**
   host1(config-interface)#**ip policy-parameter hierarchical B 2**
   host1(config-interface)#**ip policy-parameter hierarchical C 2**
   host1(config-interface)#**ip policy input POL statistics enabled**
   host1(config-interface)#**exit**

   Policy list POL contains three parameter names that must be substituted with actual values. This attachment consumes two internal rate-limit instances: one for R1 and one for R2. This attachment also consumes one parent group instance for IPG1, one parent-group instance for (EPG1, parameter A) tuple, one for (EPG1, parameter B) tuple, and one for (EPG2, parameter C) tuple as in Step 3. When you use the **policy-parameter** command, parameter A is substituted with value 1 and parameters B and C are substituted with value 2. Because of this policy attachment and the policy-parameter commands, the following aggregation nodes are identified: (slot 5, ingress, EPG1, 1), (slot 5, ingress, EPG1, 2), (slot 5, ingress, EPG2, 2). The (slot 5, ingress, EPG1, 1) node was already created in Step 3 and was named ER1-instance-1. The other two aggregation nodes are now created and named ER1-instance-2 and ER2-instance-2, respectively. ER1-instance-1 is referenced by parent-group instance (EPG1, parameter A), ER1-instance-2 is referenced by parent group instance (EPG1, parameter B), and ER2-instance-2 is referenced by the parent group instance (EPG2, parameter C).

5. Attach POL to atm5/0.3 as input policy.

   host1(config)#**interface atm 5/0.3**
   host1(config-interface)#**ip policy-parameter hierarchical A 2**
   host1(config-interface)#**ip policy-parameter hierarchical B 1**
   host1(config-interface)#**ip policy-parameter hierarchical C 2**
   host1(config-interface)#**ip policy input POL statistics enabled**
   host1(config-interface)#**exit**

   Policy list POL contains three parameter names that need to be substituted with actual values. This attachment consumes two internal rate-limit instances: one for

R1 and one for R2. This attachment also consumes one parent group instance for IPG1, one parent-group instance for (EPG1, parameter A) tuple, one for (EPG1, parameter B) tuple, and one for (EPG2, parameter C) tuple. When you use the **policy-parameter** command, parameters A and C are substituted with value 2 and parameter B is substituted with value 1. Because of this policy attachment and use of the **policy-parameter** commands, the following aggregation nodes are identified; (slot 5, ingress, EPG1, 2), (slot 5, ingress, EPG1, 1), (slot 5, ingress, EPG2, 2). All three aggregation nodes were created in earlier steps and were named ER1-instance-2, ER1-instance-1, and ER2-instance-2, respectively. ER1-instance-2 is referenced by parent-group instances (EPG1, parameter A), ER1-instance-1 is referenced by parent group instance (EPG1, parameter B), and ER2-instance-2 is referenced by the parent group instance (EPG2, parameter C).

Related Documentation

- Example: Hierarchical Policy Quick Configuration on page 136

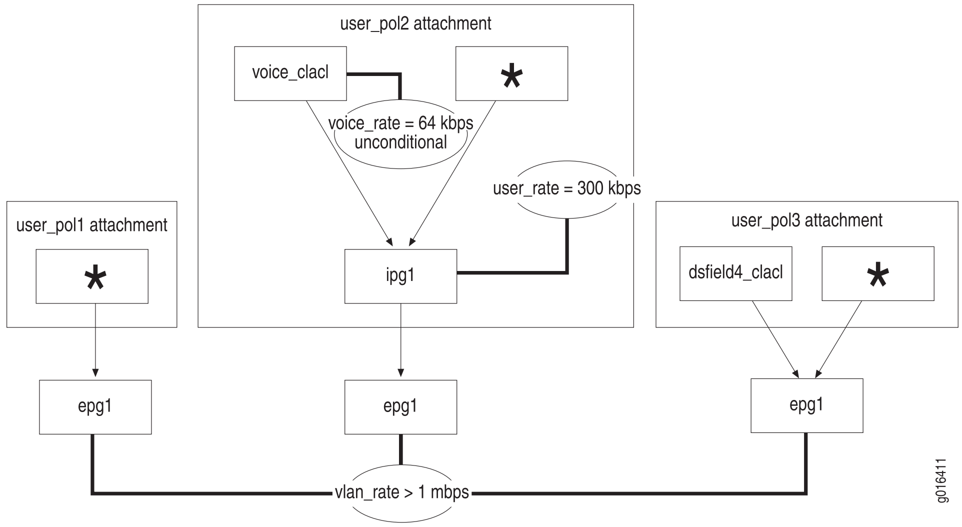
Example: VLAN Rate Limit Hierarchical Policy for Interface Groups Configuration

In this example, three users from a small business office are connected to an E Series router through the same VLAN interface. The contracted maximum for the business is 1 Mbps in the upstream direction. The downstream direction is served through QoS profiles and therefore is not shown here.

Figure 9 on page 141 shows the following:

- User user_pol1 is attached to the first user's IP interface and does not have a rate limit.
- User user_pol2 is attached the second user's interface and has an individual rate limit of 300Kbps and preferred voice traffic at 64Kbps.
- User user_pol3 is attached to the third user's interface and has some traffic marked with a low delay (Dsfield = 4), but there are no rate limitations applied.
- Policer instance VLAN_RATE is shared across all three instances of EPG1 and limits the total upstream traffic from three users to 1 Mbps.

Figure 9: VLAN Rate-Limit Configuration



1. Create a rate limit to enforce the contracted maximum for the small business. Create an external parent group to hold this rate limit.

   host1(config)#**rate-limit-profile VLAN_RATE two-rate hierarchical**
   host1(config-rate-limit-profile)#**committed-rate 1000000**
   host1(config-rate-limit-profile)#**committed-action transmit final**
   host1(config-rate-limit-profile)#**exit**

   host1(config)#**parent-group EPG1**
   host1(config-parent-group)#**rate-limit-profile VLAN_RATE**
   host1(config-parent-group)#**exit**

   Verify the parent group configuration.

   host1#**show parent-group EPG1**

```
                                  Parent Group Table
                                  ------ ----- ------
Parent Group EPG1
   Reference count: 0
   Rate limit profile: VLAN_RATE
```

2. Create a policy list to attach to user 1.

   host1(config)#**policy-parameter A hierarchical**
   host1(config-policy-parameter)#**exit**

   host1(config)#**ip policy-list USER_POL1**
   host1(config-policy-list)#**classifier-group * external parent-group EPG1 parameter A**
   host1(config-policy-list-classifier-group)#**forward**
   host1(config-policy-list-classifier-group)#**exit**
   host1(config-policy-list)#**exit**

   Verify the policy list configuration.

```
host1#show policy-list USER_POL1
                                      Policy Table
                                      ------ -----
IP Policy USER_POL1
   Administrative state: enable
   Reference count:      0
   Classifier control list: *, precedence 100, external parent-group EPG1
parameter A
      forward
```

3. Create a policy list to attach to user 2. Also, create a rate limit to police voice traffic and another rate limit to police all traffic for user 2. Because voice traffic is preferred, it borrows the tokens unconditionally from all aggregate policers in the hierarchy.

```
host1(config)#rate-limit-profile VOICE_RATE two-rate hierarchical
host1(config-rate-limit-profile)#committed-rate 64000
host1(config-rate-limit-profile)#committed-action transmit unconditional
host1(config-rate-limit-profile)#exit

host1(config)#rate-limit-profile USER_RATE two-rate hierarchical
host1(config-rate-limit-profile)#committed-rate 300000
host1(config-rate-limit-profile)#committed-action transmit conditional
host1(config-rate-limit-profile)#exit

host1(config)#ip classifier-list VOICE_CLACL udp any any eq 10000

host1(config)#ip policy-list USER_POL2
host1(config-policy-list)#classifier-group VOICE_CLACL parent-group IPG1
host1(config-policy-list-classifier-group)#rate-limit-profile VOICE_RATE
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group * parent-group IPG1
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#parent-group IPG1 external parent-group EPG1
  parameter A
host1(config-policy-list-parent-group)#rate-limit-profile USER_RATE
host1(config-policy-list-parent-group)#exit
host1(config-policy-list)#exit
```

Verify the policy list configuration.

```
host1#show policy-list USER_POL1
                                      Policy Table
                                      ------ -----
IP Policy USER_POL2
   Administrative state: enable
   Reference count:      0
   Classifier control list: VOICE_CLACL, precedence 100, parent-group IPG1
      rate-limit-profile VOICE_RATE
   Classifier control list: *, precedence 100, parent-group IPG1
      forward
   Parent group: IPG1, external parent-group EPG1 parameter A
      rate-limit-profile USER_RATE
```

4. Create a policy list to attach to user 3 and mark Dsfield=4 traffic with a special traffic class.


```
host1(config)#ip classifier-list DSFIELD4_CLACL ip any any dsfield 4
host1(config)#ip policy-list USER_POL3
host1(config-policy-list)#classifier-group DSFIELD4_CLACL external parent-group
   EPG1 parameter A
host1(config-policy-list-classifier-group)#traffic-class LOW_DROP
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group * external parent-group EPG1
   parameter A
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#exit
```

The policies created earlier are attached statically to the user's corresponding entry interface in the E Series router. In this case, fast3/0.1 connects to user 1, fast3/0.2 connects to user 2, and fast3/0.3 connects to user 3.

5. Create the major interface.

```
host1(config)#interface fastEthernet 3/0
host1(config-interface)#encapsulation vlan
host1(config-interface)#exit
```

6. Create an interface for user 1, attach USER_POL1, and map parameter A to the VLAN interface stacked below the shared IP interface.

```
host1(config)#interface fastEthernet 3/0.1
host1(config-interface)#vlan id 1
host1(config-interface)#exit

host1(config)#interface ip 3/0.1.1
host1(config-interface)#ip policy-parameter hierarchical A vlan
host1(config-interface)#ip policy input USER_POL1 statistics enabled
host1(config-interface)#exit
```

7. Create the interface for user 2, attach USER_POL2, and map parameter A to the VLAN interface.

```
host1(config)#interface ip 3/0.1.2
host1(config-interface)#ip policy-parameter hierarchical A vlan
host1(config-interface)#ip policy input USER_POL2 statistics enabled
host1(config-interface)#exit
```

8. Create the interface for user 3, attach USER_POL3, and map parameter A to the VLAN interface.

```
host1(config)#interface ip 3/0.1.3
host1(config-interface)#ip policy-parameter hierarchical A vlan
host1(config-interface)#ip policy input USER_POL3 statistics enabled
host1(config-interface)#exit
```

9. For dynamic users, under each user's record in RADIUS, you can specify the ingress policy name. However, you can only specify the policy parameter through the profile.

```
host1(config)#profile PPPOE_PROF1
host1(config-profile)#ip policy-parameter hierarchical A vlan
host1(config-profile)#exit

host1(config)#interface fastEthernet 3/0.1
```

```
host1(config-interface)#vlan id 1
host1(config-interface)#encapsulation pppoe
host1(config-interface)#profile PPPOE_PROF1
host1(config-interface)#pppoe auto-configure
host1(config-interface)#exit
```

Related Documentation

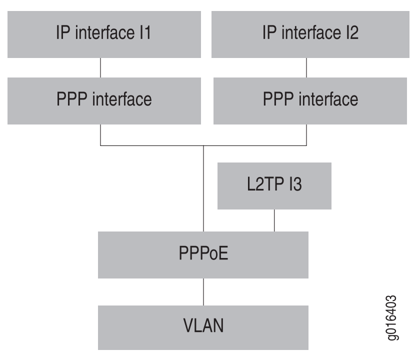
- Hierarchical Rate Limits Overview on page 65

Example: Wholesale L2TP Model Hierarchical Policy Configuration

In this example:

- There are two terminated subscribers and their corresponding IP interfaces are I1 & I2 in the E Series router.
- There is a single tunneled subscriber whose interface is I3.
- Interfaces I1 and I2 have dedicated 1 Mbps bandwidth each and interface I3 has dedicated 10 Mbps bandwidth. However, if interface I3 is not forwarding any traffic, then the allocated 10 Mbps can be shared by interfaces I1 and I2. Therefore, interfaces I1 and I2 can individually go up to a maximum of 11 Mbps if only one is actively sending traffic. If both interfaces are actively sending traffic, they can both get a maximum of 6 Mbps. However, any time interface I3 is actively sending traffic, it can forward up to the contracted 10 Mbps and interfaces I1 and I2 fall back to 1 Mbps.

Figure 10: Interface Stack for Wholesale L2TP Mode

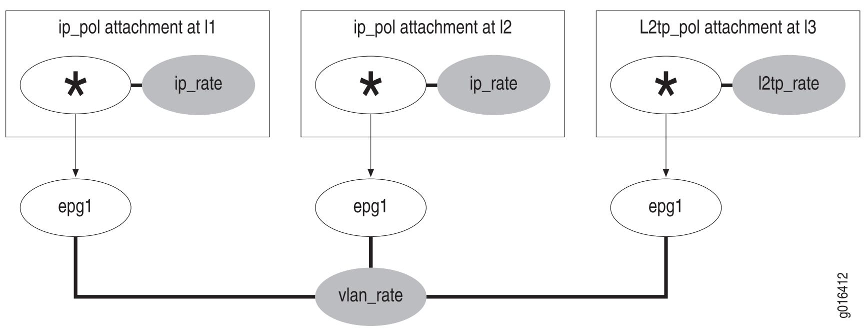


To use this example, you must configure the following:

- At interfaces I1 and I2:
  - IP_RATE, Committed Rate:1 Mbps
  - Peak Rate: 11 Mbps
  - Committed Action: transmit unconditional
  - Conformed Action: transmit conditional
  - Exceeded Action: drop
- At I3—L2TP_RATE:

  - Committed Rate: 10 Mbps
  - Peak Rate: 0 Mbps
  - Committed Action: transmit unconditional
  - Conformed Action: drop
  - Exceeded Action: drop
- Policers at I1, I2, and I3 feed into a single policer that has the following configuration:
  - VLAN_RATE, Committed Rate: 12 Mbps
  - Peak Rate: 0 Mbps
  - Committed Action: transmit final
  - Conformed Action: drop
  - Exceeded Action: drop
- IP policy USER_POL1 is attached as input to I1, IP policy USER_POL2 is attached as input to I2, and L2TP policy USER_POL3 is attached as input to I3.
- Policer instance VLAN_RATE is shared across all three instances of EPG1.

Figure 11: Wholesale L2TP Configuration



1. Create a rate-limit that can be shared across all forwarding interfaces. Create an external parent group to hold this rate limit.

   host1(config)#**rate-limit-profile VLAN_RATE two-rate hierarchical**
   host1(config-rate-limit-profile)#**committed-rate 12000000**
   host1(config-rate-limit-profile)#**committed-action transmit final**
   host1(config-rate-limit-profile)#**exit**

   host1(config)#**parent-group EPG1**
   host1(config-parent-group)#**rate-limit-profile VLAN_RATE**
   host1(config-parent-group)#**exit**

2. Create a policy list to attach to users 1 and 2.

   host1(config)#**rate-limit-profile IP_RATE two-rate hierarchical**
   host1(config-rate-limit-profile)#**committed-rate 1000000**
   host1(config-rate-limit-profile)#**committed-action transmit unconditional**
   host1(config-rate-limit-profile)#**peak-rate 11000000**

```
host1(config-rate-limit-profile)#conformed-action transmit conditional
host1(config-rate-limit-profile)#exit

host1(config)#policy-parameter A hierarchical
host1(config-policy-parameter)#exit
host1(config)#ip policy-list IP_POL
host1(config-policy-list)#classifier-group * external parent-group EPG1
  parameter A
host1(config-policy-list-classifier-group)#rate-limit-profile IP_RATE
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#exit
```

3. Create a policy list to attach to user 3.

```
host1(config)#rate-limit-profile L2TP_RATE two-rate hierarchical
host1(config-rate-limit-profile)#committed-rate 10000000
host1(config-rate-limit-profile)#committed-action transmit unconditional
host1(config-rate-limit-profile)#exit

host1(config)#l2tp policy-list L2TP_POL
host1(config-policy-list)#classifier-group * external parent-group EPG1
  parameter A
host1(config-policy-list-classifier-group)#rate-limit-profile L2TP_RATE
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#exit
```

4. In both terminated users' record in RADIUS, you must specify the ingress policy name IP_POL. You must specify the ingress policy name L2TP_POL in the tunneled user's record in RADIUS. However, be sure to specify the policy parameter through a profile.

```
host1(config)#profile PPPOE_PROF1
host1(config-profile)#ip policy-parameter hierarchical A 1
host1(config-profile)#l2tp policy-parameter hierarchical A 1
host1(config-profile)#exit

host1(config)#interface fastEthernet 3/0.1
host1(config-interface)#vlan id 1
host1(config-interface)#encapsulation pppoe
host1(config-interface)#profile PPPOE_PROF1
host1(config-interface)#pppoe auto-configure
host1(config-interface)#exit
```

Related Documentation

- Hierarchical Rate Limits Overview on page 65

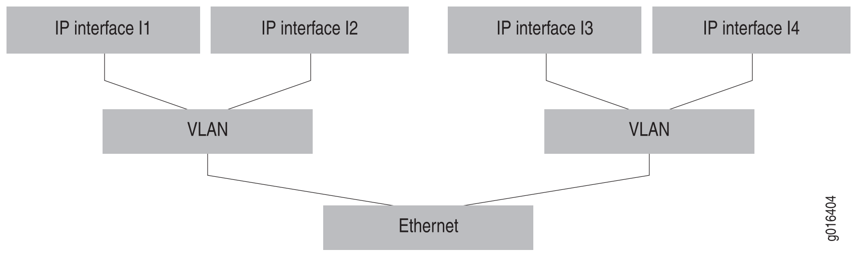
Example: Aggregate Rate Limit for All Nonvoice Traffic Hierarchical Policy Configuration

In this example:

- There are four IP sessions and their corresponding interfaces are I1, I2, I3, and I4.
- Each interface corresponds to a dynamic user.

- All users can send a maximum of 1 Mbps video traffic each, but the total bandwidth for all video traffic combined is 1.5 Mbps for a specific VLAN.
- Similarly, all users can send a maximum of 5 Mbps data traffic, but the sum of all data traffic on an Ethernet port is 10 Mbps. Interfaces I1-I4 are interfaces where you can attach policies.

Figure 12: Interface Stack for Aggregate Rate Limit

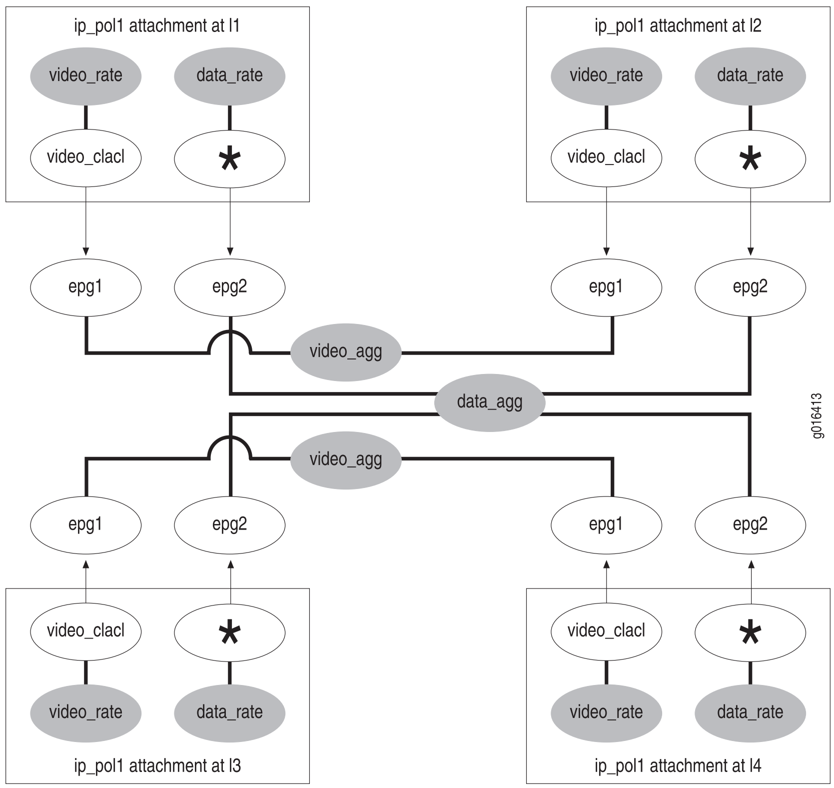


This example uses the following:

- At I1, I2, I3, I4:
  - Classified Video Flow. VIDEO_RATE, Committed Rate: 1 Mbps
  - Peak Rate: 0 Mbps
  - Committed Action: transmit conditional
  - Conformed Action: drop
  - Exceeded Action: drop
- At I1, I2, I3, I4:
  - Classified Data Flow. DATA_RATE, Committed Rate: 5 Mbps
  - Peak Rate: 0 Mbps
  - Committed Action: transmit conditional
  - Conformed Action: drop
  - Exceeded Action: drop
- All classified video flow policers over each VLAN interface feed into a single policer with the following configuration:
  - VIDEO_AGG, Committed Rate: 1.5 Mbps
  - Peak Rate: 0 Mbps
  - Committed Action: transmit final
  - Conformed Action: drop
  - Exceeded Action: drop

- All classified data flow policers over each Ethernet port feed into a single policer with the following configuration:
  - DATA_AGG, Committed Rate: 10 Mbps
  - Peak Rate: 0 Mbps
  - Committed Action: transmit final
  - Conformed Action: drop
  - Exceeded Action: drop
- Policy IP_POL1 is attached to I1, I2, I3, and I4

Figure 13: Aggregate Rate Limit for Nonvoice Traffic Configuration



1. Create a rate limit that can be shared across all video streams. Create an external parent group to hold this rate limit.

```
host1(config)#rate-limit-profile VIDEO_AGG two-rate hierarchical
host1(config-rate-limit-profile)#committed-rate 1500000
host1(config-rate-limit-profile)#committed-action transmit final
host1(config-rate-limit-profile)#exit

host1(config)#parent-group EPG1
host1(config-parent-group)#rate-limit-profile VIDEO_AGG
host1(config-parent-group)#exit
```


2. Create a policy list to attach to all IP sessions.

```
host1(config)#rate-limit-profile VIDEO_RATE two-rate hierarchical
host1(config-rate-limit-profile)#committed-rate 1000000
host1(config-rate-limit-profile)#committed-action transmit conditional
host1(config-rate-limit-profile)#exit

host1(config)#rate-limit-profile DATA_RATE two-rate hierarchical
host1(config-rate-limit-profile)#committed-rate 5000000
host1(config-rate-limit-profile)#committed-action transmit conditional
host1(config-rate-limit-profile)#exit

host1(config)#policy-parameter A hierarchical
host1(config-policy-parameter)#exit
host1(config)#policy-parameter B hierarchical
host1(config-policy-parameter)#exit

host1(config)#ip policy-list IP_POL1
host1(config-policy-list)#classifier-group VIDEO_CLACL external parent-group EPG1
  parameter A
host1(config-policy-list-classifier-group)#rate-limit-profile VIDEO_RATE
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group * external parent-group EPG2
  parameter B
host1(config-policy-list-classifier-group)#rate-limit-profile DATA_RATE
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#exit
```

3. In all users' records in RADIUS, specify the ingress policy name IP_POL1. However, be sure to specify the policy parameter through the profile.

```
host1(config)#profile PPPOE_PROF1
host1(config-profile)#ip policy-parameter hierarchical A vlan
host1(config-profile)#ip policy-parameter hierarchical B ethernet
host1(config-profile)#exit

host1(config)#interface fastEthernet 3/0.1
host1(config-interface)#vlan id 1
host1(config-interface)#encapsulation pppoe
host1(config-interface)#profile PPPOE_PROF1
host1(config-interface)#pppoe auto-configure
host1(config-interface)#exit
```

Related Documentation

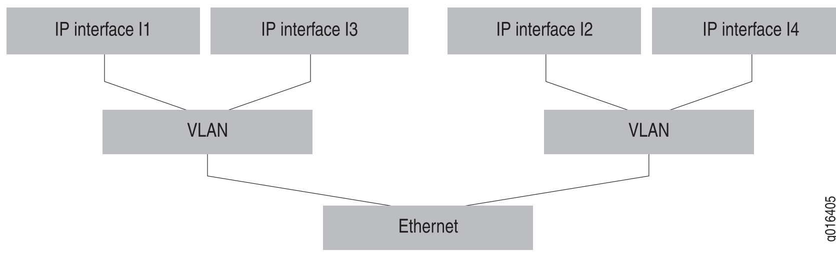
- Hierarchical Rate Limits Overview on page 65

Example: Arbitrary Interface Groups Hierarchical Policy Configuration

In this example, there are four terminated sessions and their corresponding IP interfaces are I1, I2, I3, and I4. Figure 14 on page 150 shows the following:

- Sessions I1 and I2 are for the same subscriber: I1 carries only voice traffic and I2 carries all other traffic for this subscriber
- Sessions I3 and I4 are for another subscriber.
- Voice traffic has a contracted minimum of 64 Kbps, but the combined voice and other traffic for each subscriber has a contracted maximum of 1 Mbps.
- Interfaces I1-I4 are interfaces where you can attach policies.

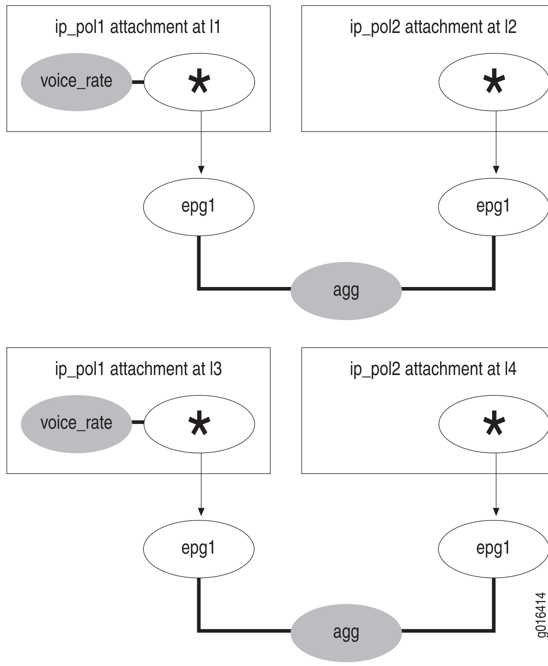
Figure 14: Interface Stack for Arbitrary Interface Groups



This example uses the following:

- At I1 and I3:
  - VOICE_RATE, Committed Rate: 64 Kbps
  - Peak Rate: 0 Mbps
  - Committed Action: transmit unconditional
  - Conformed Action: drop
  - Exceeded Action: drop
- At I2 and I4:
  - No policer configured
  - I1 and I2 feed into a single policer with the following configuration: AGG, Committed Rate: 1 Mbps, Peak Rate: 0 Mbps, Committed Action: transmit, Conformed Action: drop, Exceeded Action: drop

Figure 15: Arbitrary Interface Groups Configuration



1. Create an aggregate rate limit that can be shared across multiple interfaces. Create an external parent group to hold this rate limit.

```
host1(config)#rate-limit-profile AGG two-rate hierarchical
host1(config-rate-limit-profile)#committed-rate 1000000
host1(config-rate-limit-profile)#committed-action transmit final
host1(config-rate-limit-profile)#exit

host1(config)#parent-group EPG1
host1(config-parent-group)#rate-limit-profile AGG
host1(config-parent-group)#exit
```

2. Create a policy list to be attached to all voice sessions.

```
host1(config)#rate-limit-profile VOICE_RATE two-rate hierarchical
host1(config-rate-limit-profile)#committed-rate 64000
host1(config-rate-limit-profile)#committed-action transmit unconditional
host1(config-rate-limit-profile)#exit

host1(config)#policy-parameter A hierarchical
host1(config-policy-parameter)#exit

host1(config)#ip policy-list IP_POL1
host1(config-policy-list)#classifier-group * external parent-group EPG1
   parameter A
host1(config-policy-list-classifier-group)#rate-limit-profile VOICE_RATE
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#exit
```

3. Create a policy list to attach to all other sessions.


```
host1(config)#ip policy-list IP_POL2
host1(config-policy-list)#classifier-group * external parent-group EPG1
   parameter A
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#exit
```

4. Attach IP_POL1 to the voice session of first user and attach IP_POL2 to the other session for the same user. Specify the same ID for parameter A.

```
host1(config)#interface fastEthernet 3/0.1
host1(config-interface)#vlan id 1
host1(config-interface)#exit

host1(config)#interface ip 3/0.1.1
host1(config-interface)#ip policy-parameter hierarchical A 1
host1(config-interface)#ip policy input IP_POL1 statistics enable
host1(config-interface)#exit

host1(config)#interface fastEthernet 3/0.2
host1(config-interface)#vlan id 2
host1(config-interface)#exit

host1(config)#interface ip 3/0.2.1
host1(config-interface)#ip policy-parameter hierarchical A 1
host1(config-interface)#ip policy input IP_POL2 statistics enable
host1(config-interface)#exit
```

5. Attach IP_POL1 to the voice session of the second user and attach IP_POL2 to the other session for the same user. Specify a different ID for parameter A.

```
host1(config)#interface ip 3/0.1.2
host1(config-interface)#ip policy-parameter hierarchical A 2
host1(config-interface)#ip policy input IP_POL1 statistics enable
host1(config-interface)#exit

host1(config)#interface ip 3/0.2.2
host1(config-interface)#ip policy-parameter hierarchical A 2
host1(config-interface)#ip policy input IP_POL2 statistics enable
host1(config-interface)#exit
```

Related Documentation

- Hierarchical Rate Limits Overview on page 65

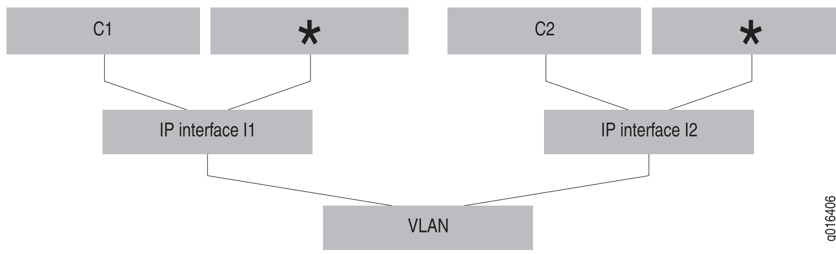
Example: Service and User Rate-Limit Hierarchy Overlap Hierarchical Policy Configuration

In the service and user rate-limit hierarchy overlap configuration example:

- The service provider has to enforce a bandwidth limit on a video service over a VLAN and wants to limit the maximum bandwidth of each user's total traffic.
- There are two terminated sessions and their corresponding IP interfaces are I1 and I2.

- Each session contains a video flow classified by C1 and all other traffic is classified by an asterisk (*).
- All video flows over the VLAN are rate-limited to a common rate of 1Mbps.
- Each session is individually rate-limited by 2 Mbps.
- You can attach policies at interface I1-I2.

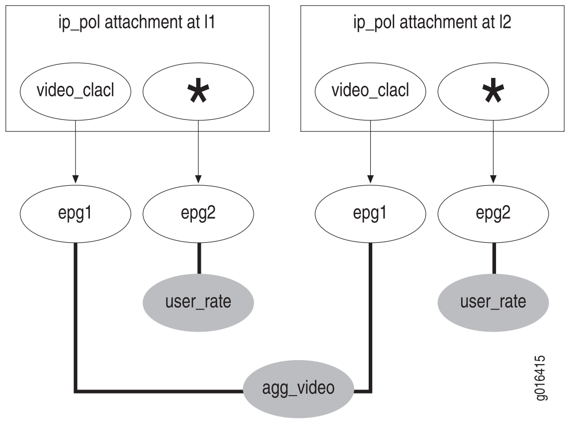
Figure 16: Interface Stack for Service and User Rate-Limit Hierarchy Overlap



This example uses the following:

- At I1 and I2:
  - USER_RATE, Committed Rate: 2 Mbps
  - Peak Rate: 0 Mbps
  - Committed Action: transmit final
  - Conformed Action: drop
  - Exceeded Action: drop
- Both C1 and C2 feed into a single policer with the following configuration:
  - AGG_VIDEO, Committed Rate: 1 Mbps
  - Peak Rate: 0 Mbps
  - Committed Action: transmit conditional
  - Conformed Action: drop
  - Exceeded Action: drop

Figure 17: Service and User Rate-Limit Hierarchy Overlap Configuration



1. Create an aggregate rate limit that can be applied to each IP session. Create an external parent group to hold this rate limit.

   host1(config)#**rate-limit-profile USER_RATE two-rate hierarchical**
   host1(config-rate-limit-profile)#**committed-rate 2000000**
   host1(config-rate-limit-profile)#**committed-action transmit final**
   host1(config-rate-limit-profile)#**exit**

   host1(config)#**parent-group EPG2**
   host1(config-parent-group)#**rate-limit-profile USER_RATE**
   host1(config-parent-group)#**exit**

2. Create an aggregate rate limit that can be shared across multiple video streams. Create an external parent group to hold this rate limit.

   host1(config)#**rate-limit-profile AGG_VIDEO two-rate hierarchical**
   host1(config-rate-limit-profile)#**committed-rate 1000000**
   host1(config-rate-limit-profile)#**committed-action transmit conditional**
   host1(config-rate-limit-profile)#**exit**

   host1(config)#**policy-parameter B hierarchical**
   host1(config-policy-parameter)#**exit**

   host1(config)#**parent-group EPG1**
   host1(config-parent-group)#**next-parent EPG2 parameter B**
   host1(config-parent-group)#**rate-limit-profile AGG_VIDEO**
   host1(config-parent-group)#**exit**

3. Create a policy list to be attached to each IP session.

   host1(config)#**ip classifier-list VIDEO_CLACL udp any any eq 4000**

   host1(config)#**policy-parameter A hierarchical**
   host1(config-policy-parameter)#**exit**

   host1(config)#**ip policy-list IP_POL**
   host1(config-policy-list)#**classifier-group VIDEO_CLACL external parent-group EPG1 parameter A**


```
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group * external parent-group EPG2
   parameter B
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#exit
```

4. Attach IP_POL to each IP session. Specify the same ID for parameter A, but a different ID for parameter B.

```
host1(config)#interface fastEthernet 3/0.1
host1(config-interface)#vlan id 1
host1(config-interface)#exit

host1(config)#interface ip 3/0.1.1
host1(config-interface)#ip policy-parameter hierarchical A vlan
host1(config-interface)#ip policy-parameter hierarchical B forwarding
host1(config-interface)#ip policy input IP_POL statistics enable
host1(config-interface)#exit

host1(config)#interface ip 3/0.1.2
host1(config-interface)#ip policy-parameter hierarchical A vlan
host1(config-interface)#ip policy-parameter hierarchical B forwarding
host1(config-interface)#ip policy input IP_POL statistics enable
host1(config-interface)#exit
```

Related Documentation

- Hierarchical Rate Limits Overview on page 65

Example: Percentage-Based Hierarchical Rate-Limit Profile for External Parent Group

The following example creates rate-limit profiles with rate based on percentage and burst in milliseconds. You can create a policy using these rate-limit profiles and attach them to different interfaces using different parameter values.

1. Create a policy parameter refRlpRate.

```
host1(config)#policy-parameter refRlpRate reference-rate
host1(config-policy-parameter)#reference-rate 100000
host1(config-policy-parameter)#exit
```

2. Create a hierarchical rate-limit profile rlpData.

```
host1(config)#rate-limit-profile rlpData hierarchical
host1(config-rate-limit-profile)#committed-rate refRlpRate percentage 10
host1(config-rate-limit-profile)#peak-rate refRlpRate percentage 100
host1(config-rate-limit-profile)#exit
```

3. Associate this hierarchical rate-limit profile with the external parent group.

```
host1(config)#parent-group epg1
host1(config-parent-group)#rate-limit-profile rlpData
host1(config-parent-group)#exit
```

4. Create a hierarchical policy parameter.

   host1(config)#**policy-parameter A hierarchical**
   host1(config-policy-parameter)#**aggregation-node forwarding**
   host1(config-policy-parameter)#**exit**

5. Create a policy that references the external parent group.

   host1(config)#**ip policy-list P**
   host1(config-policy)#**classifier-group data external parent-group epg1 parameter A**
   host1(config-policy-classifier-group)#**forward**
   host1(config-policy-classifier-group)#**exit**
   host1(config-policy)#**exit**

6. Attach an IP policy P at interface atm5/0.1 by specifying a different reference-rate value.

   host1(config)#**interface atm 5/0.1**
   host1(config-if)#**ip policy-parameter reference-rate refRlpRate 5000000**
   host1(config-if)#**ip policy input P stats enabled**

7. Attach an IP policy P at interface atm5/0.2 to use the default value.

   host1(config)#**interface atm 5/0.2**
   host1(config-if)#**ip policy input P stats enabled**

8. Display the policy list.

   host1#**show policy-list**

```
Policy Table
------ -----
IP Policy P
   Administrative state: enable
   Reference count:      2
   Classifier control list: cl1, precedence 100, external parent-group epg1
parameter A
      forward

   Referenced by interface(s):
      ATM5/0.1 input policy, statistics disabled, virtual-router default
      ATM5/0.2 input policy, statistics enabled, virtual-router default

   Referenced by profile(s):
      None

   Referenced by merge policies:
      None
```

9. Display the interface configuration to verify the reference rate (committed rate and peak rate) being different for both interface attachments (only the policy-related section of the output is shown).

   host1#**show ip interface atm 5/0.1**

```
  IP policy input P
    classifier-group data entry 1, external parent-group epg1 aggregate-node
forwarding
      0 packets, 0 bytes
      forward
    external parent-group epg1 aggregate-node forwarding
```



```
      rate-limit-profile rlpData
        committed rate: 500000 bps, committed burst: 8192 bytes (default)
        peak rate: 5000000 bps, peak burst: 62500 bytes (default)
        committed: 0 packets, 0 bytes, action: transmit conditional
        conformed: 0 packets, 0 bytes, action: transmit conditional
        exceeded:  0 packets, 0 bytes, action: drop
        unconditional: 0 packets, 0 bytes
        saturated: 0 packets, 0 bytes
```

`host1#`**show ip interface atm 5/0.2**

```
  IP policy input P
    classifier-group data entry 1, external parent-group epg1 aggregate-node
forwarding
      0 packets, 0 bytes
      forward
    external parent-group epg1 aggregate-node forwarding
      rate-limit-profile rlpData
        committed rate: 10000 bps, committed burst: 8192 bytes (default)
        peak rate: 100000 bps, peak burst: 8192 bytes (default)
        committed: 0 packets, 0 bytes, action: transmit conditional
        conformed: 0 packets, 0 bytes, action: transmit conditional
        exceeded:  0 packets, 0 bytes, action: drop
        unconditional: 0 packets, 0 bytes
        saturated: 0 packets, 0 bytes
```

Related Documentation

- External Parent Groups on page 132
- Hierarchical Rate Limits Overview on page 65

Example: PPP Interfaces Hierarchical Policy Configuration

Figure 18 on page 159 shows the following example:

- Two dual-stack PPP user sessions are configured over the same VLAN interface.
- Each PPP session corresponds to a dynamic user.
- Each PPP user can send a combined IPv4 and IPv6 video traffic with a maximum rate of 1 Mbps.
- One of the subscribers, PPP user1, can send a default maximum rate of 3 Mbps combined IPv4/IPv6 session traffic, whereas the other subscriber, PPP user2, can send a default maximum rate of 5 Mbps combined IPv4/IPv6 session traffic.
- I1 and I3 are interfaces where IPv4 policies are attached.
- I2 and I4 are interfaces where IPv6 policies are attached.
- PPP 1 is the PPP session of dynamic subscriber 1. I1 and I2 belong to this subscriber.
- PPP 2 is the PPP session of dynamic subscriber 2. I3 and I4 belong to this subscriber.
- vlan denotes the VLAN interface over which subscribers are configured.

This example uses the following:

- At I1 and I3:

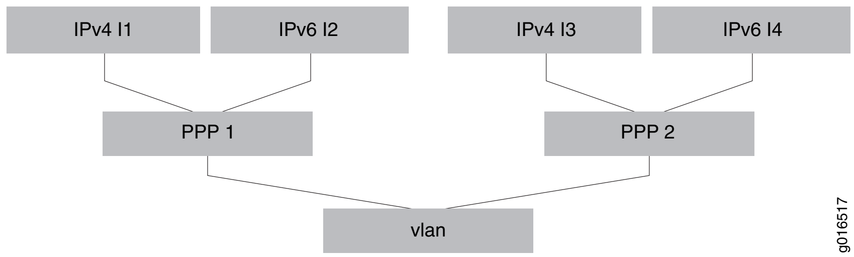
  - Classified Video Flow. VIDEO_RATE, Committed Rate: 1 Mbps
  - Peak Rate: 0 Mbps
  - Committed Action: transmit unconditional
  - Conformed Action: drop
  - Exceeded Action: drop
- At I2 and I4:
  - Classified Video Flow. VIDEO_RATE, Committed Rate: 1 Mbps
  - Peak Rate: 0 Mbps
  - Committed Action: transmit unconditional
  - Conformed Action: drop
  - Exceeded Action: drop

All classified video flow policers over each PPP interface feed into a single policer with the following configuration:

- At I1 and I3:
  - All Flows. SESSION_RATE, Committed Rate: 3 Mbps
  - Peak Rate: 0 Mbps
  - Committed Action: transmit conditional
  - Conformed Action: drop
  - Exceeded Action: drop
  - At I2 and I4:
    - All Flows. SESSION_RATE, Committed Rate: 5Mbps
    - Peak Rate: 0 Mbps
    - Committed Action: transmit conditional

- Conformed Action: drop
- Exceeded Action: drop

Figure 18: Interface Stack for Hierarchical Policy Configuration



1. Create a hierarchical policy parameter list for PPP interfaces.

   host1(config)#**policy-parameter P1_PPP hierarchical**
   host1(config-policy-parameter)#**exit**

2. Create a reference rate parameter to be used in external parent groups associated with PPP sessions.

   host1(config)#**policy-parameter sessionRlpRate reference-rate**
   host1(config-policy-parameter)#**reference-rate 3000000**
   host1(config-policy-parameter)#**exit**

3. Create an aggregate session rate-limit (using reference-rate) that can be shared between IPv4 and IPv6 interfaces of each PPP session. Create an external parent group to hold this rate-limit.

   host(config)#**rate-limit-profile SESSION_AGG two-rate hierarchical**
   host(config-rate-limit-profile)#**committed-rate sessionRlpRate percent 100**
   host(config-rate-limit-profile)#**committed-action transmit conditional**
   host(config-rate-limit-profile)#**exit**

   host(config)#**parent-group EPG_SESSION**
   host(config-parent-group)#**rate-limit-profile SESSION_AGG**
   host(config-parent-group)#**exit**

4. Create an aggregate video rate-limit that can be shared between IPv4 and IPv6 interfaces of each PPP session. Create an external parent group to hold this rate-limit.

   host(config)#**rate-limit-profile VIDEO_AGG two-rate hierarchical**
   host(config-rate-limit-profile)#**committed-rate 1000000**
   host(config-rate-limit-profile)#**committed-action transmit unconditional**
   host(config-rate-limit-profile)#**exit**

   host(config)#**parent-group EPG_VIDEO**
   host1(config-parent-group)#**next-parent EPG_SESSION parameter P1_PPP**
   host(config-parent-group)#**rate-limit-profile VIDEO_AGG**
   host(config-parent-group)#**exit**

5. Create IPv4/ IPv6 policy-lists to be attached to all PPP sessions.

   host(config)#**ip policy-list IP_POL1**

```
host(config-policy-list)#classifier-group VIDEO_CLACL_V4 external parent-group
   EPG_VIDEO parameter P1_PPP
host(config-policy-list-classifier-group)#forward
host(config-policy-list-classifier-group)#exit
host(config-policy-list)#classifier-group * external parent-group EPG_SESSION
   parameter P1_PPP
host(config-policy-list-classifier-group)#forward
host(config-policy-list-classifier-group)#exit
host(config-policy-list)#exit

host(config)#ipv6 policy-list IP_POL2
host(config-policy-list)#classifier-group VIDEO_CLACL_V6 external parent-group
   EPG_VIDEO parameter P1_PPP
host(config-policy-list-classifier-group)#forward
host(config-policy-list-classifier-group)#exit
host(config-policy-list)#classifier-group * external parent-group EPG_SESSION
   parameter P1_PPP
host(config-policy-list-classifier-group)#forward
host(config-policy-list-classifier-group)#exit
host(config-policy-list)#exit
```

6. Specify the policy parameter and attachments through the profile.

```
host(config)#profile PPPOE_PROF1
host(config-profile)#ip policy-parameter hierarchical P1_PPP ppp-interface
host(config-profile)#ip policy input IP_POL1 sta enabled merge
host(config-profile)#ipv6 policy-parameter hierarchical P1_PPP ppp-interface
host(config-profile)#ipv6 policy input IP_POL2 sta enabled merge
host(config-profile)#exit
```

7. Use another profile for the second subscriber to specify a different session rate of 5 Mbps by overriding the default rate of 3 Mbps , and specify the policy parameter through the profile.

```
host(config)#profile PPPOE_PROF2
host(config-profile)#ip policy-parameter hierarchical P1_PPP ppp-interface
host(config-profile)#ip policy input IP_POL1 sta enabled merge
host(config-profile)#ipv6 policy-parameter hierarchical P1_PPP ppp-interface
host(config-profile)#ipv6 policy input IP_POL2 sta enabled merge
host(config-profile)#ip policy-parameter reference-rate sessionRlpRate 5000000
host(config-profile)#ipv6 policy-parameter reference-rate sessionRlpRate 5000000
host(config-profile)#exit
```

8. Configure the VLAN, PPPoE, and PPP encapsulation and specify unique profile to use for each PPP user.

```
host(config)#interface gig 2/1/0.1
host(config-interface)#vlan id 1
host(config-interface)#encapsulation pppoe
host(config)#interface gig 2/1/0.1.1
host(config-interface)#encapsulation ppp
host(config-interface)#profile PPPOE_PROF1
host(config)#interface gig 2/1/0.1.2
host(config-interface)#encapsulation ppp
host(config-interface)#profile PPPOE_PROF2
host(config-interface)#exit
```


NOTE: You can optionally specify an aggregation node for a policy parameter during its definition in Global Configuration mode. This example configures parameters at the interface level. However, you can also configure these same settings at the global level, without defining them at the interface or profile level. This feature supports both these methods of configurations.

Related Documentation

- Hierarchical Rate Limits Overview on page 65

CHAPTER 8

Policy Resources

This chapter provides information about configuring policy resources. The chapter discusses the following topics:

Policy Resources Overview

The maximum number of policies that you can attach to interfaces on an E Series router depends on the classifier entries that make up the policy and the number of attachment resources available on the interface. JunosE Software allocates interface attachment resources when you attach policies to interfaces. See "Interface Attachment Resources Overview" on page 184 for information about attachment resources.

An E Series router supports software and hardware classifiers. A policy can be made up of any combination of software and hardware classifiers. You use the **classifier-list** command to configure all classifiers.

There are two categories of hardware classifiers, depending on the type of line module being used. OC48/STM16, GE-2, GE-HDE, ES2 4G, ES2 10G, and ES2 10G Uplink line modules support content-addressable memory (CAM) hardware classifiers—all other line modules support FPGA hardware classifiers. Table 17 on page 164 lists the classifiers supported on OC48/STM16, GE-2, GE-HDE, ES2 4G, ES2 10G, and ES2 10G Uplink line modules; Table 18 on page 165 lists the classifiers supported on all other line modules.

Table 17: Classifier Support (OC48/STM16, GE-2, GE-HDE, ES2 4G, ES2 10G, and ES2 10G Uplink Line Modules)

Interface Type	Hardware Classifier	Software Classifier
All interface types (except IP and IPv6)	–	• Color • Traffic class • User packet class
Frame Relay	Not supported	• DE bit
GRE tunnels	Not supported	• ToS
IP	• Color • Destination address • Destination port • Destination route class • ICMP type and code • IGMP type • IP flags • IP fragmentation • Local • Protocol • Source address • Source port • Source route class • TCP flags • ToS • Traffic class • User packet class	Not supported

Table 17: Classifier Support (OC48/STM16, GE-2, GE-HDE, ES2 4G, ES2 10G, and ES2 10G Uplink Line Modules) *(continued)*

Interface Type	Hardware Classifier	Software Classifier
IPv6	• Color • Destination address • Destination port • Destination route class • ICMPv6 type and code • Local • Protocol • Source address • Source port • Source route class • TC flags • TCP flags • Traffic class • User packet class	Not supported
MPLS	Not supported	• EXP
VLAN	Not supported	• User priority

Table 18: Classifier Support (All Line Modules Except OC48/STM16, GE-2, GE-HDE, ES2 4G, ES2 10G, and ES2 10G Uplink)

Interface Type	Hardware Classifier	Software Classifier
All interface types	–	• Color • Traffic class • User packet class
Frame Relay	Not supported	• DE bit
GRE tunnels	Not supported	• ToS
IP	• Destination address • Destination port • ICMP type and code • IGMP type • Protocol • Source address • Source port	• Destination route class • IP flags • IP fragmentation • Local • Source route class • TCP flags • ToS

Table 18: Classifier Support (All Line Modules Except OC48/STM16, GE-2, GE-HDE, ES2 4G, ES2 10G, and ES2 10G Uplink) *(continued)*

Interface Type	Hardware Classifier	Software Classifier
IPv6	• Destination address • Destination port • ICMPv6 type and code • Protocol • Source address • Source port	• Destination route class • Local • Source route class • TC field • TCP flags
MPLS	Not supported	• EXP
VLAN	Not supported	• User priority

Related Documentation

- CAM Hardware Classifiers and Interface Attachment Resources on page 184
- FPGA Hardware Classifiers on page 166
- Interface Attachment Resources Overview on page 184

FPGA Hardware Classifiers

Classification is the process of taking a single data stream in and sorting it into multiple output substreams. The classifier engine on an E Series router is a combination of PowerPC processors, working with an FPGA for a hardware assist.

In the Differentiated Services (DiffServ) architecture, two basic types of classifiers exist. The first classifier type is a multifield (MF) classifier. The MF classifier can examine multiple fields in the IP datagram header to determine the service class to which a packet belongs.

FPGA hardware classifiers are supported on all line modules except the OC48/STM16, GE-2, and GE-HDE line modules. "Policy Resources Overview" on page 163 lists the FPGA classifiers and software classifiers supported for each interface type.

An E Series router supports two versions of policies that are based on FPGA hardware classifiers. One version has a maximum of 16 classifier entries per policy, and the second version has 17 to 32 classifier entries per policy. The line module supports 16,255 policies when all policies have 16 hardware classifier entries or fewer, and supports 8127 policies when all policies have 17 to 32 hardware classifier entries.

You can configure a combination of the two versions of FPGA hardware classifier-based policies—you can have some that contain 16 or fewer classifier entries and others with more than 16 entries. In this case, between 8127 and 16,255 policies are supported, depending on the actual configuration.

You can also configure hardware classifier-based policies that have more than 32 classifier entries. The router groups the classifiers into blocks of 32. For example, if you configure a policy with 100 classifier entries, the router groups these as 3 policies that have 32 classifier entries and 1 policy with 4 classifier entries. The group with 4 classifier entries actually consumes 16 classifier resources, which is the minimum number consumed for a group in a mixed-mode hardware classifier configuration.

Unlike policies that are based on software classifiers, policies that are based on FPGA hardware classifiers consume resources at a rate of one resource per policy, regardless of the number of different hardware classifier categories in the policy. For example, if a classifier list has three hardware classifiers, such as destination address, source address, and protocol, the policy referencing that classifier list consumes only a single hardware classifier resource.

The same is true when multiple policy rules reference the classifier list. For example, if four policy rules reference the same classifier list (which contains three hardware classifiers), then still only one classifier entry is consumed.

Related Documentation

- Interface Attachment Resources Overview on page 184
- Policy Resources Overview on page 163

CAM Hardware Classifiers Overview

Content-addressable memory (CAM) hardware classifiers are supported on the OC48/STM16, GE-2, ES2 4G, ES2 10G, ES2 10G Uplink, and GE-HDE line modules."Policy Resources Overview" on page 163 lists CAM hardware classifiers and the software classifiers supported for each interface type.

The OC48/STM16 line module supports 128,000 144-bit CAM entries, and the GE-2 and GE-HDE line modules support 64,000 144-bit CAM entries. The ES2 4G LMs on E120 and E320 routers support 256,000 144-bit CAM entries, and the ES2 10G and ES2 10G Uplink LMs on E120 and E320 routers support 128,000 144-bit CAM entries. For most configurations, each classifier entry in a policy consumes one CAM entry. However, a policy that has only the default classifier consumes no CAM resources.

In this example, the policy consumes a total of four CAM entries: two entries for clacl1, one for clacl2, and one for the default classifier.

```
host1(config)#ip classifier-list clacl1 ip host 192.168.1.1 host 192.168.2.2 tos 1
host1(config)#ip classifier-list clacl1 ip host 192.168.1.1 host 192.168.2.2 tos 2
host1(config)#ip classifier-list clacl2 tcp any any tcp-flags "SYN"
host1(config)#ip policy-list policy1
host1(config-policy-list)#classifier-group clacl1
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group clacl2
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group *
host1(config-policy-list-classifier-group)#filter
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#exit
```

A single classifier entry consumes more than one CAM entry when:

- A classifier entry contains a port range. For example:

  host1(config)#**ip classifier-list clacl3 tcp any any range 5 8**

- A classifier entry contains the **not** keyword. Although this keyword is supported for IP classifier lists, we recommend that you not use it—you can usually achieve the desired behavior without this keyword.

  host1(config)#**ip classifier-list clacl4 ip not host 1.1.1.1 any**

In these cases, the actual number of entries that are consumed depends on the configuration.

Related Documentation

- CAM Hardware Classifiers and Interface Attachment Resources on page 184
- Interface Attachment Resources Overview on page 184
- Policy Resources Overview on page 163
- Size Limit for IP and IPv6 CAM Hardware Classifiers on page 168

Size Limit for IP and IPv6 CAM Hardware Classifiers

In JunosE Release 10.1.x and lower-numbered releases, the maximum width of a CAM hardware classifier entry for IPv4 or IPv6 in a single policy was 128 bits. This limitation enabled only 128 bits of classification data to be supported per policy. Any policy configuration (sum of all CLACL entries) with more than 128 bits of classification data failed when a policy was attached to an interface. This 128-bit size limitation applied to both IPv4 and IPv6 classification data. Although this limitation was acceptable for IPv4 classification, it posed problems when full IPv6 classification was required to be performed. In JunosE Release 10.2.x and later, based on the size limit for a combined IPv6 classifier entry, a maximum of 336 bits of CAM entry is supported for full IPv6 classification.

Some independent classifiers share the same classifier entry location, while others are combined together to form a larger classifier field. The smallest IPv6 classifier can consume 8 bits and the largest IPv6 classifier can consume 336 bits. Beginning with JunosE Release 10.2.x, variable-sized CAM entries are supported for IPv6 policies to avoid wasteful use of CAM entries. In earlier releases, the number of CAM entries per line module was predefined because all CAM entries were of a fixed size of 128 bits. With the support for variable-sized CAM entries for IPv6 policies, a dynamic algorithm is used for CAM resource management. This feature is supported on GE-2 and GE-HDE line modules on ERX14xx models, ERX7xx models, and the ERX310 router and ES2 4G LMs on E120 and E320 routers.

NOTE: OC48/STM16 line modules on ERX14xx models, ERX7xx models, and the ERX310 router support only 128-bit IPv6 classification.

Based on the size limit for a combined IPv6 classifier entry, a maximum of 336 bits of CAM entry is supported for full IPv6 classification. An additional 16 bits that are reserved for rule set ID are added to the total classifier entry size, which causes the total CAM entry size required to be 352 bits. Some of the mutually exclusive classification fields share the same classifier entry location, while a few other smaller fields are combined to form a single larger classifier field.

NOTE: Range vector hardware classifiers on line modules supported full IPv6 classification even in JunosE releases earlier than Release 10.2.x.

IP Classifiers and Size Limits

Table 19 on page 169 lists all IP classifiers and the size limit of each classifier entry.

Table 19: Size Limit of Individual IP Classifiers

IP Classifier	Size Limit (Bits)
Color	2
Destination address	32
Destination port	16
Destination route class	8
ICMP type	8
ICMP code	8
IGMP type	8
IP flags	3
IP fragmentation	2
Local	1
Protocol	8
Source address	32
Source port	16
Source route class	8
TCP flags	6
ToS	8
Traffic class	3
User packet class	4

Table 20 on page 170 lists the IP classifiers that share the same classifier entry location and those that are combined to form a larger classifier field. The table also lists the rules that apply to these types of classifier combinations.

The format in the classifier entry combinations in Table 20 on page 170 is based on the conventions for CLI commands, except that the pipe symbol (|) represents a choice of one or both options to the left and right of the pipe symbol.

Table 20: Size Limit of Combined IP Classifiers

IP Classifier Entry Combination	Size Limit (Bits)	Rule
Color or TCP flags or both	8	When you specify one or both of the color and TCP flags classifiers, 8 bits are added to the total classifier entry size.
Destination address	32	–
Destination address route class	8	–
[Destination port] and [[ICMP type] | [ICMP code] | [IGMP type] or nil]	16	The ICMP type, ICMP code, IGMP type, and destination port classifiers share the same classifier field location. When you specify the destination port classifier, 16 bits are added to the total classifier entry size. If you also specify the ICMP type, ICMP code, and IGMP type classifier, no additional bits are added.
[IP flags] | [IP fragmentation] | [Traffic class]	8	When you specify one or more of the IP flags, traffic class, and IP fragmentation classifiers, 8 bits are added to the total classifier entry size.
Protocol	8	–
[Source port] and [[ICMP type] | [ICMP code] | [IGMP type]]	16	The ICMP type, ICMP code, IGMP type, and source port classifiers share the same classifier field location. When you specify the source port classifier, 16 bits are added to the total classifier entry size. When you also specify the ICMP type, ICMP code, and IGMP type classifiers, no additional bits are added.
Source address	32	–
[not Source port] and [not Destination port] and [[ICMP type] | [ICMP code] | [IGMP type]]	16	When you do not specify the source port and destination port classifiers, but you specify one or more of ICMP type, ICMP code, and IGMP type, 16 bits are added to the total classifier entry size. ICMP type, ICMP code, and IGMP type require 16 bits even if the source port and destination port classifications are not configured.

Table 20: Size Limit of Combined IP Classifiers *(continued)*

IP Classifier Entry Combination	Size Limit (Bits)	Rule
ToS	8	–
User packet class or local or both	8	When you specify one or both of the user packet class and local classifiers, 8 bits are added to the total classifier entry size.

IPv6 Classifiers and Size Limits

Table 21 on page 171 lists all IPv6 and the size limit of each classifier entry.

Table 21: Size Limit of Individual IPv6 Classifiers

IPv6 Classifier Entry	Size Limit (Bits)
Color	2
Destination address	128
Destination port	16
Destination route class	8
ICMPv6 type	8
ICMPv6 code	8
Local	1
Protocol	8
Source address	128
Source port	16
Source route class	8
TC field	8
TCP Flags	6
Traffic class	3
User packet class	4

Table 22 on page 172 lists the IPv6 classifiers that share the same classifier entry location and those that are combined to form a larger classifier field. The table also lists the rules that apply to these types of classifier combinations.

The format in the classifier entry combinations in Table 22 on page 172 is based on the conventions for CLI commands, except that the pipe symbol (|) represents a choice of one or both options to the left and right of the pipe symbol.

Table 22: Size Limit of Combined IPv6 Classifiers

IPv6 Classifier Entry Combination	Size Limit (Bits)	Rule
Color or TCP flags or both	8	When you specify the color and/or TCP flags classifiers, 8 bits are added to the total classifier entry size.
Destination address (first word)	32	–
Destination address (second word)	32	–
Destination address (third word)	32	–
Destination address (fourth word)	32	–
Destination address route class	8	–
[Destination port] and [[ICMPv6 type] | [ICMPv6 code or nil]]	16	When you specify the destination port classifier, 16 bits are added to the total classifier entry size. If you also specify the ICMPv6 type and ICMPv6 code classifiers, no additional bits are added to the total classifier entry size.
[No source port] and [no destination port] and [[ICMPv6 type] | [ICMPv6 code]]	16	When you do not specify the source port and destination port classifiers, and you have already specified one or more of the ICMPv6 Type and ICMPv6 code classifiers, 16 bits are added to the total classifier entry size. The ICMPv6 type and ICMPv6 code classifiers require 16 bits even if you have not specified the source port and destination port classifiers.
Protocol	8	–
Source address (first word)	32	–
Source address (second word)	32	–
Source address (third word)	32	–
Source address (fourth word)	32	–

Table 22: Size Limit of Combined IPv6 Classifiers *(continued)*

IPv6 Classifier Entry Combination	Size Limit (Bits)	Rule
Source address route class	8	–
[source port] and [[ICMPv6 type] | [ICMPv6 code]]	16	When you specify the source port classifier, 16 bits are added to the total classifier entry size. If you also specify the ICMPv6 type and ICMPv6 code classifiers, no additional bits are added.
TC field	8	–
[User packet class] | [traffic class] | [local]	8	When you specify one or more of the user packet class, traffic class, and local classifiers, 8 bits are added to the total classifier entry size.

Related Documentation

- CAM Hardware Classifiers and Interface Attachment Resources on page 184
- CAM Hardware Classifiers Overview on page 167

Creating and Attaching a Policy with IP Classifiers

In this example, a policy with a combination of IP classifiers is created and attached. The configuration conforms to the 128 bit limit.

1. Match all TCP SYN packets from 1.1.1.1 to any DA with port 2000.

   ```
   host1(config)#ip classifier-list tcpCLACL tcp host 1.1.1.1 any eq 2000 tcp-flags "SYN"
   ```

2. Match all IP packets with the don't fragment flag set to host 2.2.2.2.

   ```
   host1(config)#ip classifier-list ipCLACL ip any host 2.2.2.2 ip-flags "dont-fragment"
   ```

3. Match all ICMP echo packets.

   ```
   host1(config)#ip classifier-list icmpCLACL icmp any any 8 0
   ```

4. Match all frames with the color red.

   ```
   host1(config)#ip classifier-list colorCLACL color red ip any any
   ```

5. Create a policy list.

   ```
   host1(config)#ip policy-list ipPol
   host1(config-policy-list)#classifier-group colorCLACL
   host1(config-policy-list-classifier-group)#filter
   host1(config-policy-list-classifier-group)#classifier-group tcpCLACL
   host1(config-policy-list-classifier-group)#filter
   host1(config-policy-list-classifier-group)#classifier-group icmpCLACL
   host1(config-policy-list-classifier-group)#filter
   host1(config-policy-list-classifier-group)#classifier-group ipCLACL
   host1(config-policy-list-classifier-group)#filter
   ```

6. Apply the policy list to an interface.

host1(config)#interface atm 5/0/0.1
host1(config-if)#ip policy input ipPol

Table 23 on page 174 lists the active classifiers in the policy named ipPol and the size of each classifier.

Table 23: Classification Fields for Example 1

Classifiers	Size (Bits)
Source address	32
Destination address	32
Destination port, ICMP type, ICMP code	16
Protocol	8
Color and TCP flags	8
TOS	8
IP flags	8

The total value of the classifiers requested in the ipPol policy is 112, which is less than 128 bit CAM entry size limit.

In this example, a policy with a combination of IP classifiers is created and attached. The configuration exceeds the 128 bit limit.

1. Match all TCP packets from 1.1.1.1 port 10 to 2.2.2.2 port 20.

   host1(config)#ip classifier-list tcpCLACL tcp host 1.1.1.1 eq 10 host 2.2.2.2 eq 20

2. Match all IP fragmentation offset equal to 1.

   host1(config)#ip classifier-list ipFragCLACL ip any any ip-frag-offset eq 1

3. Match all frames with the color red.

   host1(config)#ip classifier-list colorCLACL color red traffic-class best-effort ip any any

4. Match all frames with UPC 1.

   host1(config)#ip classifier-group upcCLACL user-packet-class 1 ip any any

5. Create a policy list.

   host1(config)#ip policy-list ipPol
   host1(config-policy-list)#classifier-group colorCLACL
   host1(config-policy-list-classifier-group)#filter
   host1(config-policy-list-classifier-group)#classifier-group ipFragCLACL
   host1(config-policy-list-classifier-group)#filter
   host1(config-policy-list-classifier-group)#classifier-group igmpCLACL
   host1(config-policy-list-classifier-group)#forward
   host1(config-policy-list-classifier-group)#classifier-group lowDelayCLACL
   host1(config-policy-list-classifier-group)#traffic-class strict-priority

```
host1(config-policy-list-classifier-group)#classifier-group tcpCLACL
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#classifier-group *
host1(config-policy-list-classifier-group)#filter
```

6. Apply the policy list to an interface.

```
host1(config)#interface atm 5/0/0.1
host1(config-if)#ip policy input ipPol
% too many classifier fields in policy
```

Table 24 on page 175 lists the active classifiers in the policy named ipPol and the size of each classifier.

Table 24: Classification Fields for Example 2

Classifiers	Size (Bits)
Source address	32
Source port	16
Destination port	16
Protocol	8
User packet class	8
Color	8
IP fragmentation	8
ToS	8

The configuration fails because the total value of the classifiers requested in the ipPol policy is 136, which is greater than 128 bit CAM entry size limit.

Related Documentation

- Interface Attachment Resources Overview on page 184

Examples: Variable-Sized CAM Classification for IPv6 Policies

Variable-sized CAM entries are supported for IPv6 policies to avoid wasting memory space. For example, if the classifier entries in a policy consume a 576-bit CAM entry when a 144-bit CAM entry is sufficient to store the classifier, over 400 bits of CAM memory are wasted. CAM memory is divided into blocks at the hardware level. Each CAM block can support 8000 144-bit, 4000 288-bit, or 2000 576-bit CAM entries. Based on the IPv6 header CAM entry size calculation, the minimum entry size required for IPv6 classification is 8 bits and the maximum entry size required is 336 bits.

Policy Manager calculates the CAM bit size and configures the CAM entries on the line modules. The size of the CAM entry is determined using the limits defined for each of the

IP classifier entry combination. In earlier releases, any policy configuration with CAM entries that exceeded the 128-bit limitation failed to be attached to the interface because it was not allowed by Policy Manager.

Beginning with JunosE Release 10.2.x, the IPv6 classification functionality is modified to classify traffic on more than 128 bits. To improve scalability for IPv6 policies, Policy Manager uses the optimum CAM entry size, depending on the IPv6 policy definition. The policy definition of IPv6 is used to determine which classification fields in the combined IPv6 classifier are present and the CAM entry length is computed dynamically. The following three different kinds of results are possible for an IPv6 policy:

- Sum of all classifier fields is less than or equal to 128 bits
- Sum of all classifier fields is between 128 bits and 272 bits
- Sum of all classifier fields is between 272 bits and 336 bits

CAM hardware classifiers support four types of CAM entries—72-bit, 144-bit, 288-bit, and 576-bits (16-bits are reserved for rule set id). Each of the policies fit into one of these four CAM entry types. The 72-bit CAM entry is not chosen as CAM devices on some line modules do not support this size limit. Therefore, the 144-bit, 288-bit, and 576-bit CAM entries are used as the variable-length CAM entries for IPv6 policies.

The following sections describe examples for each type of variable length IPv6 classification and the number of CAM entries for each case:

144-bit IPv6 Classification Example

In this example, a policy with a combination of IPv6 classifiers is created and attached. The configuration conforms to the 144 bit limit.

1. Match all TCP SYN packets from 1:1:: to any DA with port 2000.

   host1(config)#**ipv6 classifier-list tcpCLACL source-address 1:1::/32 tcp destination-port eq 2000 tcp-flags "SYN"**

2. Match all IPv6 packets to net 2:2::.

   host1(config)#**ipv6 classifier-list ipv6CLACL destination-address 2:2::/32**

3. Match all ICMPv6 echo packets.

   host1(config)#**ipv6 classifier-list icmpv6CLACL icmpv6 icmp-type 8 icmp-code 0**

4. Match all frames with the color red.

   host1(config)#**ipv6 classifier-list colorCLACL color red**

5. Create an IPv6 policy list.

   host1(config)#**ipv6 policy-list ipv6Pol**
   host1(config-policy-list)#**classifier-group colorCLACL**
   host1(config-policy-list-classifier-group)#**filter**
   host1(config-policy-list-classifier-group)#**classifier-group tcpCLACL**
   host1(config-policy-list-classifier-group)#**filter**
   host1(config-policy-list-classifier-group)#**classifier-group icmpv6CLACL**
   host1(config-policy-list-classifier-group)#**filter**
   host1(config-policy-list-classifier-group)#**classifier-group ipv6CLACL**


```
host1(config-policy-list-classifier-group)#filter
```

The policy ipv6Pol is requesting classification on Source Address (first word), Destination Address (first word), Destination Port, Protocol, TCP Flags, ICMPv6 Type, ICMPv6 Code, Color, and TC field. Table 25 on page 177 lists the active classifiers in the policy named ipv6Pol and the size of each classifier.

Table 25: IPv6 Classification Fields for a 144-bit CAM Entry

Classifiers	Size (Bits)
Source address (first word)	32
Destination address (first word)	32
Destination port, ICMPv6 type, ICMPv6 code	16
Protocol	8
Color and TCP flags	8
TC field	8

The sum of all classification fields requested in ipv6Pol is 104. This size causes Policy Manager to use 144-bit CAM entry for every classifier in this policy. One CAM entry is needed for each classifier in the policy and therefore, four 144-bit CAM entries are needed in all.

288-bit IPv6 Classification Example

The following example creates and attaches a policy, which requests classification on a single host address and TCP. The configuration exceeds the 128 bit limit.

1. Match all TCP packets from host 1:1:1:1:1:1:1:1 to any DA

   ```
   host1(config)#ipv6 classifier-list sourceCLACL source-address 1:1:1:1:1:1:1:1/128 tcp
   ```

2. Create an IPv6 policy list.

   ```
   host1(config)#ipv6 policy-list ipv6Pol
   host1(config-policy-list)#classifier-group sourceCLACL
   host1(config-policy-list-classifier-group)#forward
   host1(config-policy-list-classifier-group)#classifier-group *
   host1(config-policy-list-classifier-group)#filter
   ```

   The policy ipv6Pol is requesting classification on Source Address (all 4 words) and Protocol. Table 26 on page 177 lists the active classifiers in the policy named ipv6Pol and the size of each classifier.

Table 26: IPv6 Classification Fields for a 288-bit CAM Entry

Classifiers	Size (Bits)
Source address (first word)	32

Table 26: IPv6 Classification Fields for a 288-bit CAM Entry *(continued)*

Classifiers	Size (Bits)
Source address (second word)	32
Source Address (third word)	32
Source Address (fourth word)	32
Protocol	8

The sum of all classification fields requested in ipv6Pol is 136, which is greater than 128-bit CAM entry size limit. Although this configuration fails to attach to the interface in JunosE releases earlier than Release 10.2.0, it is successfully attached to the interface, beginning with JunosE Release 10.2.x, and the next higher 288-bit CAM entry is allocated for this policy (two 288-bit entries because of two classifiers being defined in the policy).

576-bit IPv6 Classification Example

In this example, a policy with a combination of IPv6 classifiers is created and attached.

1. Match all TCP packets from host 1:1:1:1:1:1:1:1 to host 100::1 destined to port 80 from source port 10000

   host1(config)#**ipv6 classifier-list tcpCLACL source-host 1:1:1:1:1:1:1:1 destination-host 100::1 tcp source-port eq 10000 destination-port eq 80**

2. Match all frames with the color red

   host1(config)#**ipv6 classifier-list colorCLACL color red**

3. Create an IPv6 policy list.

   host1(config)#**ipv6 policy-list ipv6Pol**
   host1(config-policy-list)#**classifier-group tcpCLACL**
   host1(config-policy-list-classifier-group)#**forward**
   host1(config-policy-list-classifier-group)#**classifier-group colorCLACL**
   host1(config-policy-list-classifier-group)#**forward**
   host1(config-policy-list-classifier-group)#**classifier-group ***
   host1(config-policy-list-classifier-group)#**filter**

   The policy ipv6Pol is requesting classification on Source Address (all 4 words), Destination address (all 4 words) and Protocol. Table 27 on page 178 lists the active classifiers in the policy named ipv6Pol and the size of each classifier.

Table 27: IPv6 Classification Fields for a 576-bit CAM Entry

Classifiers	Size (Bits)
Source address (first word)	32
Source address (second word)	32

Table 27: IPv6 Classification Fields for a 576-bit CAM Entry *(continued)*

Classifiers	Size (Bits)
Source Address (third word)	32
Source address (fourth word)	32
Destination Address (first word)	32
Destination address (second word)	32
Destination Address (third word)	32
Destination Address (fourth word)	32
Protocol	8
Destination Port	16
Source Port	16
Color	8

The sum of all classification fields requested in ipv6Pol is 304, which is greater than 128-bit CAM entry size limit. Although this configuration fails to attach to the interface in earlier releases, it is successfully attached to the interface, beginning with this release, and the maximum 576-bit CAM entry is allocated for this policy (three 576-bit entries, one for each classifier in the policy).

Related Documentation

- CAM Hardware Classifiers Overview on page 167
- Performance Impact and Scalability Considerations on page 179
- Size Limit for IP and IPv6 CAM Hardware Classifiers on page 168

Performance Impact and Scalability Considerations

The following sections describe how the memory usage and performance of the line modules on which the variable-sized CAM entries are supported is affected, and also of the maximum number of policies that can be supported with variable-sized CAM entries.

Performance Impact

Some performance impact might occur due to the variable size of the CAM entries. This performance impact is caused by CAM addressing, which works on 72 bits. 576-bit classification requests now require up to 8 lookups to the CAM hardware (8 * 72 = 576). The CAM device has a search rate of up to 83 million per second for 144 bit entries.

Scalability Considerations

One CAM entry is required per classifier for each unique policy on each line module. Regardless of the classifier definition for an IPv4 policy, each IPv4 classifier consumes 144 bits (one 144-bit CAM entry). However, default classifiers do not consume CAM entries.

As described in "Examples: Variable-Sized CAM Classification for IPv6 Policies" on page 175, an IPv6 CAM entry size is 144 bits, 288 bits, or 576 bits, depending on the sum of the classification fields in the policy definition. However, all IPv6 classifiers consume the same CAM entry size in a policy.

The following factors are used to determine the CAM resources available for policies when variable-sized CAM entries are present:

- CAM Device Block Size and CAM Entry Allocation on page 180
- Number of CAM Entries Per Allocation and Free Entries on page 180

CAM Device Block Size and CAM Entry Allocation

Using GE-2 line modules, for example, we can demonstrate how the number of CAM entries it supports is divided into different blocks to store policies. GE-2 line modules contain 64,000 144-bit CAM entries. Each entry is divided into eight 8000 144-bit blocks. Each block can hold equal-sized CAM entries only—144-bit, 288-bit, and 576-bit CAM entries. If no more IPv6 policies are created and when the remaining seven blocks are used, the 576-bit CAM block is not available to store IPv4 policies that require 144-bit CAM entries only.

A default classifier within a policy also consumes the same sized CAM entry as the size computed for the policy. In lower numbered releases, a single 144-bit entry was reserved for default classifiers. In this release, the number of 144-bit entries reserved for default classifiers depends on the number of blocks assigned for such CAM entries and whether the attached policy contains 288-bit or 576-bit entries. For example, if the first block is used by the 576-bit CAM entry, four 144-bit entries are reserved for the default classifier.

Number of CAM Entries Per Allocation and Free Entries

The total number of CAM blocks is divided into two equal partitions. The first or lower half of the CAM blocks is reserved for 144-bit CAM entries, and the second or higher half of CAM blocks is reserved for the combination of 288-bit and 576-bit CAM entries, when an IPv6 policy that contains 288-bit or 576-bit CAM entries is attached to an interface. If IPv6 policies do not contain 288-bit or 576-bit CAM entries, all the blocks are used for 144-bit entries.

Assume that, on a GE-2 line module, out of the total of eight blocks, four blocks are completely used for 144-bit CAM entries and the remaining four blocks are allocated in common for 144-bit, 288-bit, and 576-bit entries. Each of the blocks reserved exclusively for 144-bit entries can contain 8000 entries, while each of the blocks reserved for the combination of the variable-sized entries can either contain 2000 576-bit entries or 4000 288-bit entries. The block that is common to the variable-sized entries is available for

144-bit entries only if an IPv6 policy does not contain 288-bit or 576-bit entries. Otherwise, when the first IPv6 policy that contains 288-bit or 576-bit entries is attached to an interface and if previously configured policies consumes more than 4 blocks, the IPv6 policy attachment fails.

The block that is common to the variable-sized entries is not available for 144-bit CAM entries when you configure any 288-bit or 576-bit entries, even though you remove them later. It is also not available for any 288-bit or 576-bit entries when the 144-bit entries spill into this block, even though you remove the 144-bit entries later.

NOTE: ES2 4G LMs contain a total of 32 blocks, of which 16 blocks are assigned for 144-bit entries. The remaining 16 blocks are assigned for the combination of 144-bit, 288-bit, and 576-bit entries (pool common to these three variable-sized entries).

Table 28 on page 181 lists the maximum policies supported with variable length IPv6 CAM classification and one classifier per policy. The following note is referred to in Table 28 on page 181.

1. The number of unique policies supported depends on the line module and the numbers used are to illustrate the impact with CAM entries. The actual policies vary according to the line module.

Table 28: Maximum Policies with One Classifier per Policy for GE-2 LMs

Number/Type of Policies	Total 144-bit CAM entries	Number of IPv4 policies (144-bit) with one CLACL (See Note 1)	Number of IPv6 policies (144-bit) with one CLACL	Number of IPv6 policies (288-bit) with one CLACL (See Note 1)	Number of IPv6 policies (576-bit) with one CLACL (See Note 1)	Number of maximum policies per LM (one CLACL per policy) (See Note 1)
All IPv4 policies	64,000	64,000	0	0	0	64,000
All IPv6 policies	64,000	0	64,000	0	0	64,000
All IPv6 policies	64,000	0	0	16,000	0	16,000
All IPv6 policies	64,000	0	0	0	8000	8000
Equal number of identical IPv4/IPv6 policies	64,000	32,000	32,000	0	0	64,000
Equal number of identical IPv4/IPv6 policies	64,000	16,000	0	16,000	0	32,000 (+ 16,000 144-bit entries available)

Table 28: Maximum Policies with One Classifier per Policy for GE-2 LMs *(continued)*

Number/Type of Policies	Total 144-bit CAM entries	Number of IPv4 policies (144-bit) with one CLACL (See Note 1)	Number of IPv6 policies (144-bit) with one CLACL	Number of IPv6 policies (288-bit) with one CLACL (See Note 1)	Number of IPv6 policies (576-bit) with one CLACL (See Note 1)	Number of maximum policies per LM (one CLACL per policy) (See Note 1)
Equal number of identical IPv4/IPv6 policies	64,000	8000	0	0	8000	16,000 (+ 24,000 144-bit entries available)

Table 29 on page 182 lists the maximum policies supported with variable length IPv6 CAM classification and four classifiers per policy.

Table 29: Maximum Policies with Four Classifiers per Policy for GE-2 LMs

Number/Type of Policies	Total 144-bit CAM entries	Number of IPv4 policies (144-bit) with four CLACLs	Number of IPv6 policies (144-bit) with four CLACL s	Number of IPv6 policies (288-bit) with four CLACLs	Number of IPv6 policies (576-bit) with four CLACLs	Number of maximum policies per LM (four CLACLs per policy)
All IPv4 policies	64,000	16,000	0	0	0	16,000
All IPv6 policies	64,000	0	16,000	0	0	16,000
All IPv6 policies	64,000	0	0	4000	0	4000
All IPv6 policies	64,000	0	0	0	2000	2000
Equal number of identical IPv4/IPv6 policies	64,000	8000	8000	0	0	16,000
Equal number of identical IPv4/	64,000	4000	0	4000	0	8000 (+ 16,000 144-bit entries available)
Equal number of identical IPv4/IPv6 policies	64,000	2000	0	0	2000	4000 (+ 24,000 144-bit entries available)

Related Documentation

- Examples: Variable-Sized CAM Classification for IPv6 Policies on page 175
- Size Limit for IP and IPv6 CAM Hardware Classifiers on page 168

Software Classifiers Overview

An E Series router supports a variety of software classifiers, depending on the type of interface. "Policy Resources Overview" on page 163 lists the supported software classifiers for each interface type.

A line module supports 16,383 software classifiers. Software classifiers are consumed at a rate of one resource per classifier category per policy. For example, if you configure a policy that has three different destination route class rules, then because all three rules are for the same classifier category, that policy consumes only one software classifier resource. However, if you configure a policy that requires classification on three different classifier categories, such as ToS, color, and TCP flags, then that policy consumes three of the available 16,383 software classifier resources.

NOTE: Policy consumption is per policy definition per line module.

In this example, the policy list named polWestford5 references four classifier lists with a combination of software and hardware classifiers.

```
host1(config)#ip classifier-list clacl100 color red ip any any
host1(config)#ip classifier-list clacl200 color yellow user-packet-class 6 ip host  10.1.1.1
  host 10.1.1.2
host1(config)#ip classifier-list clacl300 color green user-packet-class 5 ip any any
host1(config)#ip classifier-list clacl400 color red ip host 10.1.1.10 any
host1(config)#ip policy-list polWestford5
host1(config-policy-list)#classifier-group clacl100
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group clacl200
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group clacl300
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group clacl400
host1(config-policy-list-classifier-group)#forward
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#classifier-group *
host1(config-policy-list-classifier-group)#filter
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#exit
```

For a given line module, the policy list named polWestford5 consumes a total of one FPGA hardware classifier resource and two software classifier resources, as indicated in Table 30 on page 184.

Table 30: Resource Consumption

Number of Resources Consumed	Classifier Category
1 hardware	• Protocol • Destination address • Source address
1 software	Color
1 software	User-packet-class

Related Documentation

- Policy Resources Overview on page 163

Interface Attachment Resources Overview

JunosE Software allocates interface attachment resources when policies are attached to interfaces—when you attach a policy to an interface, the policy consumes one of the interface's attachment resources. Each interface has two attachment resource pools. IP and IPv6 policy attachments are allocated from the interface's IP attachment resource pool; all other attachments are allocated from the interface's layer 2 attachment resource pool.

- The type of line module determines the number of policies attachments supported by interfaces. See *ERX Module Guide, Appendix A, Module Protocol Support* for more information about supported line modules. See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the modules that support BGP.
- On ASIC-based line modules (OC48/STM16, GE-2, and GE-HDE line modules), you can have a maximum of 8191 IP policy attachments and 8191 layer 2 policy attachments for ingress policies per forwarding controller, and 8191 IP policy attachments and 8191 layer 2 policy attachments for egress policies per forwarding controller.
- On FPGA-based line modules, you can have a maximum of 8191 IP policy attachments and 8191 layer 2 policy attachments per forwarding controller.

Related Documentation

- Policy Resources Overview on page 163

CAM Hardware Classifiers and Interface Attachment Resources

CAM hardware classifiers are supported on OC48/STM16, GE-2, and GE-HDE ASIC-based line modules. Policies that use CAM hardware classifiers consume one interface attachment resource, regardless of the number of classifier entries in a policy.

Related Documentation

- CAM Hardware Classifiers Overview on page 167
- Interface Attachment Resources Overview on page 184

Range Vector Hardware Classifiers and Interface Attachment Resources

Range vector classifiers, which include all software classifiers and FPGA-based hardware classifiers, consume one interface attachment resource for every 32 classifier entries in a policy.

The following examples illustrate how JunosE Software allocates interface attachment resources. These examples apply to software and FPGA-based hardware policies:

- A policy with 0 classifier entries consumes 1 interface attachment resource.
- A policy with 1–32 classifier entries consumes 1 interface attachment resource.
- A policy with 33–64 classifier entries consumes 2 interface attachment resources.
- A policy with 65–96 classifier entries consumes 3 interface attachment resources.
- A policy with 487–512 classifier entries consumes 16 interface attachment resources.

Related Documentation

- Interface Attachment Resources Overview on page 184

Detection of Corruption in the FPGA Statistics for Policies of Subscribers Managed by the SRC Software

When a bit flip occurs in the RAM of the field-programmable gate array (FPGA) statistics of a router that is functioning as the Session and Resource Control (SRC) client, the router may transmit erroneous and inconsistent statistical details for IPv4 and IPv6 subscribers to the SRC server or the Common Open Policy Service (COPS) server. This affects the computation of accounting information for subscriber sessions. When incorrect policy statistical details are sent from the SRC client, you can resolve the problem of inconsistent subscriber accounting only by replacing the defective hardware.

You can configure a software detection mechanism that identifies corruption in the FPGA statistics and prevents the SRC client from sending erroneous subscriber statistics to the SRC server. The capability to detect incorrect statistics operates by comparing the following statistical counters against a threshold value:

- Packets processed by an interface
- Packets for which policies are attached to the interface

If the difference between the interface counters and policy counters for ingress or egress policies collected over two polling intervals matches or exceeds the specified threshold value, a corruption is detected in the FPGA statistics and the subscriber statistics are not forwarded to the SRC server. If the difference between the interface counters and policy counters for ingress or egress policies collected over two polling intervals is less than the specified threshold value, no corruption is detected in the FPGA statistics and the collected subscriber statistical details are sent to the SRC server.

You can now use the **fpga-stats-monitoring-enable** command in Privileged Exec mode to enable the capability to detect corruption in the FPGA statistics and prevent the transmission of incorrect statistical details to the SRC server for subscriber policies managed by the SRC software. You can now use the **fpga-stats-monitoring threshold *thresholdValue*** command in Global Configuration mode to specify a threshold value to be used to determine corruption in the FPGA statistics. The threshold value is matched against the difference of the interface and policy counter values (for ingress and egress policies) collected over two consecutive polling periods.

In a Layer 2 Tunneling Protocol (L2TP) network that is established over a Point-to-Point Protocol (PPP) link between the router and the customer premises equipment (CPE) or the client, you can enable the router to manage subscriber policies using the SRC server. In such a network topology, the SRC client or the router sends COPS request messages to the SRC server. The SRC server sends provisioned policies to the SRC client, which installs the default service policies. When the SRC server sends a decision (DEC) packet to enable the policies to be attached to the interface, a new subscriber session is established after the user is successfully authenticated. The SRC client sends the Acct-Start message to the RADIUS server for the newly logged-in subscriber.

When the SRC server requests subscriber statistics counter values from the SRC client, which is also the RADIUS client, the router retrieves the accounting information by sending an Interim-Acct message to the RADIUS server and transmits the retrieved counter values to the SRC server. When the PPP session is terminated, the SRC client sends the Acct-Stop message to the RADIUS server and transmits the collected accounting details to the SRC server. The Delete Request (DRQ) messages are sent to the SRC server at this point.

The detection mechanism for corruption in the FPGA statistics is triggered for periodic DEC packets that the SRC client receives from the SRC server. You can set up the interval at which these DEC packets are sent in the SRC software. After you enable the corruption detection mechanism on the router or the SRC client, the detection feature is triggered when one of the following events occurs on the SRC client:

- Receipt of a DEC message from the SRC server to attach the service policy to an interface
- Receipt of a DEC message from the SRC server to retrieve interim accounting statistics
- Subscriber session goes down and the final accounting report is sent to the SRC server

Guidelines for Configuring the Capability to Detect Corruption in the FPGA Statistics

Keep the following points in mind when you configure the capability to detect corruption in the FPGA statistics. You must specify a threshold to determine discrepancies in the statistics:

- When a subscriber attempts to establish sessions over a defective slot where corruption in the FPGA statistics is detected, the subscriber will not be allowed to log in.
- The configuration settings related to the detection of corruption in the FPGA statistics are preserved across unified in-service software upgrade (ISSU), stateful switch route processor (SRP) switchover, and stateful line module switchover operations.
- For L2TP subscribers, the corruption in the FPGA statistics is detected on ES2 10G ADV line module (LMs) or ES2 4G LMs with Service input/output adapters (IOAs), and this

validation of the state of the FPGA statistics is not performed on the access interfaces. This method of detection occurs because the interface statistics are maintained only in the Service IOA.

- When a corruption is detected on ES2 4G LMs or ES2 10G ADV LMs with ES2-ES1 Service IOA, establishment of subscriber sessions over such line modules is not allowed. This prevention of creation of subscriber sessions occurs because the maximum number of tunnel-service interfaces that can be provisioned on a tunnel-server port is set to zero in such a case. New subscriber logins are not allowed and existing subscriber sessions are retained until they log out. You must remove stateful switchover configuration on such LMs to enable the secondary line module to handle new subscriber logins.
- Even if the interval to poll accounting statistics from the SRC client is configured at a higher frequency, such as at periodic intervals of one second on the SRC server, the performance of the router is not impacted because of the handling of such DEC messages from the SRC server.
- The session termination request is sent to the SRC server when corruption is detected for a slot over which the subscriber is logged in. The existing subscriber session is terminated and new subscribers cannot establish a session over the defective slot.
- The detection mechanism for corruption in the FPGA statistics has a limitation in the calculation of policy statistics when the ingress or egress traffic does not match any of the classifier rules configured within a policy. To avoid this discrepancy, a default classifier group should be added to the policy so that no traffic remains unaccounted.
- The detection mechanism for corruption in the FPGA statistics cannot detect bit flips in least significant bits, which result in statistics corruption lower than the configured threshold value.
- Information about defective slots is not persistent across unified ISSU, stateful SRP switchover, and stateful line module switchover operations. Therefore, if subscribers attempt to log in to a slot, which was determined to be corrupted prior to the restart of the router, they are permitted to log in until the detection capability classifies the slot to be defective again after the router went through a stateful reset.

Related Documentation

- Computation of the Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 188
- Example: Computation of the Threshold Value by Using Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 190
- Scenarios for the Detection of Corruption in the FPGA Statistics and the Determination of the Threshold on page 192
- Configuring the Capability to Detect Corruption in the FPGA Statistics for Policies Managed by the SRC Software on page 195
- Monitoring the Detection of Corrupted FPGA Statistics Settings on page 235

Computation of the Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics

The mechanism to detect corruption in the FPGA statistics functions by comparing the interface statistics (for either incoming or outgoing packets) and the aggregate of policy statistics (attached to input or output interfaces). These interface and policy counter values are obtained from the output of the **show ip interface** or **show ipv6 interface**command. A differential count of the interface and policy statistics is computed and the value is matched against a threshold value that you specified. If a discrepancy is observed during the detection process, the SRC client stops reporting statistics to the SRC server.

The detected discrepancy is recorded in a system logging message. The subscriber that logs in over a defective slot is logged out and new subscriber sessions are blocked on the defective slot.

To perform a comparison of the interface statistical counters and the policy statistical counters, the detection mechanism computes the policy counters based on certain factors and attributes. The following sections describe the calculation methods of policy counters for the detection of corrupted FPGA statistics.

Processing the Extra Header in Policy Counters

The policy counter that denotes the number of bytes of traffic to which policies are applied is always higher than the interface counter that denotes the number of bytes of traffic processed by an interface. The higher value of the policy counter is because of the extra header that it takes into consideration. In an L2TP topology, the following attributes are accounted for ingress and egress policy counter in bytes:

- Policy counter in bytes for ingress interfaces contains an additional value of 10 bytes per packet, which is caused by the headers (PPP header of 4 bytes and L2TP header of 6 bytes).
- Policy counter in bytes for egress interfaces contains an additional value of 38 bytes per packet, which is caused by the headers (IP header of 20 bytes, UDP header of 4 bytes, PPP header of 4 bytes, and L2TP header of 6 bytes).

The policy counter is calculated using the following formula:

Policy counter in bytes = (Policy counter in packets x Extra header) + Interface counter in bytes

Processing the Egress Policy Counters

The egress policy counters, as a measure of the number of packets and bytes, are always larger than the egress interface counters because some packets might be filtered by the outbound policy before they are forwarded out of the interface. The filtered packets and bytes counter is accounted as Out Policed Packets or Out Policed Bytes (in the output of the **show ip interface** command).

The policy counters, as a measure of the number of packets and bytes for egress policies, is calculated using the following formula:

Egress policy counter in packets = Policy counter in packets – Out Policed Packets counter

Egress policy counter in bytes = Policy counter in bytes – Out Policed Bytes Counter

Processing the Received Multicast Packets with Applied Policies

When the router receives certain destination packets on the PPP link, the policy statistics counter is not incremented because some of the packets are discarded even before they reach the policy statistics counter.

The interface counters, as a measure of the number of packets and bytes of traffic arriving on an interface, is calculated using the following formula:

Ingress interface counter in bytes = Multicast byte counter + Policy counter in bytes

Ingress interface counter in packets = Multicast packets counter + Policy counter in packets

This method of calculating counters is needed because in a multicast network, the number of received multicast packets is equal to the number of discarded packets.

Comparing the Interface and Policy Counters Over Two Polling Intervals

After computing the ingress and egress interface and policy counters to account for the extra header and multicast packet extra bytes, the interface and policy counters in bytes are stored in the application software. The detection mechanism for corruption in the FPGA statistics logic compares two successive retrieved values of the statistical counters to detect corruption as follows. Assume that interface and policy statistics are obtained at two intervals, namely interval_1 and interval_2. Interface_counter1 and Policy_counter1 counters are collected at interval_1, and Interface_counter2 and Policy_counter2 counters are collected at interval_2.

Difference between policy counters at interval_1 and interval_2 = delta_policy_counter = (policy_counter2 – policy_counter1)

Difference between interface counters at interval_1 and interval_2 = delta_interface_counter = (interface_counter2 – interface_counter1)

Difference between interface and policy counters collected at two intervals = delta_interface_counter – delta_policy_counter

The difference between the interface and policy counters derived at two successive intervals is compared against the configured threshold. The threshold is the maximum permissible deviation between interface and policy counter values. If the threshold is higher than the difference between the interface and policy counters, no corruption has occurred in the FPGA statistics. If the threshold is lower than the difference between the interface and policy counters, corruption has occurred in the FPGA statistics.

Related Documentation

- Detection of Corruption in the FPGA Statistics for Policies of Subscribers Managed by the SRC Software on page 185

- Example: Computation of the Threshold Value by Using Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 190
- Scenarios for the Detection of Corruption in the FPGA Statistics and the Determination of the Threshold on page 192
- Configuring the Capability to Detect Corruption in the FPGA Statistics for Policies Managed by the SRC Software on page 195
- Monitoring the Detection of Corrupted FPGA Statistics Settings on page 235

Example: Computation of the Threshold Value by Using Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics

The difference between interface and policy counters retrieved and computed during two consecutive polling intervals is compared with the configured threshold value. In the following examples, the differential value is zero, which signifies that no corruption is identified in the statistics. If the differential value is greater than or equal to the threshold value, corruption is detected.

The packets that do not match any of the classifier rules configured within a policy are considered as part of a default classifier control list. To avoid discrepancies in the calculated interface counters and policy counters, a default classifier group should be added to the policy so that no traffic remains unaccounted.

The following sections describe sample computations of the differences between interface and policy counters for ingress and egress packets.

Statistics Calculation for Incoming Packets

The following example describes how the difference between interface and policy counters for traffic arriving at an interface to which policies are applied is calculated:

Ingress Interface Packet Counter = In Received Packets = 775,132

Ingress Policy Packet Counter = Sum of Ingress Policy Counters = 0 + 1000 + 774,132 = 775,132

Difference between ingress policy and interface counters as numbers of packets = {(Ingress Interface Packet Counter – Ingress Policy Packet Counter) } = {(775,132 – 775,132)} = 0

The difference in counter values is 0. That means no corruption has occurred.

Ingress Interface Byte Counter = In Received Bytes = 155,026,400

Ingress Policy Byte Counter = Sum of Ingress Policy Bytes – (Ingress Policy Packet Counter x Extra Header)

The inbound policy byte counter contains an extra header of 10 bytes (PPP + L2TP).

Ingress Policy Byte Counter = 210,000 + 162,567,720 – (775,132 x 10) = 155,026,400

Difference between ingress policy and interface counters as numbers of bytes = ([Ingress Interface Byte Counter] – [Ingress Policy Byte Counter]) = ([155,026,400] – [155,026,400]) = 0

The difference in counter values is 0. That means no corruption has occurred.

Statistics Calculation for Outgoing Packets

The following example describes how the difference between interface and policy counters for traffic being forwarded from an interface and for which policies are applied is calculated:

Egress Interface Packet Counter = Out Forwarded Packets = 775,140

Egress Policy Packet Counter = Sum of Egress Policy Counter – Out Policed Packets

Egress Policy Packet Counter = (774,140 + 1000) – 0 = 775,140

Difference between egress policy and interface counters as numbers of packets = ([Egress Interface Packet Counter – Egress Policy Packet Counter]) = ([775,140] – [775,140]) = 0

The difference in counter values is 0. That means no corruption has occurred.

Egress Interface Byte Counter = Out Forwarded Bytes = 184,483,320

Egress Policy Byte Counter = (Sum of Egress Policy Bytes) – (Egress Policy Packet counter x Extra Header) – (Out Policed Bytes)

The outbound policy byte counter contains an extra header of 38 bytes (headers for IP, UDP, PPP, and L2TP).

Egress Policy Byte Counter = (213,662,640 + 276,000) – (775,140 x 38) – 0 = 184,483,320

Difference between egress policy and interface counters as numbers of bytes = ([Egress Interface Byte Counter] – [Egress Policy Byte Counter]) = ([184,483,320] – [184,483,320]) = 0

The difference in counter values is 0. That means no corruption has occurred.

Related Documentation

- Detection of Corruption in the FPGA Statistics for Policies of Subscribers Managed by the SRC Software on page 185
- Computation of the Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 188
- Scenarios for the Detection of Corruption in the FPGA Statistics and the Determination of the Threshold on page 192
- Configuring the Capability to Detect Corruption in the FPGA Statistics for Policies Managed by the SRC Software on page 195
- Monitoring the Detection of Corrupted FPGA Statistics Settings on page 235

Scenarios for the Detection of Corruption in the FPGA Statistics and the Determination of the Threshold

This section describes scenarios for the detection of corruption in the FPGA statistics and the determination of the threshold:

- When a bit flip occurs in policy counters
- When a bit flip occurs in interface counters
- When a policy is reattached to an interface
- When a bit flip occurs when a policy is attached to an interface for the first time

Bit Flip in Policy Counters

When a bit flip occurs in policy counters, the sum of the policy counters in different classifier groups is larger than the interface counter value. If the difference is greater than or equal to the configured threshold value, corruption in the FPGA statistics is detected.

Table 31: Interface and Policy Statistics When a Bit Flip Occurs in Policy Counters

Interval	Interface Counter	Policy Sum	Policy counter1	Policy counter2
Initial	0	0	0	0
Interval 1	1100	1100	100	1100
Interval 2	2200	1,000,002,200	1,000,000,200	2000

Interval 2 threshold = (100,001,100 [1,000,002,200 – 1100] – 1100 [2200-1100]) = 1,000,000,000

If the threshold is less than or equal to 1,000,000,000, corruption in the FPGA statistics is detected.

Bit Flip in Interface Counters

When a bit flip occurs in interface counters, the interface counter value is larger than the sum of the policy counters in different classifier groups. If the difference is greater than or equal to the configured threshold value, corruption in the FPGA statistics is detected.

Table 32: Interface and Policy Statistics When a Bit Flip Occurs in Interface Counters

Interval	Interface counter	Policy Sum	Policy counter1	Policy counter2
Initial	0	0	0	0
Interval 1	1100	1100	100	1000
Interval 2	1,000,002,200	2200	200	2000

Interval 2 threshold = (1100 [2200 – 1100] – 1,000,001,100 [1,000,002,200 – 1100]) = 1,000,000,000

If the threshold is less than or equal to 1,000,000,000, corruption in the FPGA statistics is detected.

Reattachment of a Policy to an Interface

Consider a scenario in which a policy is reapplied to an interface either because of a fast or a full resynchronization of the SRC server or because of a previously attached policy being removed and reapplied to the interface. In this case, the policy counters are reinitialized to 0, and the sum of policy counters in different classifier groups is less than the interface counters. If the difference is greater than or equal to the configured threshold value, corruption in the FPGA statistics is detected.

Table 33: Interface and Policy Statistics When a Policy is Reapplied to an Interface

Interval	Interface Counter	Policy Sum	Policy counter1	Policy counter2
Initial	0	0	0	0
Interval 1	1100	1100	100	1000
Interval 2	2200	0	0	0
Interval 3	3300	1100	100	1000

Interval 2 threshold = (-1100 [0-1100] – 1100 [2200-1100]) = 2200

If the threshold is less than or equal to 2200, corruption in the FPGA statistics is detected.

Bit Flip in Policy Counters When a Policy is Attached to an Interface for the First Time

In this scenario, a bit flip occurs in the policy statistical counter at the time of attachment of a policy to an interface. In such scenarios, the policy counters are larger than the interface counters even when the policy is applied for the first time. If the difference between policy and interface counters over polling intervals is greater than or equal to the configured threshold value, corruption in the FPGA statistics is detected.

Table 34: Interface and Policy Statistics When a Bit Flip Occurs when a Policy is Attached to an Interface for the First Time

Interval	Interface Counter	Policy Sum	Policy counter1	Policy counter2
Initial	0	1,000,000,000	1,000,000,000	0
Interval 2	1,100	1,000,001,100	1,000,000,100	1000
Interval 3	2200	1,000,002,200	1,000,000,200	2000

Initial threshold = (100,000,000 [1,000,000,000-0] – 0 [0-0]) = 1,000,000,000

If the threshold is less than or equal to 1,000,000,000, corruption in the FPGA statistics is detected.

Related Documentation

- Detection of Corruption in the FPGA Statistics for Policies of Subscribers Managed by the SRC Software on page 185
- Computation of the Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 188
- Example: Computation of the Threshold Value by Using Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 190
- Configuring the Capability to Detect Corruption in the FPGA Statistics for Policies Managed by the SRC Software on page 195
- Monitoring the Detection of Corrupted FPGA Statistics Settings on page 235

System Operations When Corrupted FPGA Statistics Is Detected

When you configure the software detection mechanism that identifies corruption in the FPGA statistics, the SRC client does not send erroneous subscriber statistics to the SRC server for policies managed by the SRC software based on the configured threshold value. The router or the SRC client might be configured to use the AAA server for authentication of subscribers that attempt to log in. The AAA server can be a server enabled with RADIUS for authenticating requests from the router, which is the AAA client or the RADIUS client.

If the difference between the interface counters and policy counters for ingress or egress policies collected over two polling intervals matches or exceeds the specified threshold value, a corruption is detected in the FPGA statistics for a slot over which the subscriber is logged in. In such a scenario, the existing subscriber session is terminated and new subscribers cannot establish a session over the defective slot. The AAA application stores details regarding the defective slot and the slot information is deleted when the line module configured in the affected slot is reloaded. The slot details are not preserved across a stateful switch route processor (SRP) switchover process.

Based on the configured threshold, you can configure the router or the SRC client to trigger SNMP traps when corruption is determined in the FPGA statistics.

If you do not enable stateful line module switchover or line module redundancy for a particular slot, the AAA application does not deactivate the slot where corruption is detected. In such a scenario, existing subscriber sessions are preserved and remain active over the affected slot. However, new subscribers cannot establish a session over the defective slot.

If you enable line module redundancy and if corruption in FPGA statistics is detected on a slot in which the primary line module configured for redundancy (stateless switchover) resides, the AAA application disables the affected slot and the standby line module takes over as the primary line module. In this scenario, all the existing subscriber sessions are disconnected and users need to log in again to reestablish their connections.

If you enable stateful line module switchover and if corruption in FPGA statistics is detected on a slot in which the primary line module configured for stateful switchover resides, the AAA application disables the affected slot and the standby line module takes over as the primary line module. When the standby module becomes the newly active primary module, incorrect statistics for affected subscribers are not preserved on the standby module because subscriber sessions are already terminated on the newly active primary module.

Related Documentation

- Detection of Corruption in the FPGA Statistics for Policies of Subscribers Managed by the SRC Software on page 185
- Computation of the Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 188
- Configuring the Capability to Detect Corruption in the FPGA Statistics for Policies Managed by the SRC Software on page 195
- Monitoring the Detection of Corrupted FPGA Statistics Settings on page 235
- fpga-stats-monitoring trap enable

Configuring the Capability to Detect Corruption in the FPGA Statistics for Policies Managed by the SRC Software

You can configure the router, which functions as an SRC client, to perform a validation of the FPGA statistics and identify any corruption in the statistical values that are computed based on interface and policy counters in the output of the **show ip interface** or **show ipv6 interface** commands. You must enable the capability to check the FPGA statistics for corruption and also specify a threshold value, exceeding which the FPGA statistics is determined to be defective. In such a scenario, you can prevent the SRC client from sending incorrect and discrepant statistics to the SRC server because of hardware corruption.

To configure the capability to check for corruption in the FPGA statistics for policies managed by the SRC server:

1. Enable the detection functionality to identify inaccuracies in the FPGA statistics before the counter values are reported to the SRC server during a COPS session.

   ```
   host1(config)#fpga-stats-monitoring-enable
   ```

   By default, the functionality to detect corruption in the FPGA statistics is disabled. Use the **no** version of this command to disable this functionality.

2. Specify a threshold value that is used as a checkpoint to determine whether the FPGA statistics is corrupted. The threshold is the maximum permissible deviation between interface and policy counter values. If the threshold is higher than the difference between the interface and policy counters, no corruption has occurred in the FPGA statistics. If the threshold is lower than the difference between the interface and policy counters, corruption has occurred in the FPGA statistics.

   ```
   host1(config)#fpga-stats-monitoring threshold 40
   ```

In this example, the threshold value is set as 40. If the difference between the interface counters and policy counters for ingress or egress policies collected over two polling intervals equals or exceeds 40, a corruption is detected in the FPGA statistics and the subscriber statistics are not forwarded to the SRC server. If the difference between the interface counters and policy counters for ingress or egress policies collected over two polling intervals is less than 40, no corruption is identified in the FPGA statistics and the collected subscriber statistical details are sent to the SRC server.

3. Enable the capability to generate SNMP traps when corruption is determined in the FPGA statistics.

   host1(config)#**fpga-stats-monitoring trap enable**

   By default, SNMP traps are not generated when corruption has occurred in the FPGA statistics and the threshold is lower than the difference between the interface and policy counters. Use the **no** version of this command to disable this functionality.

Related Documentation

- Detection of Corruption in the FPGA Statistics for Policies of Subscribers Managed by the SRC Software on page 185
- Computation of the Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 188
- Example: Computation of the Threshold Value by Using Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 190
- Scenarios for the Detection of Corruption in the FPGA Statistics and the Determination of the Threshold on page 192
- Monitoring the Detection of Corrupted FPGA Statistics Settings on page 235

CHAPTER 9

Monitoring Policy Management

This chapter explains how to set a statistics baseline and use the **show** command to display your policy configuration and monitor policy statistics.

This chapter discusses the following topics:

Monitoring Policy Management Overview

You can set a statistics baseline and use the **show** command to display your policy configuration and monitor policy statistics. When you set baseline statistics, you can retrieve statistics beginning at the time when the baselining is set. The policy log rule provides a way to monitor a packet flow by capturing a sample of the packets that satisfy the classification of the rule in the system log. See *JunosE System Event Logging Reference Guide* for information about logging.

NOTE: You can use the output filtering feature of the **show** command to include or exclude lines of output based on a text string you specify. See *Chapter 2, Command-Line Interface* in *JunosE System Basics Configuration Guide* for details.

Related Documentation

- Policy Management Configuration Tasks on page 6
- Policy Management Overview on page 3

Setting a Statistics Baseline for Policies

Purpose

You can set a baseline for policy statistics by using the **baseline interface** command and the **atm policy**, **frame-relay policy**, **gre-tunnel policy**, **ip policy**, **ipv6 policy**, **l2tp policy**, **mpls policy**, and **vlan policy** commands. If you do not enable baselining, **show** command output fields for baseline counters display the contents of the regular statistics counters.

If you enable statistics, you can enable or disable baselining of the statistics. The router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline when baseline-relative statistics are retrieved. Unlike other baseline statistics, policy baseline statistics are not stored in nonvolatile storage (NVS).

If you issue the **baseline interface** command for an interface without first enabling policy statistics baselining on that interface, a warning message indicates that policy baseline statistics are not enabled.

Enable a baseline for the statistics for the attachment of a policy list with statistics enabled to the ingress of an interface.

Action

Enable baseline counters.

```
host1(config)#interface atm 12/0.1
host1(config-subif)#ip policy input routeForXYZCorp statistics enabled baseline enabled
```

Run the **show ip interface** command with the **delta** keyword to show baseline counters:

```
host1#show ip interface atm 12/0.1 delta
atm12/0.1 is up, line protocol is up
  Network Protocols: IP
  Internet address is 200.200.1.1/255.255.255.0
  Broadcast address is 255.255.255.255
  Operational MTU = 9180  Administrative MTU = 0
  Operational speed = 155520000  Administrative speed = 0
  Discontinuity Time = 1251181
  Router advertisement = disabled
  Administrative debounce-time = disabled
  Operational debounce-time    = disabled
```

```
Access routing = disabled
Multipath mode = hashed

In Received Packets 5, Bytes 540
In Policed Packets 0, Bytes 0
In Error Packets 0
In Invalid Source Address Packets 0
In Discarded Packets 0
Out Forwarded Packets 5, Bytes 540
Out Scheduler Drops Packets 0, Bytes 0
Out Policed Packets 5, Bytes 540
Out Discarded Packets 0

IP Policy input routeForXYZCorp
  classifier-group *
     filter
     5 Packets  540 Bytes dropped
```

Related Documentation

- atm policy
- frame-relay policy
- gre-tunnel policy
- ip policy
- ipv6 policy
- l2tp policy
- mpls policy
- vlan policy

Monitoring the Policy Configuration of ATM Subinterfaces

Purpose Display information about a subinterface's ATM policy lists.

Action To display information about ATM policy lists:

```
host1#show atm subinterface
ATM policy input PolCbr
  classifier-group *
3096packets, 377678 bytes
traffic-class best-effort
color green
```

Meaning Table 35 on page 199 lists the **show atm subinterface** command output fields.

Table 35: show atm subinterface Output Fields

Field Name	Field Description
ATM policy	Type and name of the ATM policy
mark-clp	CLP bit value, 0 or 1
color	Color applied to packet flow for queuing: green, yellow, or red

Table 35: show atm subinterface Output Fields *(continued)*

Field Name	Field Description
classifier-group	Name of the classifier control list used by the policy
filter	Filter policy action
forward	Forward policy action
traffic-class	Traffic class in the policy list
user packet class	User packet class in the policy list

Related Documentation

- show atm interface

Monitoring Classifier Control Lists

Purpose Display a list of classifier control lists or details of classifier control lists.

Action To display a list of CLACLs:

host1#**show classifier-list**

```
                         Classifier Control List Table
                         ---------- ------- ---- -----
GRE Tunnel greClass.1
VLAN lowLatencyLowDrop.1
VLAN excellentEffort.1
VLAN bestEffort.1
VLAN lowLatency.1
IP wstFd.1 source-route-class 44 destination-route-class 55 3 any any
IP XYZCorpPermit.1 local true color green ip any any
IP routeForXYZCorp.1 color red tcp any any
IP XYZCorpIcmpEchoRequests.1 ip any any
IP XYZCorpPrecedence.1 tcp any any tos 5
IP XYZCorpPrecedence67.1 udp any any
IPv6 IPv6Precedence.1 color yellow
IPv6 IPv6Precedence67.1
L2TP l2tpclass.1 color green user-packet-class 8
MPLS mplsClass.1 user-packet-class 10 exp-bits 3 exp-mask 7
Frame relay frMatchDeSet.7 user-packet-class 8 de-bit 0
```

To display details of each CLACL:

host1#**show classifier-list detailed**

```
                         Classifier Control List Table
                         ---------- ------- ---- -----
IP Classifier Control List XYZCorpPermit
   Reference count:      1
   Entry count:          1

   Classifier-List XYZCorpPermit Entry 1
      Color:                      green
      Protocol:                   ip
```



```
      Not Protocol:               false
      Source IP Address:          0.0.0.0
      Source IP WildcardMask:     255.255.255.255
      Not Source Ip Address:      false
      Destination IP Address:     0.0.0.0
      Destination IP WildcardMask:255.255.255.255
      Not Destination Ip Address: false

GRE Tunnel Classifier Control List greClass
   Reference count:      0
   Entry count:          2

   Classifier-List greClass Entry 1
      User Packet Class:          8
      DS Field:                   3

   Classifier-List greClass Entry 2
      Color:                      yellow

VLAN Classifier Control List bestEffort
   Reference count:      0
   Entry count: 1

   Classifier-List bestEffort Entry 1
      Color:                      red
      User Packet Class:          15
      User Priority bits:         7

IPv6 Classifier Control List IPv6Classifier
   Reference count:      0
   Entry count:          1

   Classifier-List IPv6Classifier Entry 1
      User Packet Class:          3
      Traffic Class Field:        200

L2TP Classifier Control List l2tpclass
   Reference count:      0
   Entry count:          1

   Classifier-List l2tpclass Entry 1
      Color:                      green
      User Packet Class:          8

MPLS Classifier Control List mplsClass
   Reference count:      0
   Entry count:          1

   Classifier-List mplsClass Entry 1
      User Packet Class:          10
      EXP Bits:                   3
      EXP Mask:                   7
Frame relay Classifier Control List frMatchDeSet
   Reference count:      2
   Entry count:          1

   Classifier-List frMatchDeSet Entry 7
      Traffic Class:              toBoston
      User Packet Class:          8
      DE Bit:                     0
```

Meaning Table 36 on page 202 lists the **show classifier-list** command output fields.

Table 36: show classifier-list Output Fields

Field Name	Field Description
Reference count	Number of times the CLACL is referenced by policies
Entry count	Number of entries in the classifier list
Classifier-List	Name of the classifier list
Entry	Entry number of the classifier list rule
Color	Packet color to match: green, yellow, or red
Protocol	Protocol type
Not Protocol	If true, matches any protocol except the preceding protocol; if false, matches the preceding protocol
Source IP Address	Address of the network or host from which the packet is sent
Source IP WildcardMask	Mask that indicates addresses to be matched when specific bits are set
Not Source Ip Address	If true, matches any source IP address and mask except the preceding source IP address and mask; if false, matches the preceding source IP address and mask
Destination IP Address	Number of the network or host from which the packet is sent
Destination IP WildcardMask	Mask that indicates addresses to be matched when specific bits are set
Not Destination Ip Address	If true, matches any destination IP address and mask except the preceding destination IP address and mask; if false, matches the preceding destination IP address and mask
Traffic Class	Name of the traffic class to match
User Packet Class	User packet value to match
DS Field	DS field value to match
TOS Byte	ToS value to match
Precedence	Precedence value to match

Table 36: show classifier-list Output Fields *(continued)*

Field Name	Field Description
User Priority bits	User priority bits value to match
Traffic Class Field	Traffic class field value to match
EXP Bits	MPLS EXP bit value to match
EXP Mask	Mask applied to EXP bits before matching
DE Bit	Frame Relay DE bit value to match
Destination Route Class	Route class used to classify packets based on the packet's destination address
Source Route Class	Route class used to classify packets based on the packet's source address
Local	If true, matches packets destined to a local interface; if false, matches packets that are traversing the router

Related Documentation

- show classifier-list

Monitoring Color-Mark Profiles

Purpose Display information about color-mark profiles.

Action To display information about color-mark profiles:

```
host1#show color-mark-profile A
                            Color Mark Profile Table
                            ----- ---- ------- -----
IP Color-Mark-Profile: A
    Mask:                        255
    Green mark:                  64
    Yellow mark:                 -
    Red mark:                    8
```

Related Documentation

- show color-mark-profile

Monitoring Control Plane Policer Information

Purpose Display information about control plane policer for a specified protocol or all protocols.

Action To display information about control plane policer:

```
host1#show control-plane policer protocol
Burst
```

```
                                          Rate   Size   Packets    Packets
         Protocol                Enabled (pps) (pkts) Committed  Exceeded
----------------------------- ------- ----- ------ ---------- ----------
PppEchoRequest                  false    50     50          0          0
PppEchoReply                    false    50     50          0          0
PppEchoReplyFast                false    50     50          0          0
PppControl                      false    50     50          0          0
AtmControl                      false    50     50          0          0
AtmOam                          false    50     50          0          0
AtmDynamicIf                    false    50     50          0          0
AtmInverseArp                   false    50     50          0          0
FrameRelayControl               false    50     50          0          0
FrameRelayArp                   false    50     50          0          0
PppoeControl                    false    50     50          0          0
PppoePppConfig                  false    50     50          0          0
EthernetArp                     false    50     50          0          0
EthernetArpMiss                 false    50     50          0          0
EthernetLacp                    false    50     50          0          0
EthernetDynamicIf               false    50     50          0          0
```

Meaning Table 37 on page 204 lists the **show control-plane policer** command output fields.

Table 37: show control-plane policer Output Fields

Field Name	Field Description
Protocol	Name of the protocol
Enabled	True or False
Rate (pps)	Rate, in packets per second in the range 0–10000
Burst Size (pkts)	Burst size, in packets, in the range 0–10000
Packets Committed	Number of packets committed
Packets Exceeded	Number of packets exceeded

Related Documentation

- show control-plane policer

Monitoring the Policy Configuration of Frame Relay Subinterfaces

Purpose Display information about a subinterface's Frame Relay policy lists.

Action To display information about Frame Relay policy lists:

```
host1#show frame-relay subinterface
Frame relay sub-interface SERIAL5/0:1/1.1, status is up
Number of sub-interface down transitions is 0
Time since last status change 03:04:59
No baseline has been set
  In bytes: 660                  Out bytes: 660
  In frames: 5                   Out frames: 5
```



```
  In errors: 0                     Out errors: 0
  In discards: 0                   Out discards: 0
  In unknown protos: 0
Frame relay policy output frOutputPolicy
  classifier-group frGroupA entry 1
     5 packets, 640 bytes
     mark-de 1
Frame relay sub-interface SERIAL5/1:1/1.1, status is up
Number of sub-interface down transitions is 0
Time since last status change 03:05:09
No baseline has been set
  In bytes: 660                    Out bytes: 660
  In frames: 5                     Out frames: 5
  In errors: 0                     Out errors: 0
  In discards: 0                   Out discards: 0
  In unknown protos: 0
Frame relay policy input frInputPolicy
  classifier-group frMatchDeSet entry 1
      5 packets, 660 bytes
      color red
```

Meaning Table 38 on page 205 lists the **show frame-relay subinterface** command output fields.

Table 38: show frame-relay subinterface Output Fields

Field Name	Field Description
Frame Relay policy	Type and name of the VLAN policy
mark-de	DE bit value
color	Color applied to packet flow for queuing: green, yellow, or red
classifier-group	Name of the classifier control list used by the policy
filter	Filter policy action
forward	Forward policy action
traffic class	Traffic class in the policy list
user-packet-class	User packet class in the policy list

Related Documentation

- show frame-relay subinterface

Monitoring GRE Tunnel Information

Purpose Display information about GRE tunnels. The **state** keyword displays tunnels that are in a specific state: **disabled**, **down**, **enabled**, **not-present**, or **up**. The **ip** keyword to display tunnels associated with an IP address. To display information about a specific tunnel, include the name of the tunnel. To display information about tunnels on a specific virtual router, include the name of the virtual router.

Action To display information about GRE Tunnel policy lists:

```
host1#show gre tunnel detail tunnelGre50
GRE tunnel tunnelGre50 is Down
Tunnel operational configuration
  Tunnel mtu is '10240'
  Tunnel source address is '0.0.0.0'
  Tunnel destination address is '0.0.0.0'
  Tunnel transport virtual router is source
  Tunnel checksum option is disabled
  Tunnel sequence number option is disabled
  Tunnel up/down trap is enabled
  Tunnel-server location is 6/0
  Tunnel administrative state is Up
Statistics     packets        octets       discards    errors
  Data rx      0              0            0           0
  Data tx      0              0            0           0
GRE tunnel policy input routeGre25
  classifier-group gre6 entry 1
     0 packets, 0 bytes
     traffic-class best-effort
     mark 4 mask 255
GRE tunnel policy output routeGre35
  classifier-group gre14 entry 1
     0 packets, 0 bytes
     traffic-class best-effort
     mark 4 mask 255
```

Meaning Table 39 on page 206 lists the **show gre tunnel** command output fields.

Table 39: show gre tunnel Output Fields

Field Name	Field Description
GRE tunnel policy input	Policy for outbound traffic
GRE tunnel policy output	Policy for inbound traffic
traffic-class	Name of traffic class
classifier-group	Name of classifier group
entry	Identifier for the entry in the classifier group
packets	Number of packets
bytes	Number of bytes
mark	ToS byte setting for the classifier control list
mask	Mask value corresponding to the ToS

Related Documentation

- show gre tunnel

Monitoring Interfaces and Policy Lists

Purpose Display information about an interface and its policy lists. The **delta** keyword displays baselined statistics and the **brief** keyword displays the operational status of all configured interfaces

Action To display information about interfaces and policy lists:

NOTE: For tunnel interfaces, the packets field under the IP policy output section in the output of the **show ip interfaces** command displays the number of fragments or packets sent out from the tunnel interface for which an output policy is attached, depending on whether you enabled preservation of output policy statistics using the **enable-frag-stats** command. Although this field displays the unit of measure as packets, it denotes the number of fragments if statistics generation for output policies based on fragments is enabled. Otherwise, this field indicates the number of output policed packets on the tunnel interface.

```
host1#show interfaces fastEthernet 1/0.1
FastEthernet1/0.1 is Up, Administrative status is Up
  VLAN ID: 100

  In: Bytes 4156, Packets 30
   Errors 0, Discards 0
  Out: Bytes 6406, Packets 45
   Errors 0, Discards 0

VLAN policy input vlanPol1
  classifier-group vlan20 entry 1
    5 packets, 730 bytes
    filter

host1#show ip interfaces atm 5/0.2
ATM5/0.2 line protocol Atm1483 is down, ip is down (ready)
  Network Protocols: IP
  Internet address is 2.2.2.2/255.255.255.255
  Broadcast address is 255.255.255.255
  Operational MTU = 0  Administrative MTU = 0
  Operational speed = 100000000  Administrative speed = 0
  Discontinuity Time = 0
  Router advertisement = disabled
  Proxy Arp = disabled
  Network Address Translation is disabled
  TCP MSS Adjustment = disabled
  Administrative debounce-time = disabled
  Operational debounce-time    = disabled
  Access routing = disabled
  Multipath mode = hashed
  Auto Configure = disabled
  Auto Detect = disabled
  Inactivity Timer = disabled
```



```
In Received Packets 0, Bytes 0
  Unicast Packets 0, Bytes 0
  Multicast Packets 0, Bytes 0
In Policed Packets 0, Bytes 0
In Error Packets 0
In Invalid Source Address Packets 0
In Discarded Packets 0
Out Forwarded Packets 0, Bytes 0
  Unicast Packets 0, Bytes 0
  Multicast Routed Packets 0, Bytes 0
Out Scheduler Dropped Packets 0, Bytes 0
Out Policed Packets 0, Bytes 0
Out Discarded Packets 0

IP policy input P
  classifier-group data entry 1
    0 packets, 0 bytes
    rate-limit-profile rlpData
         committed rate: 10000 bps, committed burst: 8192 bytes (default)
         peak Rate: 100000 bps, peak burst: 1875 bytes
      committed: 0 packets, 0 bytes, action: transmit
      conformed: 0 packets, 0 bytes, action: transmit
      exceeded:  0 packets, 0 bytes, action: drop
  classifier-group voice entry 1
    0 packets, 0 bytes
    rate-limit-profile rlpVoice
         committed rate: 64000 bps, committed burst: 100000 bytes (default)
         peak Rate: 100000 bps, peak burst: 1875 bytes
      committed: 0 packets, 0 bytes, action: transmit
      conformed: 0 packets, 0 bytes, action: transmit
      exceeded:  0 packets, 0 bytes, action: drop
  classifier-group video entry 1
    0 packets, 0 bytes
    rate-limit-profile rlpVideo
         committed rate: 70000 bps, committed burst: 875 bytes
         peak Rate: 100000 bps, peak burst: 1875 bytes
      committed: 0 packets, 0 bytes, action: transmit
      conformed: 0 packets, 0 bytes, action: transmit
      exceeded:  0 packets, 0 bytes, action: drop
IP policy output P
  classifier-group data entry 1
    0 packets, 0 bytes
    rate-limit-profile rlpData
         committed rate: 20000 bps, committed burst: 150 bytes
         peak Rate: 200000 bps, peak burst: 3750 bytes
      committed: 0 packets, 0 bytes, action: transmit
      conformed: 0 packets, 0 bytes, action: transmit
      exceeded:  0 packets, 0 bytes, action: drop
  classifier-group voice entry 1
    0 packets, 0 bytes
    rate-limit-profile rlpVoice
         committed rate: 64000 bps, committed burst: 100000 bytes
         peak Rate: 200000 bps, peak burst: 3750 bytes
      committed: 0 packets, 0 bytes, action: transmit
      conformed: 0 packets, 0 bytes, action: transmit
      exceeded:  0 packets, 0 bytes, action: drop
  classifier-group video entry 1
    0 packets, 0 bytes
    rate-limit-profile rlpVideo
         committed rate: 140000 bps, committed burst: 850 bytes
         peak Rate: 200000 bps, peak burst: 3750 bytes
      committed: 0 packets, 0 bytes, action: transmit
```



```
        conformed: 0 packets, 0 bytes, action: transmit
        exceeded:  0 packets, 0 bytes, action: drop
```

Meaning Table 40 on page 209 lists the **show interfaces** command output fields.

Table 40: show interfaces Output Fields

Field Name	Field Description
Subinterface number	Location of the subinterface that carries the VLAN traffic
Administrative status	Operational state that you configured for this interface: up or down
VLAN ID	Domain number of the VLAN
In Bytes	Number of bytes received on the VLAN subinterface
In Packets	Sum of all unicast, broadcast, and multicast packets received on the VLAN or S-VLAN subinterface
In Errors	Value is always 0 (zero)
In Discards	Value is always 0 (zero)
Out Bytes	Number of bytes sent on the VLAN or stacked VLAN (S-VLAN) subinterface
Out Packets	Number of packets sent on the VLAN or S-VLAN subinterface
Out Errors	Value is always 0 (zero)
Out Discards	Value is always 0 (zero)
VLAN policy	Type and name of the VLAN policy

Related Documentation

- show interfaces

Monitoring the Policy Configuration of IP Interfaces

Purpose Display information about an IP interface (including policy list statistics).

Action To display information about IP policy lists on ATM interfaces:

`host1#`**show ip interface atm 11/0.6**

```
ATM11/0.6 line protocol Atm1483 is down, ip is down (ready)
  Network Protocols: IP
```

```
Internet address is 10.12.1.1/255.255.255.0
Broadcast address is 255.255.255.255
Operational MTU = 0  Administrative MTU = 0
Operational speed = 100000000  Administrative speed = 0
Discontinuity Time = 0
Router advertisement = disabled
Proxy Arp = disabled
ARP spoof checking = enabled
Network Address Translation is disabled
TCP MSS Adjustment = disabled
Administrative debounce-time = disabled
Operational debounce-time    = disabled
Access routing = disabled
Multipath mode = hashed
Auto Configure = disabled
Auto Detect = disabled
Re-Authenticate Auto Detect = disabled
Inactivity Timer = disabled
Use Framed Routes = disabled
Warm-restart initial-sequence-preference: Operational = 0 Administrative = 0

In Received Packets 0, Bytes 0
  Unicast Packets 0, Bytes 0
  Multicast Packets 0, Bytes 0
In Policed Packets 0, Bytes 0
In Error Packets 0
In Invalid Source Address Packets 0
In Discarded Packets 0
Out Forwarded Packets 0, Bytes 0
  Unicast Packets 0, Bytes 0
  Multicast Routed Packets 0, Bytes 0
Out Scheduler Dropped Packets 0, Bytes 0
Out Policed Packets 0, Bytes 0
Out Discarded Packets 0

IP policy output testPol
  classifier-group *
    0 packets, 0 bytes
    forward
       Virtual-router: default
       List:
         interface ATM11/0.1, order 100, rule 2 (not supported in output
         policy)
         next-hop 1.1.1.1, order 100, rule 3 (active) (not supported in
         output policy)
    rate-limit-profile R2
      committed rate: 0 bps, committed burst: 0 bytes (default)
      peak rate: 0 bps, peak burst: 0 bytes (default)
      committed: 0 packets, 0 bytes, action: transmit conditional
      conformed: 0 packets, 0 bytes, action: transmit conditional
      exceeded:  0 packets, 0 bytes, action: drop
    log
  classifier-group video entry 1
    0 packets, 0 bytes
    filter
```

To display information about IP policy lists on tunnel interfaces:

NOTE: If you enable scheduler profile–based computation of service session accounting by using the **service-accounting- statistics scheduler-based** command, the forwarded packets and bytes fields, and the dropped packets and bytes fields are displayed in the rate-limit-profile section under the IP policy output heading for policies with hierarchical parent groups. The committed, conformed, exceeded, saturated, and unconditional packets and bytes fields are not displayed in the rate-limit-profile section in the output of the command for policies with hierarchical parent groups. These fields are displayed instead of the forwarded packets and bytes fields, and the dropped packets and bytes fields only if you disable scheduler profile-based computation of service session accounting.

NOTE: The packets field in the IP policy output section displays the number of fragments or packets sent out from the tunnel interface for which an output policy is attached, depending on whether you enabled preservation of output policy statistics using the **enable-frag-stats** command. Although this field displays the unit of measure as packets, it denotes the number of fragments if statistics generation for output policies based on fragments is enabled. Otherwise, this field indicates the number of output policed packets on the tunnel interface.

host1#**show ip interface tunnel l2tp:1/19/21**

```
Out Forwarded Packets 1, Bytes 1500
    Unicast Packets 1, Bytes 1500
    Multicast Routed Packets 0, Bytes 0
  Out Scheduler Dropped Packets 0, Bytes 0
  Out Policed Packets 0, Bytes 0
  Out Discarded Packets 0

  IP policy output POLICY-USER-OUT
    classifier-group USER-OUT-PERMIT entry 1
      2 packets, 1596 bytes
      forward
  queue 0: traffic class best-effort, bound to ip TUNNEL l2tp:1/19/21
    Queue length 0 bytes
    Forwarded packets 2, bytes 1590
```

Meaning Table 41 on page 211 lists the **show ip interfaces** command output fields.

Table 41: show ip interfaces Output Fields

Field Name	Field Description
Network Protocols	Protocols configured on the interface
Internet address	IP address of the interface

Table 41: show ip interfaces Output Fields *(continued)*

Field Name	Field Description
Broadcast address	Broadcast address used by the interface
Operational MTU	Operational maximum transmission unit (MTU) for packets sent on this interface
Administrative MTU	Administrative maximum transmission unit for packets sent on this interface
Operational speed	Speed known to the IP layer in bits per second; equal to the administrative speed if configured, otherwise inherited from the lower layer
Administrative speed	Configured speed known to the IP layer in bits per second
Discontinuity Time	Time since the counters on the interface became invalid; for example, when the line module was reset
Router Advertisement	When enabled by the **ip irdp** command, the router advertises its presence via the ICMP Router Discovery Protocol (IRDP)
Administrative debounce-time	Administrative time delay that an interface must remain in a new state before the routing protocols react to the state change
Operational debounce-time	Time delay that an interface must remain in a new state before the routing protocols react to the state change
Access routing	When enabled, an access route is installed to the host on the other end of the interface
In Received Packets	Number of packets received on the interface; indicates whether packets are unicast or multicast
In Received Bytes	Number of bytes received on the interface; indicates whether bytes are unicast or multicast
In Policed Packets	Number of packets policed on the interface; discarded because they exceeded a traffic contract to their destination
In Policed Bytes	Number of bytes policed on the interface; discarded because they exceeded a traffic contract to their destination
In Error Packets	Number of packets determined to be in error at the interface

Table 41: show ip interfaces Output Fields *(continued)*

Field Name	Field Description
In Invalid Source Address Packets	Number of packets determined to have originated from an invalid source address
Out Forwarded Packets	Number of packets forwarded from the interface; indicates whether packets are unicast or multicast
Out Forwarded Bytes	Number of bytes forwarded from the interface; indicates whether bytes are unicast or multicast
Out Scheduler Drops Packets	Number of packets dropped by the out scheduler; indicates whether packets are committed, conformed, or exceeded
Out Scheduler Drops Bytes	Number of bytes dropped by the out scheduler; indicates whether bytes are committed, conformed, or exceeded
Policy	Indicates which policy is attached and whether it is on the input or output of the interface
classifier-group	Name of a CLACL attached to the interface and number of entry
exception http-redirect	Number of packets and bytes assigned to http-redirect
filter	Number of packets and bytes dropped because of the CLACL
color	Explicit color applied to packet flow for queuing; green, yellow, or red:
Packets logged	Number of packets colored
Bytes logged	Number of bytes colored
next hop	Address of the next-hop destination:
Packets transmitted	Number of packets sent to the next-hop address
Bytes transmitted	Number of bytes sent to the next-hop address
forward	Number of packets and bytes forwarded because of the CLACL
interface	Interface rule to forward all packets that match the current classifier control list

Table 41: show ip interfaces Output Fields *(continued)*

Field Name	Field Description
next-hop	Next-hop IP addresses are used as forwarding solutions, and the order of the rule within the classifier that the router uses to choose the solutions. The phrase in parentheses describes whether the rule entry is reachable, active, and supported for the configured policy
rate-limit-profile	Name of the rate-limit profile
committed	Number of packets and bytes within the committed rate limit
conformed	Number of packets and bytes exceeding the committed rate limit but within the peak rate
exceeded	Number of packets and bytes exceeding the peak rate
action	Action performed on the packets matched by the rules in the rate-limit profile

Related Documentation

- show ip interface

Monitoring the Policy Configuration of IPv6 Interfaces

Purpose Display detailed or summary information, including policy and classifier information, for a particular IPv6 interface or for all interfaces. The default for the **show ipv6 interface** command is all interface types and all interfaces. The **brief** or **detail** keywords with the **show ipv6 interface** command displays different levels of information.

Action To display information about IPv6 policy lists:

NOTE: If you enable scheduler profile–based computation of service session accounting by using the **service-accounting- statistics scheduler-based** command, the forwarded packets and bytes fields, and the dropped packets and bytes fields are displayed in the rate-limit-profile section under the IPv6 policy output heading for policies with hierarchical parent groups. The committed, conformed, exceeded, saturated, and unconditional packets and bytes fields are not displayed in the rate-limit- profile section in the output of the command for policies with hierarchical parent groups. These fields are displayed instead of the forwarded packets and bytes fields, and the dropped packets and bytes fields only if you disable scheduler profile-based computation of service session accounting.


```
host1#show ipv6 interface FastEthernet 9/0.6
FastEthernet9/0.6 line protocol VlanSub is up, ipv6 is up
  Description: IPv6 interface in Virtual Router Hop6
  Network Protocols: IPv6
  Link local address: fe80::90:1a00:740:31cd
  Internet address: 2001:db8:1::/48
  Operational MTU 1500 Administrative MTU 0
  Operational speed 100000000 Administrative speed 0
  Creation type Static
  ND reachable time is 3600000 milliseconds
  ND duplicate address detection attempts is 100
  ND neighbor solicitation retransmission interval is 1000 milliseconds
  ND proxy is enabled
  ND RA source link layer is advertised
  ND RA interval is 200 seconds, lifetime is 1800 seconds
  ND RA managed flag is disabled, other config flag is disabled
  ND RA advertising prefixes configured on interface

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Total Dropped Packets 0, Bytes 0
    In Policed Packets 0
    In Invalid Source Address Packets 0
    In Error Packets 0
    In Discarded Packets 0

  Out Forwarded Packets 8, Bytes 768
    Unicast Packets 8, Bytes 768
    Multicast Routed Packets 0, Bytes 0
  Out Total Dropped Packets 5, Bytes 0
    Out Scheduler Dropped Packets 0, Bytes 0
    Out Policed Packets 0
    Out Discarded Packets 5

  IPv6 policy input ipv6InPol25
   classifier-group *
      0 packets, 0 bytes
      forward
         Virtual-router: default
         List:
           next-hop 2001:82ab:1020:87ec::/64, order 100, rule 3 (active)
           (not supported in output policy)
   rate-limit-profile Rlp2Mb classifier-group clgA entry 1
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
    rate-limit-profile Rlp8Mb
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
  IPv6 policy output ipv6PolOut2
    rate-limit-profile RlpOutA classifier-group clgB entry 1
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
    rate-limit-profile RlpOutB
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
  IPv6 policy local-input ipv6PolLocIn5
    rate-limit-profile Rlp1Mb classifier-group clgC entry 1
```

```
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
    rate-limit-profile Rlp5Mb
      Committed: 0 packets, 0 bytes
      Conformed: 0 packets, 0 bytes
      Exceeded:  0 packets, 0 bytes
  queue 0: traffic class best-effort, bound to ipv6 FastEthernet9/0.6
    Queue length 0 bytes
    Forwarded packets 0, bytes 0
    Dropped committed packets 0, bytes 0
    Dropped conformed packets 0, bytes 0
    Dropped exceeded packets 0, bytes 0
```

Meaning Table 42 on page 216 lists the **show ipv6 interface** command output fields.

Table 42: show ipv6 interface Output Fields

Field Name	Field Description
Description	Optional description for the interface or address specified
Network Protocols	Network protocols configured on this interface
Link local address	Local IPv6 address of this interface
Internet address	External address of this interface
Operational MTU	Value of the MTU
Administrative MTU	Value of the MTU if it has been administratively overridden using the configuration
Operational speed	Speed of the interface
Administrative speed	Value of the speed if it has been administratively overridden using the configuration
Creation type	Method by which the interface was created (static or dynamic)
ND reachable time	Amount of time (in milliseconds) that the neighbor is expected to remain reachable
ND duplicate address detection attempts	Number of times that the router attempts to determine a duplicate address
ND neighbor solicitation retransmission interval	Amount of time (in milliseconds) during which the router retransmits neighbor solicitations
ND proxy	Whether the router replies to solicitations on behalf of a known neighbor, enabled or disabled
ND RA source link layer	Whether the RA includes the link layer

Table 42: show ipv6 interface Output Fields *(continued)*

Field Name	Field Description
ND RA interval	Amount of time (in seconds) of the neighbor discovery router advertisement
ND RA lifetime	Amount of time (in seconds) of the neighbor discovery router advertisement
ND RA managed flag	State of the neighbor discovery router advertisement managed flag, enabled or disabled
ND RA other config flag	State of the neighbor discovery router advertisement other config flag, enabled or disabled
ND RA advertising prefixes	Whether advertisement prefixes for neighbor discovery router advertisement are configured
In Received Packets, Bytes	Total number of packets and bytes received on this interface
Unicast Packets, Bytes	Number of unicast packets and bytes received on the IPv6 interface; link-local received multicast packets (non-multicast-routed frames) are counted as unicast packets
Multicast Packets, Bytes	Number of multicast packets and bytes received on the IPv6 interface, which are then multicast-routed and counted as multicast packets
In Total Dropped Packets, Bytes	Total number of inbound packets and bytes dropped on this interface
In Policed Packets	Number of packets that were received and dropped because of rate limits
In Invalid Source Address Packets	Number of packets received with invalid source address (for example, spoofed packets)
In Error Packets	Number of packets received with errors
In Discarded Packets	Number of packets received that were discarded for reasons other than rate limits, errors, and invalid source address
Out Forwarded Packets, Bytes	Total number of packets and bytes that were sent from this interface
Unicast Packets, Bytes	Number of unicast packets and bytes that were sent from this interface

Table 42: show ipv6 interface Output Fields *(continued)*

Field Name	Field Description
Multicast Routed Packets, Bytes	Number of multicast packets and bytes that were sent from this interface
Out Total Dropped Packets	Total number of outbound packets and bytes dropped by this interface
Out Scheduler Dropped Packets, Bytes	Number of outbound packets and bytes dropped by the scheduler
Out Policed Packets, Bytes	Number of outbound packets and bytes dropped because of rate limits
Out Discarded Packets	Number of outbound packets that were discarded for reasons other than those dropped by the scheduler and those dropped because of rate limits
IPv6 policy	Type (input, output, local-input) and name of the policy
forward	Number of packets and bytes forwarded because of the CLACL
next-hop	Next-hop IPv6 addresses are used as forwarding solutions, and the order of the rule within the classifier that the router uses to choose the solutions. The phrase in parentheses describes whether the rule entry is reachable, active, and supported for the configured policy
rate-limit-profile	Name of the profile
classifier-group entry	Entry index
Committed	Number of packets and bytes that conform to the committed access rate
Conformed	Number of packets and bytes that exceed the committed access rate but conform to the peak access rate
Exceeded	Number of packets and bytes that exceed the peak access rate
queue, traffic class, bound to ipv6	Queue and traffic class bound to the specified IPv6 interface
Queue length	Number of bytes in the queue

Table 42: show ipv6 interface Output Fields *(continued)*

Field Name	Field Description
Dropped committed packets, bytes	Total number of committed packets and bytes dropped by this interface
Dropped conformed packets, bytes	Total number of conformed packets and bytes dropped by this interface
Dropped exceeded packets, bytes	Total number of exceeded packets and bytes dropped by this interface

Related Documentation

- show ipv6 interface

Monitoring the Policy Configuration of Layer 2 Services over MPLS

Purpose Display status and configuration information about layer 2 services over MPLS (also known as Martini, or layer 2 transport) on the router or on specific interfaces. Displays only layer 2 circuits for the specified interface.

Action To display information about layer 2 services over MPLS policy lists:

```
host1#show mpls l2transport interface
FastEthernet9/0.1
  routed to 222.9.1.3  on base LSP  tun mpls:lsp-de090100-24-37
  group-id 2 vc-id 900001 mtu 1500
  State UP
  In  Label 48 on stack
    0 pkts, 0 hcPkts, 0 octets
    0 hcOctets, 0 errors, 0 discardPkts

  Out Label 49 on  tun mpls:lsp-de090100-24-37
    0 pkts, 0 hcPkts, 0 octets
    0 hcOctets, 0 errors, 0 discardPkts
    queue 0: traffic class best-effort, bound to atm-vc ATM1/0.1
      Queue length 0 bytes
      Forwarded packets 0, bytes 0
      Dropped committed packets 0, bytes 0
      Dropped conformed packets 0, bytes 0
      Dropped exceeded packets 0, bytes 0

    MPLS policy input mplsInputPolicy
      classifier-group claclWst50 entry 1
        0 packets, 0 bytes
        rate-limit-profile rlp
          committed: 0 packets, 0 bytes, action: transmit
          conformed: 0 packets, 0 bytes, action: transmit
          exceeded:  0 packets, 0 bytes, action drop
    MPLS policy output mplsOutputPolicy
      classifier-group claclWst75 entry 1
        0 packets, 0 bytes
        rate-limit-profile rlp
          committed: 0 packets, 0 bytes, action: transmit
          conformed: 0 packets, 0 bytes, action: transmit
          exceeded:  0 packets, 0 bytes, action: drop
```


Meaning Table 43 on page 220 lists the **show mpls l2transport interface** command output fields.

Table 43: show mpls l2transport interface Output Fields

Field Name	Field Description
Interface	Specifier and status of each interface
base-LSP/remote-addr	Identifies either the tunnel that is selected to forward the traffic or the address of the router at the other end
group-id	Group ID number for the interface
vc-id	VC ID number for the interface
mtu	Maximum transmission unit for the interface
state/in/out-label	Status of the Layer 2-over-MPLS connection or the incoming/outgoing VC label
Mpls Statistics	
pkts	Number of packets received or sent
hcPkts	Number of high-capacity (64-bit) packets received or sent
octets	Number of octets received or sent
hcOctets	Number of high-capacity (64-bit) octets received or sent
errors	Number of packets that are dropped for some reason at receipt or before being sent
discardPkts	Number of packets that are discarded due to lack of buffer space at receipt or before being sent
queue, traffic class, bound to	Queue and traffic class bound to the specified interface
Queue length	Number of bytes in queue
Forwarded packets, bytes	Total number of packets and bytes forwarded by this interface
Dropped committed packets, bytes	Total number of committed packets and bytes dropped by this interface
Dropped conformed packets, bytes	Total number of conformed packets and bytes dropped by this interface

Table 43: show mpls l2transport interface Output Fields *(continued)*

Field Name	Field Description
Dropped exceeded packets, bytes	Total number of exceeded packets and bytes dropped by this interface
MPLS policy	Type (input, output) and name of policy
classifier-group	Name of a CLACL attached to the interface and number of entry
rate-limit-profile	Name of profile
Committed	Number of packets and bytes conforming to the committed access rate
Conformed	Number of packets and bytes that exceed the committed access rate but conform to the peak access rate
Exceeded	Number of packets and bytes exceeding the peak access rate

Related Documentation

- show mpls

Monitoring External Parent Groups

Purpose Display information about external parent groups.

Action To display information about external parent groups:

```
host1#show parent-group name EPG2
                                  Parent Group Table
                                  ------ ----- ------
Parent Group EPG2
   Reference count: 1
   Rate limit profile: VLAN_RATE
   Next parent group: EPG1 parameter C

   Referenced by policies:
     P1
```

Meaning Table 44 on page 221 lists the **show parent-group** command output fields.

Table 44: show parent-group Output Fields

Field Name	Field Description
Reference count	Number of references within policies and other external parent groups.
Rate limit profile	Name of hierarchical rate limit profile.

Table 44: show parent-group Output Fields *(continued)*

Field Name	Field Description
Next parent group	Name of the next parent group and parameter.
Referenced by policies	List of policies where this parent group is referenced.
Referenced by parent groups	List of parent groups where the parent group is referenced.

Related Documentation

- show parent-group

Monitoring Policy Lists

Purpose Display information about policy lists.

Action To display policy lists:

```
host1#show policy-list
                                  Policy Table
                                  ------ -----
IPv6 Policy ipv6-pol8
   Administrative state: enable
   Reference count:      2
   Classifier control list: *, precedence 100
      forward
         Virtual-router: default
         List:
           next-hop 3001:82ab:1020:87ec::/64, order 10, rule 2 (active)
         Virtual-router: vr1
         List:
           next-hop 2001:82ab:1020:87ec::/64, ignore-default-route, order 20,
           rule 3

   Referenced by interface(s):
      GigabitEthernet1/0/2.1  input policy, statistics enabled, virtual-router
default
      GigabitEthernet1/0/2.1  output policy, statistics enabled, virtual-router
default

   Referenced by profile(s):
      None

   Referenced by merged policies:
      None

IP Policy routeForABCCorp
   Administrative state: enable
   Reference count:      0
 atm-cell-mode: enabled
Classifier control list: ipCLACL10, precedence 75
 exception http-redirect
 forward
Virtual-router: default
```



```
         List:
           next-hop 192.0.2.12, order 10, rule 2 (active)
           next-hop 192.0.100.109, order 20, rule 3 (reachable)
           next-hop 192.120.17.5, order 30, rule 4 (reachable)
           interface ip3/1, order 40, rule 5
      mark tos 125
      rate-limit-profile ipRLP25
   Classifier control list: ipCLACL20, precedence 125
      filter

IPv6 Policy routeForIPv6
   Administrative state: enable
   Reference count:      0
   Classifier control list: ipv6tc67, precedence 75
      forward
         Virtual-router: default
         List:
           next-hop 3001:82ab:1020:87ec::/64, order 10, rule 2 (active)
         Virtual-router: vr1
         List:
           next-hop 2001:82ab:1020:87ec::/64, ignore-default-route, order 20,
rule 3
      color red
      mark tc-precedence 7

Frame relay Policy frOutputPolicy
   Administrative state: enable
   Reference count:      0
   Classifier control list: frMatchDeSet, precedence 100
      mark-de 1

Frame relay Policy frInputPolicy
   Administrative state: enable
   Reference count:      0
   Classifier control list: frMatchDeSet, precedence 100
      color red

GRE Tunnel Policy routeGre50
   Administrative state: enable
   Reference count:      0
   Classifier control list: gre8, precedence 150
      color red
      mark dsfield 20
      filter

L2TP Policy routeForl2tp
   Administrative state: enable
   Reference count:      0
   Classifier control list: *, precedence 100
      color red
      rate-limit-profile l2tpRLP20

MPLS Policy routeForMpls
   Administrative state: enable
   Reference count:      0
   Classifier control list: *, precedence 200
      mark-exp 2 mask 7
      rate-limit-profile mplsRLP5

VLAN Policy routeForVlan
   Administrative state: enable
   Reference count:      0
   Classifier control list: lowLatencyLowDrop, precedence 100
      traffic-class lowLatencyLowDrop
```



```
      color green
      mark-user-priority 7
   Classifier control list: lowLatency, precedence 100
      traffic-class lowLatency (suspended)
   Classifier control list: excellentEffort, precedence 100
      traffic-class excellentEffort
   Classifier control list: bestEffort, precedence 100
      traffic-class bestEffort
```

To display component policies:

`host 1#`**show policy-list comp_p1**

```
                                  Policy Table
                                  ------ -----
IP Policy comp_p1
   Administrative state: enable
   Reference count:      7
   Classifier control list: C1, precedence 90
      forward
         Virtual-router: default
         List:
           next-hop 10.1.1.1, order 100, rule 2 (active)
   Classifier control list: C2, precedence 10
      filter

   Referenced by interfaces:
      ATM3/0.3  input policy, statistics enabled, virtual-router vr1
      ATM3/0.4  output policy, statistics disabled, virtual-router vr1
      ATM3/0.5  secondary-input policy, statistics enabled, virtual-router vr1

   Referenced by profiles:
      prof_1  input policy, statistics disabled

   Referenced by merge policies:
      mpl_10
      mpl_11
      mpl_12
```

`host1#`**show policy-list comp_p2**

```
                                  Policy Table
                                  ------ -----
IP Policy comp_p2
   Administrative state: enable
   Reference count:      1
   Classifier control list: C1, precedence 90
      color red
   Classifier control list: *, precedence 1000
      filter

   Referenced by interfaces:
      ATM4/0.5  input policy, statistics enabled, virtual-router default

   Referenced by profiles:
      None

   Referenced by merge policies:
      None
```

To display component policies:

`host1#`**show policy-list mpl_10**

```
                                  Policy Table
                                  ------ -----
```



```
IP Policy mpl_10
   Administrative state: enable
   Reference count:      1
   Classifier control list: C1, precedence 90
      forward
         Virtual-router: default
         List:
           next-hop 10.1.1.1, order 100, rule 2 (active)
           next-hop 20.1.1.1, order 100, rule 3 (reachable)
   Classifier control list: C2, precedence 10
      filter
   Classifier control list: C3, precedence 10
      filter
   Classifier control list: *, precedence 1000
      forward

   Referenced by interfaces:
      ATM5/0.1  input policy, statistics enabled, virtual-router default

   Referenced by profiles:
      None

   Component policies:
      comp_p1
      comp_p3
```

To display the configuration of an IP policy list that contains inactive references to the interface to which it is attached:

host1#**show policy-list pv4**

```
                                 Policy Table
                                  ------ -----
IP Policy pv4
   Administrative state: enable
   Reference count:      2 (*)
   Classifier control list: cv4, precedence 100
      forward
   Classifier control list: *, precedence 100
      filter

   Referenced by interface(s):
      GigabitEthernet12/1.0  input policy, statistics disabled, virtual-router
default

   Referenced by profile(s):
      None

   Referenced by merged policies:
      None
```

To display rate limit hierarchy in one policy:

host1#**show policy-list P1**

```
                                  Policy Table
                                  ------ -----
IP Policy P1
   Administrative state: enable
   Reference count:      2
   Classifier control list: A, precedence 100, parent-group X
      rate-limit-profile A
mark profile A
```

```
   Classifier control list: B, precedence 100, parent-group X
      rate-limit-profile B
mark profile B
   Classifier control list: *, precedence 100, parent-group Z
mark profile D
      forward
   Parent group: X, parent-group Z
      rate-limit-profile X
   Parent group: Z
      rate-limit-profile Z

   Referenced by interface(s):
      SERIAL4/0  input policy, statistics disabled, virtual-router default
      SERIAL4/1  input policy, statistics disabled, virtual-router default

   Referenced by profile(s):
      No profile references
```

Meaning Table 45 on page 226 lists the **show policy-list** command output fields.

Table 45: show policy-list Output Fields

Field Name	Field Description
Policy	Name of the policy list.
Administrative state	For SNMP use; state is enabled when the policy list is created. Users modifying the policy list commands via telnet see the state as disabled. Modifications of a policy are not applied to an interface until the administrative state is first disabled and then reenabled.
Reference count	Number of attachments to interfaces or profiles. An asterisk enclosed in parenthesis (*), if displayed, denotes that the policy contains inactive references to interfaces. The absence of an asterisk denotes that all the attachments of this policy to the specified number of interfaces are active.
Atm cell mode	State of mode for ATM cell tax used in rate calculations.
Referenced by interfaces	List of interfaces to which policy is attached and is active; indicates whether the attachment is at input or output of interface.
Referenced by profiles	List of profiles to which policy is attached; indicates whether the attachment is at input, secondary-input, or output of interface created by the profile.
Referenced by merge policies	List of merged policies.
Referenced by component policies	List of component policies.

Table 45: show policy-list Output Fields *(continued)*

Field Name	Field Description
Classifier control list	Name of the classifier control list containing policy rules and the precedence assigned to the classifier control list.
Statistics	Enabled, disabled
Parent group	Name of the parent group.
Rule types are:	
filter	Filter policy action
exception http-redirect	HTTP redirect policy action
forward	Forward policy action
next-interface	Next-interface policy action
next-hop	Next-hop policy action
rate-limit-profile	Rate-limit-profile policy action
color	Color of a packet; green, yellow, or red
traffic-class	Traffic class in a policy list
log	Log policy action
mark tos	ToS byte in the IP header to a specified value
mark DS field	DS field value in the IP header to a specified value
mark TC precedence	Traffic class value in the IPv6 header to a specified value
mark EXP	Value assigned to EXP bits action
mark user priority	Value assigned to 802.1p VLAN user priority bit
mark DE	DE bit action
Rule status	Indicates whether the rule is suspended.

Related Documentation

- show policy-list

Monitoring Policy List Parameters

Purpose Display information about policy list parameters.

Action To display policy list information for a hierarchical policy:

```
host1#show policy-parameter
Policy Parameter hierGroup1
    Type: hierarchical
    Reference count: 8
    Aggregation node: vlan
    Referenced by interfaces: 2 references
      IP ATM5/0.1: atm-vc
      IP ATM5/0.2: 5

    Referenced by profiles: 1 references
      profile1

    Referenced by policies: 5 references
      policy1
      policy2
      policy3
Policy Parameter hierGroup2
    Type: hierarchical
    Reference count: 3
    Aggregation node: 3
    Referenced by interfaces: 1 references
      IP ATM5/0.2: atm-vp 1

    Referenced by policies: 2 references
      policy1

    Referenced by parent groups: 1 references
      extPg1
```

To display list information:

```
host1(config)#show policy-parameter

                             Policy Parameter Table
                             ------ --------- -----
Policy Parameter refRlpRate
    Type: reference-rate
    Rate: 100000
    Reference count: 7
    Referenced by interfaces: 2 references
      IP interface ATM5/0.1: 1000000
      IP interface ATM5/0.2: 200000

    Referenced by rate-limit profiles: 5 references
      rlpData
      rlpVoice
      rlpVideo

Policy Parameter otherRate
    reference-rate: 65536
    Reference count: 3
    Referenced by interfaces: 1 references
      IP interface ATM5/0.2: 100000

    Referenced by rate-limit profiles: 2 references
      rlpOther
```


Meaning Table 46 on page 229 lists the **show policy-parameter** command output fields.

Table 46: show policy-parameter Output Fields

Field Name	Field Description
Type	Type of parameter, such as hierarchical.
Reference count	Number of references in policy, interface, and external parent group profiles.
Aggregation node	Aggregation node value.
Referenced by interfaces	List of interfaces where parameter is referenced.
Referenced by profiles	List of profiles where parameter is referenced
Referenced by policies	List of policies where parameter is referenced.
Referenced by parent groups	List of external parent groups where parameter is referenced.

Related Documentation

- show policy-parameter

Monitoring Rate-Limit Profiles

Purpose Display information about rate-limit profiles.

Action To display information about rate-limit profiles:

```
host1#show rate-limit-profile
                            Rate Limit Profile Table
                            ---- ----- ------- -----
IP Rate-Limit-Profile: rlp
   Profile Type:                   one-rate
   Reference count:                0
   Committed rate:                 0
   Committed burst:                8192
   Excess burst:                   0
   Mask:                           255
   Committed rate action:          transmit
   Conformed rate action:          transmit
   Exceeded rate action:           drop
IP Rate-Limit-Profile: rlp
   Profile Type:                   two-rate hierarchical
   Color-aware                     no
   Reference count:                0
   Committed rate:                 0
   Committed burst:                8192
   Peak rate:                      0
   Peak burst:                     8192
   Mask:                           255
   Committed rate action:          transmit unconditional
   Conformed rate action:          transmit conditional
```

```
   Exceeded rate action:          drop
L2TP Rate-Limit-Profile: L2tpRlp
   Profile Type:                  two-rate
   Reference count:               0
   Committed rate:                0
   Committed burst:               8192
   Peak rate:                     0
   Peak burst:                    8192
   Committed rate action:         transmit
   Conformed rate action:         transmit
   Exceeded rate action:          drop
```

Meaning Table 47 on page 230 lists the **show rate-limit-profile** command output fields.

Table 47: show rate-limit-profile Output Fields

Field Name	Field Description
Rate-Limit-Profile	Create a rate limit profile
Profile Name	Name of the rate-limit profile
Profile Type	One-rate, two-rate, or hierarchical profile
Reference count	Number of policy lists that reference this rate-limit profile
Color-aware	Color-aware action (yes or no) taken for profile
Committed rate	Target rate for the traffic, in bits per second
Committed burst	Amount of bandwidth allocated to accommodate bursty traffic, in bytes
Excess burst	Amount of bandwidth allocated to accommodate a packet in progress when the rate is in excess of the burst, in bytes
Peak rate	Amount of bandwidth allocated to accommodate traffic flow in excess of the committed rate, in bits per second
Peak burst	Amount of bandwidth allocated to accommodate bursty traffic in excess of the peak rate, in bytes
Mask	Value of mask applied to ToS byte in IP packet header
Committed rate action	Policy action (drop, transmit, or mark) taken when traffic flow does not exceed the committed rate

Table 47: show rate-limit-profile Output Fields *(continued)*

Field Name	Field Description
Conformed rate action	Policy action (drop, transmit, or mark) taken when traffic flow exceeds the committed rate but remains below the peak rate
Exceeded rate action	Policy action (drop, transmit, or mark) taken when traffic flow exceeds the peak rate

Related Documentation

- show rate-limit-profile

Monitoring the Policy Configuration of VLAN Subinterfaces

Purpose Display information about a subinterface's VLAN policy lists.

Action To display information about VLAN policy lists:

```
host1#show vlan subinterface fastEthernet 1/0.1
VLAN ID is 100
VLAN policy input vlanPol1
  classifier-group clac1VlanBos entry 1
    5 packets, 730 bytes
    filter
```

Meaning Table 48 on page 231 lists the **show vlan subinterface** command output fields.

Table 48: show vlan subinterface Output Fields

Field Name	Field Description
Subinterface number	Location of the subinterface that carries the VLAN traffic
VLAN ID	Domain number of the VLAN
VLAN policy	Type and name of the VLAN policy
filter	Number of packets and bytes that have been policed by the policy

Related Documentation

- show vlan subinterface

Verifying Statistics Collection for Output Policies on Tunnel Interfaces

Purpose Display whether the mechanism to generate and preserve statistics based on fragments for traffic on tunnel interfaces to which output policies are attached is enabled. By default, output policy counters for tunnel interfaces are displayed as a measure of the number of packets.

Action To determine whether collection of output policy statistics based on fragments for traffic on tunnel interfaces is enabled:

```
host1#show enable frag-stats
Enabled
```

Related Documentation

- Statistics Collection for Output Policies on Tunnel Interfaces Overview on page 18
- Configuring Statistics Collection for Output Policies on Tunnel Interfaces on page 29
- enable-frag-stats
- show enable-frag-stats

Packet Flow Monitoring Overview

The policy log rule provides a way to monitor a packet flow by capturing a sample of the packets that satisfy the classification of the rule in the system log. See *JunosE System Event Logging Reference Guide* for information about logging.

To capture the interface, protocol, source address, destination address, source port, and destination port, set the policyMgrPacketLog event category to log at severity info and at low verbosity. To capture the version, ToS, len ID, flags, time to live (TTL), protocol, and checksum in addition to the information captured at low verbosity, set the verbosity to medium or high.

When the policy is configured, all packets are examined and the matching packets are placed in the log. No more than 512 packets are logged every 3 seconds. The router maintains a count of the total number of matching packets. This count is incremental even if the packet cannot be stored in the log (for example, because the count exceeds the 512-packet threshold).

This example shows how you might use classification to specify the ingress packets that are logged in to an interface.

```
host1(config)#ip policy-list testPolicy
host1(config-policy-list)#classifier-group logA
host1(config-policy-list-classifier-group)#log
host1(config-policy-list-classifier-group)#exit
host1(config-policy-list)#exit
host1(config)#interface atm 0/0.0
host1(config-subif)#ip policy input testPolicy statistics enabled
host1(config-subif)#exit
host1(config)#log destination console severity info
host1(config)#log severity info policyMgrPacketLog
host1(config)#log verbosity low policyMgrPacketLog
host1(config)#log here
```

This example provides a more detailed procedure that an ISP might use to log information during a ping attack on the network. The procedure includes the creation of the classifier and policy lists to specify the desired packet flow to monitor, the logging of the output of the classification operation, and the output of the **show** command.

In this example, a customer has reported to their ISP that an attack is occurring on their internal servers. The attack is a simple ping flood.

1. The ISP creates a classifier list to define an ICMP echo request packet flow.

   ```
   host1:vr2(config)#ip classifier-list icmpEchoReq icmp any any 8 0
   host1:vr2(config)#ip policy-list pingAttack
   host1:vr2(config-policy-list)#classifier-group icmpEchoReq
   host1:vr2(config-policy-list-classifier-group)#log
   host1:vr2(config-policy-list-classifier-group)#exit
   host1:vr2(config-policy-list)#exit

   host1:vr2(config)#interface gigabitEthernet 2/0
   host1:vr2(config-if)#ip address 10.10.10.2 255.255.255.0
   host1:vr2(config-if)#exit

   host1:vr2(config)#virtual-router vr1
   host1:vr1(config)#interface gigabitEthernet 0/0
   host1:vr1(config-if)#ip address 10.10.10.1 255.255.255.0
   host1:vr1(config-if)#ip policy input pingAttack statistics enabled
   host1:vr1(config-if)#exit
   host1:vr1(config)#exit
   ```

2. The ISP configures standard logging on the E Series router.

   ```
   host1(config)#log destination console severity info
   host1(config)#log severity info policyMgrPacketLog
   host1(config)#log here

   INFO 12/16/2003 12:59:47 policyMgrPacketLog ():
   icmpEchoReq icmp GigabitEthernet0/0 10.10.10.2 10.10.10.1 forwarded
   INFO 12/16/2003 12:59:47 policyMgrPacketLog ():
   icmpEchoReq GigabitEthernet0/0 number of hits = 21551
   INFO 12/16/2003 12:59:50 policyMgrPacketLog ():
   icmpEchoReq icmp GigabitEthernet0/0 10.10.10.2 10.10.10.1 forwarded
   INFO 12/16/2003 12:59:50 policyMgrPacketLog ():
   icmpEchoReq GigabitEthernet0/0 number of hits = 21851
   INFO 12/16/2003 12:59:53 policyMgrPacketLog ():
   icmpEchoReq icmp GigabitEthernet0/0 10.10.10.2 10.10.10.1 forwarded
   INFO 12/16/2003 12:59:53 policyMgrPacketLog ():
   icmpEchoReq GigabitEthernet0/0 number of hits = 22151
   ```

3. The ISP displays statistics for the interface.

```
host1:vr1#show ip interface gigabitEthernet 0/0
GigabitEthernet0/0 line protocol Ethernet is up, ip is up
  Network Protocols: IP
  Internet address is 10.10.10.1/255.255.255.0
  Broadcast address is 255.255.255.255
  Operational MTU = 1500  Administrative MTU = 0
  Operational speed = 1000000000  Administrative speed = 0
  Discontinuity Time = 1092358
  Router advertisement = disabled
  Proxy Arp = enabled
  Network Address Translation is disabled
  Administrative debounce-time = disabled
  Operational debounce-time    = disabled
  Access routing = disabled
```

```
Multipath mode = hashed
Auto Configure = disabled
Auto Detect = disabled
Inactivity Timer = disabled

In Received Packets 488421, Bytes 62517888
  Unicast Packets 488421, Bytes 62517888
  Multicast Packets 0, Bytes 0
In Policed Packets 0, Bytes 0
In Error Packets 0
In Invalid Source Address Packets 0
In Discarded Packets 0
Out Forwarded Packets 486152, Bytes 62232048
  Unicast Packets 486152, Bytes 62232048
  Multicast Routed Packets 0, Bytes 0
Out Scheduler Dropped Packets 0, Bytes 0
Out Policed Packets 0, Bytes 0
Out Discarded Packets 2269

IP policy input pingAttack
  classifier-group icmpEchoReq entry 1
    488421 packets, 69355782 bytes
    log

queue 0: traffic class best-effort, bound to ip GigabitEthernet0/0
  Queue length 0 bytes
  Forwarded packets 485988, bytes 70954248
  Dropped committed packets 0, bytes 0
  Dropped conformed packets 0, bytes 0
  Dropped exceeded packets 0, bytes 0
```

You can also capture traffic that transits through the router by using the policyMgrPacketLog category. When you set the logging severity level to info, you have the following options

- interface—filter on an interface
- interface-type—filter on an interface type
- policy-list—filter on a policy list

The policy list must contain the log keyword in the classifier group you want to monitor. You must also enable logging for policyMgrPacketLog and for the specific interface or policy list.

host1(config)#**log severity info policyMgrPacketLog**
host1(config)#**log severity info policyMgrPacketLog policy-list all**
host(config)#**ip policy-list test**
host(config-policy-list)#**classifier-group ***
host1(config-policy-list-classifier-group)#**log**

host1(config)#**interface fastEthernet 2/0.100**
host1(config-if)#**vlan id 100**
host1(config-if)#**ip address 100.1.1.1 255.255.255.0**
host1(config-if)#**ip policy input test**
host1(config-if)#**ip policy output test**

The packet capture can also be done for any source and destination defined in the classifier list. If the logging verbosity is set to low, you can obtain the following level of detail from the packet capture:

```
INFO 02/20/2008 10:10:23 policyMgrPacketLog:
test icmp FastEthernet2/2.100 100.1.1.2 100.1.2.2 forwarded
INFO 02/20/2008 10:10:26 ppolicyMfrPacketLog:
test icmp FastEthernet2.2.100.100.1.2.2 100.1.1.2 forwarded
```

If the logging verbosity is set to medium or high, you can obtain the following level of detail from the packet capture:

```
INFO 02/20/2008 10:15:11 policyMgrPacketLog: Classifier: test.1, prot: icmp,
intf:  FastEthernet2/2.100, sa: 100.1.1.2, da: 100.1.2.2 version: 0x45, tos:
0x0, len: 0x3e8, id: 0x714, flags: 0x0, ttl: 0x20, proto; 0x1, chksum: 0xc4fb,
forwarded

INFO 02/20/2008 10:15:14 ppolicyMfrPacketLog: classifier: test.1, prot: icmp,
intf: FastEthernet2/2.100, sa: 100.1.1.2 da: 100.1.2.2 version: 0x45, tos:
0x0, len: 0x3e8, id: 0xbe8, flags: 0x0, ttl: 0x7e, proto; 0x1, chksum: 0x6227,
forwarded
```

Related Documentation

- Monitoring the Policy Configuration of ATM Subinterfaces on page 199
- Monitoring the Policy Configuration of Frame Relay Subinterfaces on page 204
- Monitoring the Policy Configuration of IP Interfaces on page 209

Monitoring the Detection of Corrupted FPGA Statistics Settings

Purpose Display the configuration details of the FPGA statistics detection utility.

Action To display the settings of the capability to detect corruption in the FPGA statistics

```
host1#show fpga-stats-monitoring
FPGA statistics monitoring is enabled, threshold is 40
FPGA statistics monitoring trap is enabled
```

Meaning Table 49 on page 235 lists the **show fpga-stats-monitoring** command output fields.

Table 49: show fpga-stats-monitoring Output Fields

Field Name	Field Description
FPGA statistics monitoring	Displays whether the capability to detect corruption in the FPGA statistics is enabled or disabled.
threshold	Threshold value to be used to compare the differential value between the interface and policy counters for ingress or egress policies. A corruption is detected if the difference between the interface and policy counters is equal to or higher than the threshold. The FPGA statistics is not corrupt if the difference between the interface and policy counters is less than the threshold.
FPGA statistics monitoring trap	Displays whether the capability to generate SNMP traps when corruption is identified in the FPGA is enabled or disabled.

Related Documentation

- Detection of Corruption in the FPGA Statistics for Policies of Subscribers Managed by the SRC Software on page 185
- Computation of the Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 188
- Example: Computation of the Threshold Value by Using Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 190
- Scenarios for the Detection of Corruption in the FPGA Statistics and the Determination of the Threshold on page 192
- Configuring the Capability to Detect Corruption in the FPGA Statistics for Policies Managed by the SRC Software on page 195
- System Operations When Corrupted FPGA Statistics Is Detected on page 194

PART 2

Packet Mirroring

CHAPTER 10

Packet Mirroring Overview

This chapter contains the following sections:

Packet Mirroring Overview

Packet mirroring enables you to automatically send a copy of a packet to an external host for analysis. Packet mirroring has many uses, including traffic debugging and troubleshooting user networking problems.

The JunosE Software provides two methods that you can use to configure and manage your packet-mirroring environment—CLI-based and RADIUS-based.

- CLI-based packet mirroring—An authorized operator uses the router's CLI commands to configure and manage packet mirroring. You can mirror traffic related to a specific IP, IPv6, or L2TP interface or traffic related to a particular user. You also use CLI commands to create secure policies that identify the traffic to be mirrored and specify how the mirrored traffic is treated.
- RADIUS-based packet mirroring—A RADIUS administrator uses RADIUS attributes to configure packet mirroring of a particular user's traffic. The router creates dynamic secure policies for the mirroring operation.

In both the CLI-based and the RADIUS-based packet mirroring methods, the original traffic is sent to its intended destination and the mirrored traffic is sent to an analyzer (the mediation device). The mirroring operations are transparent to the user whose traffic is being mirrored.

NOTE: Packet mirroring operations require some system resources. To avoid performance degradation, limit the amount of mirrored traffic to a maximum of 5 percent of the E Series router's total traffic.

Packet mirroring is supported on ASIC-based modules. See *ERX Module Guide*, *Appendix A*, *Module Protocol Support* for information about modules supported on ERX routers. See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about modules supported on the E120 and E320 Broadband Services Routers.

Related Documentation

- CLI-Based Packet Mirroring Overview on page 245
- Comparing CLI-Based Mirroring and RADIUS-Based Mirroring on page 240
- Monitoring Packet Mirroring Overview on page 287
- Packet-Mirroring Terms on page 242
- RADIUS-Based Mirroring Overview on page 261

Comparing CLI-Based Mirroring and RADIUS-Based Mirroring

This section compares the characteristics of CLI-based and RADIUS-based mirroring techniques. You can use CLI-based mirroring for both interface-specific and user-specific mirroring; RADIUS-based mirroring is used for user-specific mirroring. This section highlights differences in configuration, security, and application of the CLI-based and RADIUS-based mirroring methods.

Configuration

This section describes differences in the configuration processes for CLI-based and RADIUS-based mirroring:

- CLI-based packet mirroring—You use CLI commands to configure and manage packet mirroring of specific interfaces and users. For interface-specific mirroring, you enable the static configuration after the IP interface is created. The interface method mirrors only the traffic on the specific interface.

  In user-specific mirroring, authentication, authorization, and accounting (AAA) uses RADIUS attributes as triggers to identify the user whose traffic is to be mirrored. The mirroring session starts when the user logs in. If the user is already logged in, AAA immediately starts the mirroring session when you enable packet mirroring.

- RADIUS-based packet mirroring—This dynamic method uses RADIUS and vendor-specific attributes (VSAs), rather than CLI commands, to identify a user whose traffic is to be mirrored and to trigger the mirroring session. A RADIUS administrator configures and enables the mirroring separate from the user's session. You can use a single RADIUS server to provision packet-mirroring operations on multiple E Series routers in a service provider's network.

  There are two variations of RADIUS-based packet mirroring. For both types, the mirroring feature is initiated without regard to the user location, router, interface, or type of traffic.

  - User-initiated mirroring—If the user is not currently logged in, the mirroring session starts when the user logs in and is authenticated by RADIUS. The user's Acct-Session-Id is the identification trigger.
  - RADIUS-initiated mirroring—If the user is already logged in, the JunosE RADIUS dynamic-request server uses RADIUS-initiated change-of-authorization (CoA)

messages to immediately start the mirroring session when the packet mirroring is enabled.

Security

The following list highlights security features provided by CLI-based and RADIUS-based mirroring:

- CLI-based packet mirroring—All packet mirroring commands are hidden by default. You must execute the **mirror-enable** command to make the mirroring commands visible. You can optionally configure authorization methods to control access to the **mirror-enable** command, which makes the packet mirroring commands available only to authorized users. The **mirror-enable** command is in privilege level 12 by default and the mirroring commands are in privilege level 13 by default. You can change the privilege levels of these commands; however, we recommend that you always put the **mirror-enable** command at a different privilege level than the mirroring commands.
- RADIUS-based packet mirroring—Access to RADIUS-based mirroring functionality is unrestricted. However, the display of mirroring functionality is restricted to privilege level 13 users by default. In addition, the user must execute the **mirror-enable** command to make the packet mirroring-related **show** commands visible.

  RADIUS-based mirroring uses dynamically created secure policies based on certain RADIUS VSAs. You attach the secure policies to the interface used by the mirrored user. The packet-mirroring VSAs that the RADIUS server sends to the E Series router are MD5 salt-encrypted.

Application

The following list compares the different types of packet-mirroring methods:

- CLI-based packet mirroring—Is useful when organizations want to provide separation between the typical network operations personnel and the mirroring operations personnel. For example, if security is essential, you might perform the entire packet-mirroring configuration on the analyzer device, separate from the normal network operations role. This way, only the authorized personnel on the analyzer device are aware of the mirroring operation. If this level of security is not required, authorized network operations personnel can perform the configuration and management on the router as usual.
  - CLI-based interface-specific mirroring—Can be useful in small networks with few E Series routers and in static environments where a user typically logs in to the same router through the same interface.
  - CLI-based user-specific mirroring—Is useful in B-RAS environments, in which users log in and log out frequently.
- RADIUS-based user-specific mirroring—Is triggered when needed, either when the specified user logs in (user-initiated) or when the user is already logged in and RADIUS-based mirroring is enabled or modified (RADIUS-initiated). RADIUS-based mirroring also provides an excellent solution for B-RAS networks, for example to troubleshoot traffic problems related to mobile users.

CLI-based user-specific and RADIUS-based user-specific mirroring are also useful to mirror L2TP traffic at the L2TP access concentrator (LAC). If the L2TP network server (LNS) and the LAC belong to different service providers, mirroring at the LAC enables mirroring to take place close to the user's domain.

Related Documentation

- CLI-Based Packet Mirroring Overview on page 245
- RADIUS-Based Mirroring Overview on page 261
- Packet Mirroring Overview on page 239

Packet-Mirroring Terms

Table 50 on page 242 defines terms used in this discussion of packet mirroring.

Table 50: Packet-Mirroring Terminology

Term	Meaning
Analyzer device	Device that receives the mirrored traffic from the E Series router. Also called the mediation device.
Analyzer interface	IP interface in analyzer mode on the E Series router that is used to direct mirrored traffic to the analyzer device.
CLI access class	Security level that grants access to specific CLI commands.
Mediation device	Device that receives the mirrored traffic from the E Series router. Also called the analyzer device.
Mirrored interface	Statically or dynamically configured interface on which traffic is being mirrored.
Mirrored user	User whose traffic is being mirrored.
Requesting authority	Group that is authorized to request or conduct packet mirroring.
Salt encryption	Random string of data used to modify a password hash.
Secure policy	Policies created with a mirror action and that contain information about where to forward mirrored traffic.
Trigger	RADIUS attribute that identifies a user whose traffic is to be mirrored. Packet mirroring starts when a trigger is detected. An E Series router supports a maximum of 100 mirror trigger rules.

Related Documentation

- CLI-Based Packet Mirroring Overview on page 245
- Packet Mirroring Overview on page 239
- Packet Mirroring Platform Considerations on page 243
- Packet Mirroring References on page 243

- RADIUS-Based Mirroring Overview on page 261

Packet Mirroring Platform Considerations

For information about modules that support packet mirroring on ERX14xx models, ERX7xx models, and the ERX310 Broadband Services Router:

- See *ERX Module Guide, Table 1, Module Combinations* for detailed module specifications.
- See *ERX Module Guide, Appendix A, Module Protocol Support* for information about the modules that support packet mirroring.

For detailed information about the modules that support packet mirroring on the E120 and E320 Broadband Services Router:

- See *E120 and E320 Module Guide, Table 1, Modules and IOAs* for detailed module specifications.
- See *E120 and E320 Module Guide, Appendix A, IOA Protocol Support* for information about the protocols and applications that support packet mirroring.

Related Documentation

- Packet Mirroring References on page 243

Packet Mirroring References

For more information about RADIUS-based packet mirroring, consult the following resources:

- RFC 3576—Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS) (July 2003)
- Lawfully Authorized Electronic Surveillance (LAES) for IP Network Access, American National Standard for Telecommunications, version PTSC-LAES-2006-084R6

Related Documentation

- Packet Mirroring Platform Considerations on page 243

CHAPTER 11

Configuring CLI-Based Packet Mirroring

Packet mirroring enables you to send a copy of a packet to an external host for analysis. Packet mirroring has many uses, including traffic debugging and troubleshooting user networking problems.

This chapter contains the following sections:

CLI-Based Packet Mirroring Overview

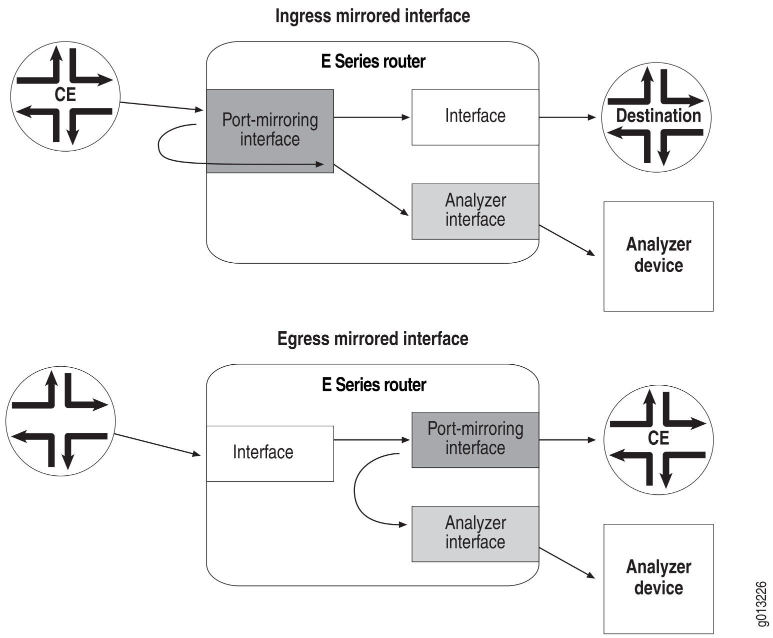
The JunosE Software enables you to use CLI commands to configure and manage packet mirroring on specific static IP interfaces, or for a specific user. You use CLI commands to create a secure policy that specifies the analyzer device and how the mirrored traffic is treated.

When you mirror an interface, you can replicate ingress and egress traffic on the interface (traffic entering or exiting the E Series router through that interface). When you mirror a user, you can replicate all traffic to or from the user.

In both interface-specific and user-specific mirroring, the original traffic is forwarded to its intended destination as usual, while the replicated copy of the traffic is forwarded to an analyzer interface on the E Series router. The analyzer interface then directs the mirrored traffic to the specified analyzer device for analysis.

Figure 19 on page 246 shows the traffic flow for ingress and egress IP interface mirroring.

Figure 19: CLI-Based Interface Mirroring



Related Documentation

- CLI-Based Packet Mirroring Sequence of Events on page 250
- Comparing CLI-Based Mirroring and RADIUS-Based Mirroring on page 240
- Configuring CLI-Based Packet Mirroring on page 251
- Monitoring CLI-Based Packet Mirroring on page 288
- Packet Mirroring Overview on page 239
- Reloading a CLI-Based Packet-Mirroring Configuration on page 248

Enabling and Securing CLI-Based Packet Mirroring

The JunosE Software enables you to create a secure environment for your packet-mirroring operation by restricting access to the packet mirroring CLI commands and information. For example, when dealing with a critical diagnostic or troubleshooting procedure, you might want the packet-mirroring feature to be available and visible to a subset of your network operations group. Or, if you are monitoring confidential traffic from a particular user, you might want the configuration and results of the mirroring operation to be available only to a unique group, such as the management group of the analyzer device.

By default, the packet mirroring configuration commands are hidden from all users. You must use the **mirror-enable** command to make the commands visible, which then enables you to configure the packet-mirroring environment. The command applies only to the current CLI session. When you log out of the current session and then log in again, the packet mirroring commands are no longer visible, NOTE: The **no mirror-enable** command makes the packet mirroring commands no longer visible. However, any active mirroring sessions are unaffected and traffic continues to be mirrored.

To create a secure packet-mirroring environment, you use a combination of the JunosE Software authorization methods and the **mirror-enable** command. You configure the authorization method to control who can use the **mirror-enable** command. Authorized users can then issue the **mirror-enable** command, making the packet mirroring commands visible. However, the commands are still hidden from unauthorized users. Table 51 on page 247 lists the commands whose visibility is controlled by the **mirror-enable** command.

Table 51: Commands Made Visible by the mirror-enable Command

• ip policy { secure-input | secure-output }	• secure ipv6 policy-list
• show ip interface (packet mirroring information)	• ipv6 policy { secure-input | secure-output }
• clear mirror log	• show ipv6 interface (packet mirroring information)
• mirror acct-session-id	• show mirror log
• mirror agent-circuit-id	• show mirror rules
• mirror agent-remote-id	• show mirror trap
• mirror analyzer-ip-address	• show mirror subscribers
• mirror calling-station-id	• show secure classifier-list
• mirror dhcp-option-82	• show secure policy-list
• mirror disable	• show snmp secure-log
• mirror ip-address	• show snmp trap (packet mirroring information)
• mirror nas-port-id	• snmp-server clear secure-log
• mirror trap-enable	• snmp-server secure-log
• mirror username	• snmp-server enable traps (packetMirror keyword)
• secure ip classifier-list	• snmp-server host (packetMirror keyword)
• secure ip policy-list	• secure ipv6 classifier-list
• secure l2tp policy-list	

To provide increased security, the **mirror-enable** command must be the only command at its access level (level 12 by default) and it also must be at a different privilege level than the other packet mirroring commands (level 13 by default) and other regular JunosE CLI commands. This separation enables you to control authorization to the **mirror-enable** command and to limit the visibility of packet mirroring commands. For example, if you are using TACACS+, the **mirror-enable** command is the only packet mirroring command that is sent to the TACACS+ server. You can also use TACACS+ to prevent unauthorized individuals from modifying the configuration of analyzed ports.

See *Chapter 7, Passwords and Security* in *JunosE System Basics Configuration Guide* for more information about access levels and *Chapter 9, Configuring TACACS+* in *JunosE Broadband Access Configuration Guide* for information about TACACS+ authorization.

Related Documentation

- CLI-Based Packet Mirroring Overview on page 245
- CLI-Based Packet Mirroring Sequence of Events on page 250
- Configuring CLI-Based Packet Mirroring on page 251
- Reloading a CLI-Based Packet-Mirroring Configuration on page 248

Reloading a CLI-Based Packet-Mirroring Configuration

You can reload your packet mirroring configuration as part of a configuration file (.cnf) reload operation or when you run a script file (.scr) that you have saved from the **show configuration** command display. When you reload a .cnf file, the packet-mirroring configuration is restored—no additional steps are required.

For a .scr file operation, the **mirror-enable** command must be enabled both before saving the scr. file from the **show configuration** display and also before you run the script to reload the packet-mirroring configuration. If the **mirror-enable** command is not enabled, the .scr file operation for the packet-mirroring configuration fails.

Related Documentation

- Enabling and Securing CLI-Based Packet Mirroring on page 246
- CLI-Based Packet Mirroring Overview on page 245
- CLI-Based Packet Mirroring Sequence of Events on page 250
- Configuring CLI-Based Packet Mirroring on page 251

- mirror-enable
- show configuration

Using TACACS+ and Vty Access Lists to Secure Packet Mirroring

This procedure uses TACACS+ and vty access lists to manage the users who have access to the **mirror-enable** command. An authorized user who issues the **mirror-enable** command then gains access to the packet mirroring CLI commands and information.

This technique enables you to restrict the visibility and use of packet mirroring commands to a controlled, authorized group of users.

1. Configure TACACS+ authorization for the access level of the **mirror-enable** command (level 12 by default).

   Configure the router either to allow or disallow authorization when the TACACS+ servers are not available.

2. Configure all vty lines and the console to use the TACACS+ authorization configuration from Step 1 for access level 12 commands.

This procedure ensures that packet mirroring commands are never sent out of the E Series router—only the **mirror-enable** command is sent. The packet mirroring configuration and all information about mirrored interfaces and subscribers are available only to users who are authorized for the packet mirroring CLI commands on the router.

Related Documentation

- CLI-Based Packet Mirroring Overview on page 245
- Configuring CLI-Based Packet Mirroring on page 251
- Using Vty Access Lists to Secure Packet Mirroring on page 249
- mirror-enable

Using Vty Access Lists to Secure Packet Mirroring

In this procedure, TACACS+ authorization is not used. However, you can still use vty access lists to control access to the **mirror-enable** command, which enables you to create isolation between the authorized packet mirroring users and unauthorized network operators.

1. Configure TACACS+ authorization for the **mirror-enable** command privilege level. Specify that authorization is denied if TACACS+ is not available. Because TACACS+ is not being used, authorization always fails.
2. Configure the *majority* of the vty lines and the console to use the authorization configuration from Step 1. (Users who use Telnet on these lines are denied access to the **mirror-enable** command.)
3. On the remaining vty lines (without the TACACS+ authorization) create an access list that contains the IP addresses of the users that you want to grant access to these vty lines—these users are granted access to the **mirror-enable** command, and therefore, the packet-mirroring feature.

This configuration grants access to the packet mirroring CLI commands to the users from the specified IP addresses. The packet mirroring commands remain hidden for all other users.

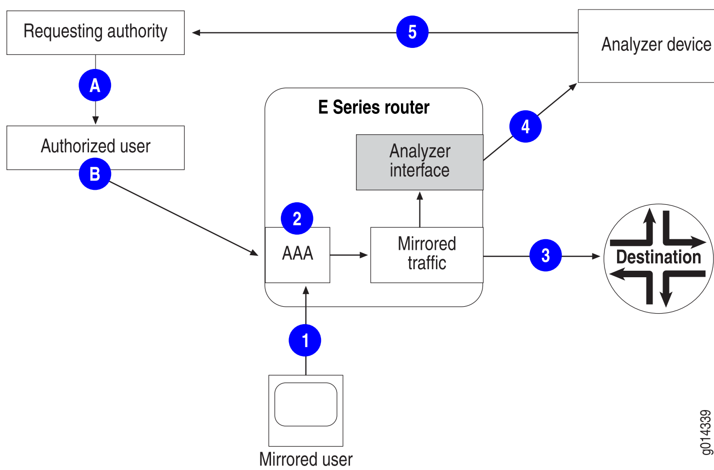
Related Documentation

- CLI-Based Packet Mirroring Overview on page 245
- Configuring CLI-Based Packet Mirroring on page 251
- Using TACACS+ and Vty Access Lists to Secure Packet Mirroring on page 248
- mirror-enable

CLI-Based Packet Mirroring Sequence of Events

Figure 20 on page 250 shows the sequence of events that take place during CLI-based mirroring. The tables after the figure describe the events indicated by the numbers and letters in the figure. Table 52 on page 250 describes the configuration process; Table 53 on page 250 describes the flow of traffic during a mirroring operation that is initiated when the user logs in; and Table 54 on page 251 describes the flow of traffic when mirroring a user who is already logged in or when mirroring a static interface.

Figure 20: CLI-Based Packet Mirroring



To create a CLI-based packet mirroring environment, you must complete the processes listed in Table 52 on page 250.

Table 52: Setting Up the CLI-Based Packet-Mirroring Environment

Process	Description
A	The authorized individual requests packet mirroring of a user's or interface's traffic and configures the analyzer device to receive mirrored traffic.
B	An individual who is authorized to use the packet mirroring CLI commands configures the packet mirroring environment, including the secure policy, analyzer interface connection to the analyzer device, and the interface or trigger information.

Table 53 on page 250 indicates the sequence of steps for a packet-mirroring operation that takes place when a user starts a new session.

Table 53: CLI-Based User-Specific Mirroring During Session Start

Step	Description
1	The user logs in to an E Series router, requesting authentication by AAA.
2	AAA authenticates the user, and the router starts mirroring the user's traffic.

Table 53: CLI-Based User-Specific Mirroring During Session Start *(continued)*

Step	Description
3	The router sends the user's original traffic to the intended destination.
4	The router sends the mirrored traffic to the analyzer device.
5	The analyzer device provides information to the requesting individual.

Table 54 on page 251 indicates the sequence of steps for a packet-mirroring operation that is configured for an interface or for a user who is already logged in.

Table 54: CLI-Based Mirroring of Currently Running Session

Step	Description
1	For user-specific mirroring, the user logs in to the E Series router; no mirroring action is configured.
2	• CLI-based packet mirroring is configured and enabled on the router. • For interface-specific mirroring, the router starts mirroring all traffic for the interface. • For user-specific mirroring, AAA verifies that the mirrored user is already logged in, then starts mirroring all subsequent traffic to or from the user.
3	The router sends the original traffic to its intended destination.
4	The router sends mirrored traffic to the analyzer device.
5	The analyzer device provides information for the requesting individual.

Related Documentation

- Enabling and Securing CLI-Based Packet Mirroring on page 246
- Configuring CLI-Based Packet Mirroring on page 251
- Reloading a CLI-Based Packet-Mirroring Configuration on page 248

Configuring CLI-Based Packet Mirroring

To configure the CLI-based packet-mirroring environment, you must coordinate the mirroring operations of two devices in the network: the E Series router and the analyzer device. The configuration of the analyzer device is mentioned in this section for reference only. The actual configuration procedures depend on the policies and guidelines established by the responsible organizations.

The **secure ip policy** and **secure ipv6 policy** commands are visible only to authorized users; the **mirror-enable** command must be enabled before using **secure ip policy** or **secure ipv6 policy** command. If you enter the **secure ip policy** or **secure ipv6 policy** command and the policy list does not exist, the router creates a policy list with a default mirror rule that disables mirroring. If you attach this policy list to an interface, there is no packet mirroring. When you use this command to create a secure policy list, statistics-related keywords are not supported.

The **secure ip classifier-list** command creates or modifies a secure IP classifier control list, which can then be included in a secure policy list.

The **secure ipv6 classifier-list** command creates or modifies a secure IPv6 classifier control list, which can then be included in a secure policy list.

NOTE: Do not use the asterisk (*) for the name of a classifier list. The asterisk is used as a wildcard for the **classifier-group** command.

Except for the following considerations, secure IP classifier lists are created and function the same as standard IP classifier lists—see "Classifier Control Lists Overview" on page 9 for information:

- The **secure ip classifier-list** and **secure ipv6 classifier-list** commands are visible only to authorized users—the **mirror-enable** command must be enabled before using this command.
- Secure IP classifier lists and secure IPv6 classifier lists are the only types of classifier lists allowed in secure policy lists
- Secure IP classifier lists and secure IPv6 classifier lists cannot be used in non-secure policy lists.
- You can associate secure IP and secure IPv6 policy classifier lists with all secure IP and secure IPv6 policies dynamically created by RADIUS. This allows you to selectively identify and drop high load traffic, such as video.

The **secure ip policy-list**, **secure ipv6 policy-list**, and **secure l2tp policy-list** commands create or modify a secure IP, IPv6, or L2TP policy list. These commands are visible only to authorized users—the **mirror-enable** command must be enabled before using this command. These commands enter Policy List Configuration mode, enabling you to specify the parameters of the secure policy list. If you enter Policy List Configuration mode and then type **exit** without specifying any parameters, the router creates a policy list with a mirror disable rule. Attaching this policy list to an interface results in no packet mirroring.

Secure IP classifier lists are the only type of classifier lists allowed in secure IP policy lists. Secure L2TP policies do not support classification. Therefore, the only classifier group you can use for secure L2TP policies is **classifier-group ***. You cannot delete a secure policy list that is currently attached to an interface.

Related Documentation

- Enabling and Securing CLI-Based Packet Mirroring on page 246
- CLI-Based Packet Mirroring Sequence of Events on page 250
- Reloading a CLI-Based Packet-Mirroring Configuration on page 248

- classifier-group
- ip analyzer

- ip mirror
- ip policy
- mirror
- mirror analyzer-ip-address
- mirror disable
- mirror disable
- secure ip classifier-list
- secure ipv6 classifier-list
- secure ip policy-list
- secure ipv6 policy-list
- secure l2tp policy-list

Configuring Triggers for CLI-Based Mirroring

In user-specific packet mirroring, you use triggers to identify the user whose traffic you want to mirror and to start the mirroring session. The triggers are similar to the RADIUS attributes used in RADIUS-based mirroring. However, for CLI-based mirroring, AAA can use any supported authentication method, including RADIUS.

NOTE: An E Series router supports a maximum of 100 mirror trigger rules.

Attributes associated with users are examined in the following order of priority to find a match. When a match is found, examination stops.

1. Account session ID
2. Calling station ID
3. Username and virtual router ID
4. IP address and virtual router ID
5. Nas-Port-Id

You specify the triggers with the **mirror** command, except that the virtual router associated with username or IP address is taken from the VR context from which you issue the command.

The following considerations apply to trigger rules:

- A new trigger rule is not applied to matching connected subscribers if any of the subscribers is mirrored by another rule.
- CLI-initiated mirroring per account session ID creates a rule that continues to exist after the subscriber logs out.

- RADIUS CoA messages affect only currently connected subscribers; they do not create persistent rules.

Related Documentation

- Configuring CLI-Based Packet Mirroring on page 251
- Example: Configuring CLI-Based Interface-Specific Packet Mirroring on page 255
- Example: Configuring CLI-Based User-Specific Packet Mirroring on page 257

Configuring the Analyzer Device

The analyzer device must be configured to receive the mirrored traffic from the E Series router's analyzer interface. You can use the **default** keyword with the **interface** command to configure an interface as the virtual router's default analyzer interface; it is then used when an analyzer interface is not explicitly specified in the **ip mirror** command. You cannot configure multiaccess interfaces, such as IP over Ethernet, as default analyzer interfaces.

You can configure any type of IP interface on the E Series router as an analyzer interface, except for special interfaces such as SRP interfaces, null interfaces, and loopback interfaces. An interface cannot be both an analyzer interface and a mirrored interface at the same time. A single analyzer interface can serve multiple mirrored sessions. Analyzer interfaces drop all nonmirrored traffic.

You can configure IP or GRE analyzer interfaces to enable traffic to flow between tunnel endpoints that are local to the router. The tunnel can be located on a shared tunnel server port or line module. For a complete list of the line modules and I/O modules available for ERX14xx models, ERX7xx models, and the ERX310 Broadband Services Router, see *ERX Module Guide*. For more information about line modules and IOAs available with the E120 and E320 Broadband Services Routers, see *E120 and E320 Module Guide*.

Shared tunnel server on the ES2 10G ADV LM supports GRE tunnels for tunneling the mirrored data packets. The mirrored data is forwarded to the analyzer device using the GRE analyzer interface. Use the **ip analyzer** command to configure the GRE tunnel interface to act as a GRE analyzer tunnel interface. The ES2 10G ADV LM does not support non-analyzer tunnel interfaces. Also, when you configure a GRE interface for checksum calculations, use of sequence numbers, session keys, and other optional parameters, the ES2 10G ADV LM does not support those GRE interfaces. However, if you have configured a non-analyzer tunnel interface or a GRE interface with optional parameters, these interfaces remain non-operational. The GRE analyzer interface forwards mirrored traffic and drops all non-mirrored traffic.

Also, placement of GRE tunnels on the supported locations is no longer synchronous with the tunnel configuration. So, you can configure tunnel servers when the chassis does not support the required resources such as shared tunnel server ports or tunnel server modules. However, the tunnels configured are non-operational. The tunnels become operational when the required resources are added to the chassis.

NOTE: If a chassis has shared or dedicated tunnel server on the ES2 4G LM and shared tunnel server on the ES2 10G ADV LM, the GRE non-analyzer tunnel interfaces are available on the ES2 4G LM. Only GRE analyzer interfaces with no optional configurations are available on the ES2 10G ADV LM shared tunnel server.

Policies are not supported on analyzer interfaces. When you configure an analyzer interface, existing policies are disabled, and no new policies are accepted.

Related Documentation

- CLI-Based Packet Mirroring Sequence of Events on page 250
- Configuring CLI-Based Packet Mirroring on page 251
- Resolving and Tracking the Analyzer Device's Address on page 276

- ip analyzer
- ip mirror

Configuring the E Series Router to Support CLI-Based Mirroring

To configure the router to support CLI-based packet mirroring:

1. Configure the analyzer interface, the route to the analyzer device, and any static ARP entries.
2. Allow authorized users to have access to the **mirror-enable** command. The users can then make the packet mirroring CLI commands visible and perform the following steps.
3. Configure the secure policy that forwards the mirrored traffic to the analyzer device.
4. (Optional) For increased security, create an IPSec tunnel between the analyzer interface and the analyzer device.
5. For interface-specific mirroring, attach the secure policy to the interface.
6. For user-specific mirroring, configure the trigger that identifies the user.

Related Documentation

- CLI-Based Packet Mirroring Sequence of Events on page 250
- Configuring CLI-Based Packet Mirroring on page 251

- mirror-enable

Example: Configuring CLI-Based Interface-Specific Packet Mirroring

This example shows the configuration of a CLI-based packet mirroring session for a particular static IP interface. The configuration results in all traffic through the interface being replicated and the replicated traffic then sent through an IPSec tunnel to the analyzer device.

1. Enable the visibility and use of the packet mirroring CLI commands.

   host1#**mirror-enable**

2. Configure the analyzer interface and a route to reach the analyzer device at 192.168.125.29.

   NOTE: If the analyzer interface is Ethernet-based, you must configure a static ARP entry for the analyzer device.

   host1(config)#**virtual-router vr1**
   host1:vr1(config)#**interface tunnel ipsec:Diag transport-virtual-router default**
   host1:vr1(config-if)#**ip analyzer**
   host1:vr1(config-if)#**exit**
   host1:vr1(config)#**ip route 192.168.125.29 255.255.255.255 tunnel ipsec:Diag**

3. Configure the secure IP policy that forwards the mirrored traffic to the analyzer device at 192.168.125.29.

   In this example, the configured mirror rule does not include the **analyzer-udp-port** keyword. Therefore, the rule sets the mirror header to **disable**, which means that the mirror header is not prepended to the mirrored packets. See "Understanding the Prepended Header During a Packet Mirroring Session" on page 273 for information about the prepended mirror header. The **classifier-group** command uses a previously configured classifier list, secClassA.

   host1:vr1(config)#**secure ip policy-list secureIpPolicy1**
   host1:vr1(config-policy-list)#**classifier-group secClassA**
   host1:vr1(config-policy-list-classifier-group)#**mirror analyzer-ip-address 192.168.125.29 analyzer-virtual-router vr1**

4. Attach the secure policy to the interfaces whose traffic you want to mirror. This example mirrors input traffic at interface ATM 5/0.1 and output traffic at interface ATM 5/0.2.

   host1:vr1(config)#**interface atm 5/0.1**
   host1:vr1(config-if)#**ip policy secure-input secureIpPolicy1**

   host1:vr1(config)#**interface atm 5/0.2**
   host1:vr1(config-if)#**ip policy secure-output secureIpPolicy1**

5. Verify the secure policy configuration.

```
host1# show secure policy-list name secureIpPolicy1
                                  Policy Table
                                  ------ -----
Secure IP Policy secureIpPolicy1
 Administrative state: enable
 Reference count:      2
 Classifier control list: secClassA
  mirror analyzer-ip-address 192.168.125.29 analyzer-virtual-router vr1
 Referenced by interface(s):
  ATM5/0.1  secure-input policy, virtual-router vr1
  ATM5/0.2  secure-output policy, virtual-router vr1
```


Related Documentation

- Configuring CLI-Based Packet Mirroring on page 251
- Configuring Triggers for CLI-Based Mirroring on page 253
- Example: Configuring CLI-Based User-Specific Packet Mirroring on page 257

Example: Configuring CLI-Based User-Specific Packet Mirroring

This example shows the configuration of a CLI-based packet mirroring session for subscribers. . The mirroring session replicates all traffic associated with each user, and then sends the replicated traffic to the analyzer device.

1. Enable the visibility and use of the packet mirroring CLI commands.

   ```
   host1#mirror-enable
   ```

2. Create the analyzer interface and the route to the analyzer device.

   - For L2TP subscribers:

     ```
     host1(config)# interface tunnel ipsec:mirror3 transport-virtual-router default
     host1(config-if)#ip analyzer
     host1(config-if)#exit
     host1(config)#ip route 192.168.99.2 255.255.255.255 tunnel ipsec:mirror3
     ```

   - For DHCP and PPP subscribers:

     ```
     host1(config)# interface atm 4/0.1
     host1(config-if)#ip address 19.0.0.2 255.255.255.0
     host1(config-if)#ip analyzer
     host1(config-if)#exit
     host1(config)#ip route 19.0.0.2 255.255.255.255 101.101.101.2
     ```

3. Configure the secure policy that forwards the mirrored traffic to the analyzer device. The **classifier-group** command uses the default classifier list, which is indicated by the asterisk character (*****).

   - For L2TP subscribers:

     ```
     host1(config)#secure l2tp policy-list l2tp_toMirrorHQ
     host1(config-policy-list)#classifier-group *
     host1(config-policy-list-classifier-group)#mirror analyzer-ip-address 192.168.99.2
       analyzer-virtual-router default analyzer-udp-port 6500 mirror-identifier 1
       session-identifier 1
     ```

   - For DHCP and PPP subscribers:

     ```
     host1(config)#secure ip policy-list secure-ipv4-policy
     host1(config-policy-list)#classifier-group *
     host1(config-policy-list-classifier-group)#mirror analyzer-ip-address 19.0.0.2
       analyzer-virtual-router default analyzer-udp-port 2500 mirror-identifier 1
       session-identifier 1
     ```

4. Configure packet mirroring for the subscriber and associate the secure policy with the user.

   - For L2TP subscribers:

host1(config)#**virtual-router lac**
host1:lac(config)#**mirror username jwbooth@isptheatre.com l2tp secure-policy-list l2tp_toMirrorHQ**

- For DHCP and PPP subscribers:

host1(config)#**mirror dhcp-option-82 agent-circuit-id "x:12000004:circuit id:45" agent-remote-id "y:12000004:remote id:89" ip secure-policy-list secure-ipv4-policy**
host1(config)#**mirror agent-circuit-id "x:12000001:pppoe agent circuit id:47" ip secure-policy-list secure-ipv4-policy**
host1(config)#**mirror agent-remote-id hex 79:3a:02:00:00:02:3a:72:65:6d:6f:74:65:20:69:64:3a:35 ip secure-policy-list secure-ipv4-policy**

Now, when the subscriber logs in , the packet mirroring session starts and the subscriber's replicated traffic is sent to the remote analyzer device.

5. Verify the packet-mirroring configuration.

```
host1# show mirror subscribers
Subscriber ID                                         ID         Secure
  Secure        Mirrored
                                                    Method       Policy Type
  Policy List    Sessions
------------------                                ----------   -----------
----------------  --------
lac:jwbooth@isptheatre.com                           username        l2tp
 l2tp_toMirrorHQ    1
x:12000004:circuit id:45.y:12000004:remote id:89     dhcp-option-82   IP
 secure-ipv4-policy 1
x:12000001:pppoe agent circuit id:47                 agent-circuit-id IP
 secure-ipv4-policy 1
79:3a:02:00:00:02:3a:72:65:6d:6f:74:65:20:69:64:3a:35 agent-remote-id  IP
 secure-ipv4-policy 1
```

6. Verify the configuration of the secure policy.

```
host1# show secure policy-list
                                  Policy Table
                                  ------ -----
Secure L2TP Policy l2tp_toMirrorHQ
 Administrative state: enable
 Reference count:      2
 Classifier control list: *
  mirror analyzer-ip-address 192.168.99.2 analyzer-virtual-router default
analyzer-udp-port 6500 mirror-id 1 session-id 1
 Referenced by interface(s):
  TUNNEL l2tp:5/1/5  secure-input policy
  TUNNEL l2tp:5/1/5  secure-output policy

Secure IP Policy secure-ipv4-policy
  Administrative state: enable
  Reference count:      6
  Classifier control list: *
mirror analyzer-ip-address 19.0.0.2 analyzer-virtual-router default
analyzer-udp-port 2500 mirror-identifier 1 session-identifier 1
Referenced by interface(s):
 ip100.1.1.3  secure-input policy, statistics disabled, virtual-router default
```



```
ip100.1.1.3  secure-output policy, statistics disabled, virtual-router default
```

Related Documentation

- Configuring CLI-Based Packet Mirroring on page 251
- Configuring Triggers for CLI-Based Mirroring on page 253
- Example: Configuring CLI-Based Interface-Specific Packet Mirroring on page 255

CHAPTER 12

Configuring RADIUS-Based Mirroring

Packet mirroring enables you to send a copy of a packet to an external host for analysis. Packet mirroring has many uses, including traffic debugging and troubleshooting user networking problems.

This chapter contains the following sections:

RADIUS-Based Mirroring Overview

RADIUS-based packet mirroring enables you to mirror traffic related to a specific user, without regard to how often the user logs in or out, or which E Series router or interface the user uses. RADIUS-based mirroring is particularly appropriate for large networks, because you can use a single RADIUS server to provision mirroring on multiple E Series routers in a service provider's network. RADIUS-based mirroring is useful when debugging network problems related to mobile users, who do not always log in to a particular router.

You configure RADIUS-based mirroring independent of the actual mirroring session—you can configure the mirroring parameters at any time. RADIUS-based mirroring uses RADIUS and VSAs, rather than CLI commands, to specify the user whose traffic is to be mirrored. The VSAs specify attributes that are carried in Access-Accept messages and change-of-authorization messages from the RADIUS dynamic-request server to the E Series router.

NOTE: You cannot use RADIUS-based packet mirroring to mirror static interfaces, which might not be authenticated through RADIUS. To mirror static interfaces, you must use CLI-based mirroring.

Related Documentation

- Comparing CLI-Based Mirroring and RADIUS-Based Mirroring on page 240
- Configuring RADIUS-Based Packet Mirroring on page 264
- Packet Mirroring Overview on page 239
- RADIUS-Based Mirroring Sequence of Events on page 266
- RADIUS Attributes Used for Packet Mirroring on page 262

RADIUS Attributes Used for Packet Mirroring

Table 55 on page 262 and Table 56 on page 262 list the packet mirroring triggers. The triggers are RADIUS attributes that identify a user whose traffic is to be mirrored. A packet mirroring session starts when the router receives a RADIUS packet that contains mirroring attributes and then applies the mirroring configuration to the appropriate interface. For example, packet mirroring starts when a logon request occurs that contains a specified User-Name attribute.

The triggers also enable RADIUS-initiated mirroring to start when the user is already logged in.

Table 55: RADIUS Attributes Used as Packet Mirroring Triggers (Vendor ID 4874)

Standard Number	Attribute Name	Order of Preference
[1]	User-Name	4
[8]	Framed-IP-Address	3
[26-1]	Virtual-Router	Used with Framed-IP-Address and User-Name
[31]	Calling-Station-ID	2
[44]	Acct-Session-ID	1
[87]	Nas-Port-ID	5
[26–159]	DHCP- Option-82	6

Table 56: RADIUS Attributes Used as Packet Mirroring Triggers (Vendor ID 3561)

Standard Number	Attribute Name	Order of Preference
[26-1]	Agent-Circuit-ID	7
[26-2]	Agent-Remote-ID	8

You add the trigger to the RADIUS record of the user whose traffic will be mirrored. In addition, you must include the RADIUS VSAs listed in Table 57 on page 263 in the mirrored user's RADIUS record.

NOTE: For IP mirroring, you must include both VSA 26-59 and VSA 26-61, or you must omit both of these VSAs. If you use only one of these VSAs, the configuration fails.

Table 57: RADIUS-Based Mirroring Attributes

Standard Number	Attribute Name	Setting
[26-58]	LI-Action	0 = disable mirroring 1 = enable mirroring 2 = no action
[26-59]	Med-Dev-Handle	String (not null-terminated)
[26-60]	Med-IP-Address	IP address of analyzer device
[26-61]	Med-Port-Number	UDP port number of monitoring application in analyzer device

An LI-Action setting of 2 specifies that the router does not perform any packet mirroring–related configuration. This setting can provide additional security by confusing unauthorized users who attempt to access packet mirroring communication between the router and the RADIUS server.

Related Documentation

- RADIUS-Based Mirroring Overview on page 261
- RADIUS-Based Mirroring Sequence of Events on page 266

RADIUS-Based Packet Mirroring Dynamically Created Secure Policies

RADIUS-based packet mirroring uses dynamically created secure policies, which are based on the RADIUS VSAs that an authorized RADIUS administrator creates. A policy is created when the packet mirroring action is initiated at the RADIUS server, and then applied to the interface that is dynamically created for the user. When the mirroring operation is disabled, the secure policy is deleted.

The E Series router creates a name for the dynamically created policies—the name consists of the string spl followed by a hexadecimal integer, such as spl_88000008. The name is displayed by the **show secure policy-list** command.

Related Documentation

- RADIUS-Based Mirroring Overview on page 261
- RADIUS-Based Mirroring Sequence of Events on page 266
- show secure policy-list

RADIUS-Based Packet Mirroring MLPPP Sessions

When you use RADIUS-based packet mirroring on MLPPP traffic, RADIUS authentication and authorization is performed on the individual links. The mirroring-related VSAs are returned with the RADIUS response. For user-initiated mirroring, which starts when the user logs in, a RADIUS response is returned for each successful authentication/authorization. For RADIUS-initiated mirroring of a user who is already logged in, a single RADIUS request is sent for each link.

- If you are mirroring an L2TP session, the packet-mirroring operation is enabled or disabled on a single link that is uniquely identified by the trigger you use (the RADIUS attributes for Acct-Session-ID or User-Name). For tunneled MLPPP, the individual links in the MLPPP bundle are mirrored separately. The packet-mirroring configuration fails if you use the Acct-Multi-Session-ID attribute (RADIUS attribute 50) for the configuration.
- If you are mirroring an IP session, the packet-mirroring operation is enabled or disabled on the MLPPP bundle as a whole. We recommend that you use the Account-Session-ID RADIUS attribute rather than the User-Name attribute as the trigger. Using the Account-Session-ID attribute is more efficient because the JunosE Software creates one secure policy that packet mirroring uses for all links in the MLPPP bundle. If you use the User-Name attribute, a secure policy is created for the first link, then removed and re-created for every other link.

Related Documentation

- Configuring RADIUS-Based Packet Mirroring on page 264
- RADIUS-Based Mirroring Overview on page 261
- RADIUS-Based Mirroring Sequence of Events on page 266

Configuring RADIUS-Based Packet Mirroring

To configure the RADIUS-based packet mirroring environment, you must coordinate the mirroring operations of three devices in the network: the RADIUS server, the E Series router, and the analyzer device. The configuration of the RADIUS server and the analyzer device is described in this section for reference only. The actual configuration procedures depend on the policies and guidelines established by the responsible organizations.

Configuring the RADIUS Server

Table 57 on page 263 lists the VSAs that are included for both types of RADIUS-based mirroring—user-initiated (when the user logs in to start a new session), and RADIUS-initiated (when the user is already logged in).

Disabling RADIUS-Based Mirroring

To disable mirroring, you include the RADIUS attribute (for example, Acct-Session-ID) and set the Mirror-Action attribute to 0 in the mirrored user's RADIUS record.

You can also use the **mirror disable** CLI commands to disable RADIUS-based mirroring. You must use the version of the **mirror disable** command that corresponds to the RADIUS attribute that was used to identify the user. For example, if you used the RADIUS Calling-Station-ID attribute to create the mirroring session, you must use the **mirror disable calling-station-id** command to disable the session.

NOTE: All RADIUS-based mirroring sessions that start when a user logs in are considered to use the Acct-Session-ID attribute. Therefore, you must use the **mirror disable acct-session-id** command to disable these sessions. For RADIUS-based sessions of a user that is already logged in, you use the **mirror disable** command with the same keyword you used to configure the session.

Configuring the Analyzer Device

The analyzer device must be configured to receive the mirrored traffic from the E Series router's analyzer interface. The analyzer interface directs mirrored traffic to the specified analyzer device for analysis. You can configure the interface as the virtual router's default analyzer interface. You cannot configure multiaccess interfaces, such as IP over Ethernet, as default analyzer interfaces.

When mirroring an IP interface, the analyzer interface must reside in the same virtual router as the mirrored interface. When mirroring an L2TP interface, the analyzer interface must reside in the default virtual router.

NOTE: You must configure a static route to reach the analyzer device through the analyzer interface. If the analyzer interface is an IP over Ethernet interface, you must also configure a static Address Resolution Protocol (ARP) entry to reach the analyzer device.

You can configure any type of IP interface on the E Series router as an analyzer interface, except for special interfaces such as SRP interfaces, null interfaces, and loopback interfaces. An interface cannot be both an analyzer interface and a mirrored interface at the same time. A single analyzer interface can support multiple mirrored interfaces. The receive side of the analyzer interface is disabled. All traffic attempting to access the router through an analyzer interface is dropped. Analyzer interfaces drop all nonmirrored traffic. Policies are not supported. When you configure an analyzer interface, existing policies are disabled, and no new policies are accepted.

Related Documentation

- RADIUS-Based Mirroring Overview on page 261
- RADIUS-Based Mirroring Sequence of Events on page 266

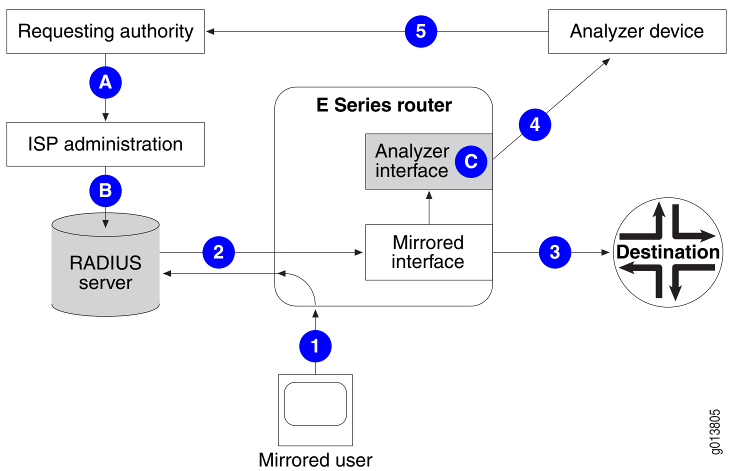
- authorization change
- ip analyzer
- key
- mirror disable

- radius dynamic-request server
- udp-port

RADIUS-Based Mirroring Sequence of Events

Figure 21 on page 266 shows the sequence of events that take place during RADIUS-based mirroring. The tables after the figure describe the events indicated by the numbers and letters in the figure. Table 58 on page 266 describes the configuration process; Table 59 on page 267 describes the flow of traffic during a mirroring operation that is initiated when the user logs in; and Table 60 on page 267 describes the flow of traffic when mirroring a user who is already logged in.

Figure 21: RADIUS-Based Packet Mirroring



To create a RADIUS-based packet-mirroring environment, you must complete the processes listed in Table 58 on page 266.

Table 58: Setting Up the RADIUS-Based Packet-Mirroring Environment

Process	Description
A	The authorized individual requests packet mirroring of the user's traffic and configures the analyzer device to receive mirrored traffic.
B	The ISP administration configures VSAs in the user's RADIUS record.
C	The E Series router administrator configures RADIUS server information and the analyzer interface connection to the analyzer device.

Table 59 on page 267 indicates the sequence of steps for a packet mirroring operation that takes place when a user starts a new session.

Table 59: RADIUS-Based Mirroring During Session Start (User-Initiated)

Step	Description
1	A user logs in to an E Series router, requesting authentication by the RADIUS server. Attributes in the logon request are examined to determine whether any match a configured trigger. The first match starts the packet mirroring session for the user.
2	• The RADIUS server authenticates the user and sends packet mirroring VSAs and any other configured VSAs to the router. • The router creates a secure policy based on the VSAs and starts mirroring the user's traffic.
3	The router sends the user's original traffic to its intended destination.
4	The router sends the mirrored traffic to analyzer device.
5	The analyzer device provides information for the requesting individual.

Table 60 on page 267 indicates the sequence of steps for a packet mirroring operation that is configured for a currently running session.

Table 60: RADIUS-Based Mirroring of Currently Running Session (RADIUS-Initiated)

Step	Description
1	A user logs in to the E Series router; no mirroring action is configured.
2	• Packet mirroring is enabled on the RADIUS server. • Authenticated users are examined to determine whether any match a configured trigger. The first match determines the router to which to send change-of-authorization messages. • The RADIUS server sends change-of-authorization messages containing packet mirroring VSAs to the router. • The router creates a secure policy based on the VSAs and starts mirroring the user's traffic.
3	The router sends the user's original traffic to its intended destination.
4	The router sends mirrored traffic to the analyzer device.
5	The analyzer device provides information for the requesting individual.

Related Documentation

- Configuring RADIUS-Based Packet Mirroring on page 264
- RADIUS-Based Mirroring Overview on page 261

Configuring Router to Start Mirroring When User Logs On

To configure the router to support user-initiated mirroring, which starts when the user logs in:

1. Configure RADIUS server authentication information in the router. See *JunosE Broadband Access Configuration Guide* for information.
2. Configure the analyzer interface to send the mirrored traffic to the analyzer device.

   ```
   host1(config)#interface fastEthernet 4/0
   host1(config-if)#ip analyzer
   ```

   Alternatively, for increased security, create the analyzer interface at one end of an IPSec tunnel to the analyzer device.

   ```
   host1(config)# interface tunnel ipsec:mirror3 transport-virtual-router default
   host1(config-if)#ip analyzer
   host1(config-if)#exit
   host1(config)#ip route 192.168.99.2 255.255.255.255 tunnel ipsec:mirror3
   ```

Related Documentation

- Configuring RADIUS-Based Packet Mirroring on page 264
- Configuring Router to Mirror Users Already Logged In on page 268
- RADIUS-Based Mirroring Sequence of Events on page 266
- interface fastEthernet
- interface tunnel
- ip analyzer

Configuring Router to Mirror Users Already Logged In

When a mirroring operation is initiated for a user who is already logged in (RADIUS-initiated mirroring), the RADIUS server uses change-of-authorization messages and passes the required RADIUS attributes and the identifier of the currently running session to the E Series router. The router uses this information to create the secure policy and attaches it to the interface that is created for the user. The E Series router must be configured to accept change-of-authorization messages from the RADIUS server.

1. Specify the RADIUS dynamic-request server that sends change-of-authorization messages to the router, and enter RADIUS configuration mode.

   ```
   host1(config)#radius dynamic-request server 192.168.11.0
   ```

2. Specify the UDP port used to communicate with the RADIUS server.

   ```
   host1(config-radius)#udp-port 3799
   ```

3. Create the key used to communicate with the RADIUS server.

   ```
   host1(config-radius)#key mysecret
   ```


4. Configure the router to receive change-of-authorization messages from the RADIUS server.

   host1(config-radius)#**authorization change**
   host1(config-radius)#**exit**
   host1(config)#**exit**

5. Verify your RADIUS-initiated mirroring configuration.

   host1#**show radius dynamic-request servers**

```
               RADIUS Request Configuration
               ----------------------------
                                       Change
                Udp                      Of
 IP Address     Port   Disconnect    Authorization   Secret
-------------   ----   ----------    -------------   ------
 10.10.3.4      3799   enabled       enabled         mysecret
```

6. Configure the analyzer interface to send the mirrored traffic to the analyzer device.

   host1(config)#**interface fastEthernet 4/0**
   host1(config-if)#**ip analyzer**

   Alternatively, for increased security, create the analyzer interface at one end of an IPSec tunnel to the analyzer device.

   host1(config)# **interface tunnel ipsec:mirror3 transport-virtual-router default**
   host1(config-if)#**ip analyzer**
   host1(config-if)#**exit**
   host1(config)#**ip route 192.168.99.2 255.255.255.255 tunnel ipsec:mirror3**

Related Documentation

- Configuring RADIUS-Based Packet Mirroring on page 264
- Configuring Router to Start Mirroring When User Logs On on page 268

- interface fastEthernet
- interface tunnel
- ip analyzer
- radius dynamic-request server
- udp-port

CHAPTER 13

Managing Packet Mirroring

Packet mirroring enables you to send a copy of a packet to an external host for analysis. Packet mirroring has many uses, including traffic debugging and troubleshooting user networking problems.

This chapter contains the following topics:

Avoiding Conflicts Between Multiple Packet Mirroring Configurations

The JunosE Software gives you a great deal of flexibility in creating your packet mirroring environment by supporting both the CLI-based and the RADIUS-based configuration methods. However, a conflict might occur when you use both methods. For example, a given subscriber might be targeted by both a CLI-based configuration and a RADIUS-based configuration. The rival configurations might use the same trigger to identify the subscriber, or they might use different triggers.

The configuration method that is applied to the subscriber depends on several variables: the trigger, when the packet mirroring configuration is created, and when the subscriber logs in. The following considerations apply to multiple packet mirroring configurations.

- CLI-based and RADIUS CoA (RADIUS-initiated mirroring) configurations identify targeted subscribers according to the following configured criteria in the order given:
    1. Account session ID
    2. Calling station ID
    3. IP address associated with the virtual router where the subscriber logs in

4. Username associated with the virtual router where the subscriber logs in
5. NAS port ID

- A RADIUS log-in configuration always implicitly uses the Acct-Session-ID to identify the subscriber. This trigger has the highest priority of the five possible identification methods. For this reason, when a subscriber logs in, an existing RADIUS login configuration always takes effect over other packet mirroring configurations.
- A RADIUS CoA configuration affects only subscribers that are currently logged in. It does not create persistent rules. Subscribers that log in after the CoA request goes out are not mirrored by the configuration.

  If a subscriber that is mirrored by a RADIUS CoA configuration subsequently logs out and then logs back in, that subscriber is no longer mirrored by the configuration. However, that subscriber might now be mirrored by an existing RADIUS login or CLI-based configuration.
- A CLI-based configuration creates persistent rules. The configuration affects subscribers that are logged in when the configuration is created, and subscribers that log in thereafter.
- You can create a new configuration or modify an existing configuration to override a configuration that is currently mirroring subscribers. You must use the same subscriber selection criteria that were used by the current configuration. The overriding configuration can be either CLI-based or a RADIUS CoA configuration; it does not have to match the configuration source used by the current configuration.
- When a CLI-based or RADIUS CoA configuration identifies a targeted subscriber group, all members of the group are examined to determine whether any of these members is already mirrored using a different identification method. If that is the case, none of the group members is mirrored by the new configuration.
- Deletion of a CLI rule has no effect on subscribers that are currently being mirrored. They continue to be mirrored as before the deletion. These subscribers are not reevaluated against any remaining identification criteria when a CLI rule is deleted.
- When mirroring is disabled by RADIUS CoA, subscribers that were being mirrored are not evaluated against an existing CLI configuration.

Consider the following scenarios.

Scenario 1: When Configurations Use the Same Identification Criteria

1. Currently logged-in subscribers are not being mirrored. These subscribers include 20 subscribers with the username joe@example.com. Their subscriber access is through virtual router boston1.
2. You create a RADIUS CoA (RADIUS-initiated) configuration that targets subscribers that match joe@example.com logging in through virtual router boston1.
3. Mirroring begins for all 20 of these subscribers.
4. Ten more subscribers with the username joe@example.com log in through VR boston1. None of these new subscribers is mirrored because the RADIUS CoA configuration makes no persistent rules.

5. You create a CLI configuration to mirror subscribers with username joe@example.com logging in through VR boston1.
6. All 30 of these subscribers are now mirrored. The CLI configuration expands the RADIUS CoA configuration because both configurations use the same identification criteria. The original mirrored users continue to be mirrored based on the CoA configuration; the new users are mirrored based on the CLI configuration.
7. You delete the CLI configuration while the subscribers are still logged in and being mirrored. The deletion has no effect on these subscribers; mirroring continues as before the deletion.

Scenario 2: When Configurations Use Different Identification Criteria

1. Currently logged-in subscribers are not being mirrored. These subscribers include 20 subscribers with the username joe@example.com. Their subscriber access is through virtual router boston1.

   The subscribers have been assigned IP addresses 10.1.1.1 through 10.1.1.20.
2. You create a RADIUS CoA (RADIUS-initiated) configuration that targets the subscriber that matches IP address 10.1.1.5 and VR boston1.
3. This subscriber is mirrored.
4. You create a CLI configuration to mirror subscribers with username joe@example.com logging in through VR boston1.
5. No additional subscribers are mirrored because one subscriber that matches that group (username and VR) is already being mirrored by another identification criterion (IP address and VR).

Related Documentation

- Comparing CLI-Based Mirroring and RADIUS-Based Mirroring on page 240
- Using Multiple Triggers for CLI-Based Packet Mirroring on page 277

Understanding the Prepended Header During a Packet Mirroring Session

During a packet mirroring session, the router prepends a special UDP/IP header to each mirrored packet that is sent to the analyzer interface. This prepended header is created by the policy-mirroring action, and is used for demultiplexing at the analyzer to sort through the multiple mirrored streams that arrive from different sources.

All mirrored L2TP session packets are prepended with a UDP/IP header. However, for IP traffic mirroring, the prepend header is optional; the header is added if the mirroring-related VSAs (VSAs 26-59 and 26-61) are both included in the RADIUS message. For CLI-based mirroring, the **analyzer-udp-port** keyword of the **mirror analyzer-ip-address** command creates the same information contained in the two VSAs. If you do not include the VSAs or the **analyzer-udp-port** keyword, an IP mirroring action is indicated, and the prepend header is not used.

NOTE: For IP mirroring, you must include both VSA 26-59 and VSA 26-61, or you must omit both of these VSAs. If you use only one of these VSAs, the configuration fails.

Figure 22 on page 274 shows the structure of the prepended header. The values in parentheses indicate the fixed value for individual fields. For fields that do not have a fixed value listed, the value is dynamically created for each mirrored packet. Table 61 on page 274 lists the fields in the prepended header and indicates the values and field length.

Figure 22: Prepended Header

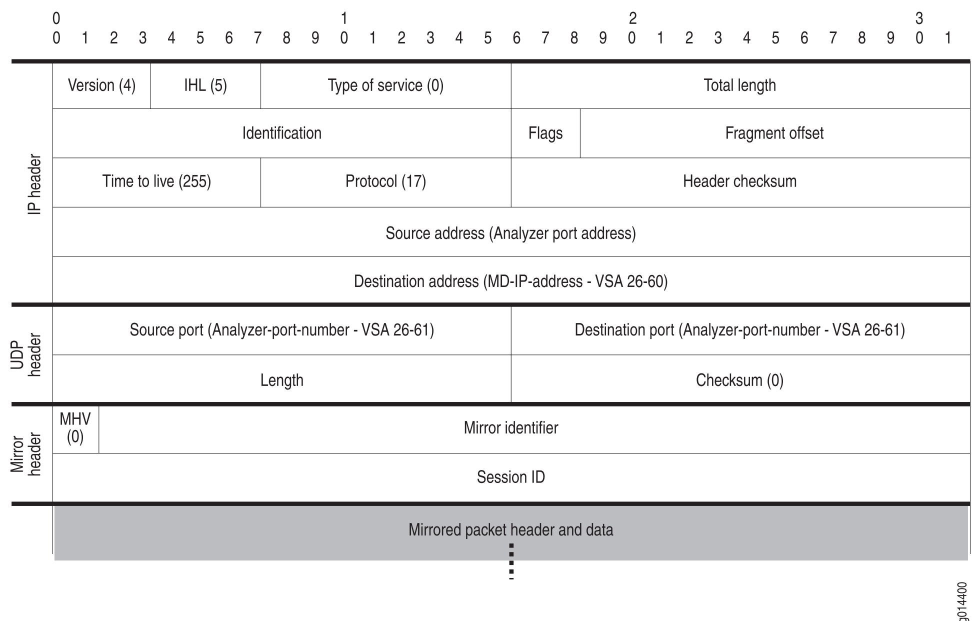


Table 61: Prepended Header Field Descriptions

Field	Value	Length (Bits)
IP Header		
Version	4	4
IHL	5	4
Type of Service	0	8
Total Length	Dynamically computed	16
Identification	Dynamically computed	16
Flags	Dynamically computed	3
Fragment Offset	Dynamically computed	13

Table 61: Prepended Header Field Descriptions *(continued)*

Field	Value	Length (Bits)
Time to Live	255	8
Protocol	17	8
Header Checksum	Dynamically computed	16
Source Address	Analyzer interface IP address	32
Destination Address	VSA 26-60	32
UDP Header		
Source Port	VSA 26-61	16
Destination Port	VSA 26-61	16
Length	Dynamically computed	16
Checksum	0	16
Mirror Header		
MHV (mirror header value)	0	2
Mirror Identifier	See "Format of the Mirror Header Attributes" on page 275 for details	30
Session-ID	See "Format of the Mirror Header Attributes" on page 275 for details	32

Format of the Mirror Header Attributes

The mirror header values are determined by the value that you configure in VSA 26-59. VSA 26-59 is declared as a hexadecimal string that can be either 8 bytes or 4 bytes long. The 8-byte format enables you to further specify the value that is used for the Session-ID field. If you use the 4-byte format, the router automatically determines the Session-ID field. The value in the 2-bit version field specifies the format that is used—0 indicates the 8-byte format, and 1 indicates the 4-byte format.

8-Byte Format

The 8-byte format of VSA 26-59 enables you to manually specify the Session-ID value in addition to the Mirror Identifier value. To use the 8-byte format, you configure the first two most significant bits of the first word of the VSA to a value of 0, which indicates two words in the VSA. The remaining 30 bits of the first word form the Mirror Identifier value, and the second word is the Session-ID field. You cannot change the order of these two words.

For example, a value of 0000030000000090 in VSA 26-59 configures the following fields in the mirror header, as shown in Figure 23 on page 276:

- MHV = 0
- Mirror Identifier = 0x300
- Session-ID = 0x90

Figure 23: 8-Byte Format of VSA 26-59

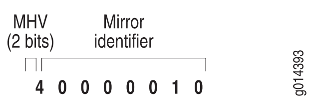
4-Byte Format

To use the 4-byte format of VSA 26-59, you configure the first two most significant bits of the VSA to a value of 1, which indicates a single word in the VSA. The remaining 30 bits of the word form the Mirror Identifier value. The router then creates the Session-ID value based on the least significant 32 bits of the Acct-Session-ID (RADIUS attribute 44).

For example, a value of 40000010 for VSA 26-59 configures the following fields in the mirror header, as shown in Figure 24 on page 276:

- MHV = 1
- Mirror Identifier = 0x10

Figure 24: 4-Byte Format of VSA 26-59



Related Documentation

- Configuring RADIUS-Based Packet Mirroring on page 264
- Packet Mirroring Overview on page 239
- RADIUS-Based Mirroring Sequence of Events on page 266

Resolving and Tracking the Analyzer Device's Address

During the packet mirroring configuration process, you specify the IP address of the analyzer device to which the mirrored traffic is sent. For CLI-based packet mirroring, you use the **mirror analyzer-ip-address** command to specify the IP address. For RADIUS-based packet mirroring, the RADIUS attribute Med-IP-Address [26-60] is the address of the analyzer device.

After configuration is complete, the router performs a route lookup to resolve the analyzer device's address and to ensure that traffic can be forwarded to the analyzer device for analysis. However, the analyzer device is considered unreachable if the router's analyzer interface is not in analyzer mode, is not yet created, or if the routes to the analyzer device are absent

If the analyzer device is unreachable, then the mirror action in the secure policy is disabled, and no packets are mirrored. The **show secure policy-list** command output indicates that the mirror action is disabled and the analyzer device is unreachable.

The router tracks the analyzer device's IP address for any route changes within the router. This tracking ability provides a degree of failure recovery by enabling you to configure multiple analyzer interfaces to serve as redundant ports to reach the analyzer device.

Related Documentation

- Configuring the Analyzer Device on page 254
- mirror analyzer-ip-address
- show secure policy-list

Using Multiple Triggers for CLI-Based Packet Mirroring

When you configure CLI-based packet mirroring, you can create multiple mirroring rules for a particular subscriber. For example. you might create two rules; one rule that uses IP address as the trigger that identifies the user and a second rule with the subscriber's username as the trigger. You can also configure RADIUS-based mirroring to use multiple methods to identify subscribers

To avoid conflicts between multiple mirroring rules, both CLI-based and RADIUS-based mirroring operations assign a precedence to the subscriber identification triggers. Subscriber information is examined for configured triggers according to the order of precedence.

The following list indicates the order of precedence for the subscriber identification triggers; Acct-Session-Id has the highest precedence. The keywords for the **mirror** and **mirror disable** command are listed below with their associated RADIUS attributes.

1. **acct-session-id**—Acct-Session-Id, RADIUS attribute [44]
2. **calling-station-id**—Calling-Station-Id, RADIUS attribute [31]
3. **ip-address**—Framed-IP-Address, RADIUS attribute [8]; associated with the virtual router where the subscriber logs in, RADIUS VSA [26-1]
4. **username**—User-Name, RADIUS attribute [1]; associated with the virtual router where the subscriber logs in, RADIUS VSA [26-1]
5. **nas-port-id**—NAS-Port-Id, RADIUS attribute [87]
6. **dhcp-option-82**—DHCP-Option-82, RADIUS attribute [26–159], Vendor ID 4874
7. **agent-circuit-id**—Agent-Circuit-ID, RADIUS attribute [26–1], Vendor ID 3561
8. **agent-remote-id**—Agent-Remote-ID, RADIUS attribute [26–2], Vendor ID 3561

For example, suppose you create the following three rules to trigger a packet mirroring session.


```
host1(config)#mirror ip-address 192.168.105.25 ip secure-policy-list securePolicyIp4
host1(config)#mirror username jwbooth@isptheatre.com ip secure-policy-list
  securePolicyIp15
host1(config)#mirror acct-session-id atm 2/1.2:0.42:0001048579 ip secure-policy-list
  securePolicyIp10
```

Regardless of the order in which you configure the rules, the subscriber information is first examined to determine whether the Acct-Session-Id matches the rule. If it does, no further examination takes place and the subscriber's traffic is mirrored,

If the Acct-Session-Id does not match, then the subscriber information is next examined to determine whether the Calling-Station-Id matches the rule. This process continues for all configured rules.

If none of the trigger rules are matched, then that subscriber's traffic is not mirrored.

If the packet mirroring request is a RADIUS-initiated session (a RADIUS-based packet mirroring session for a subscriber who is already logged in), the router verifies the validity of all of the mirroring rules related to the particular subscriber. If any of the rules fail (for example, the identification fields do not match), the packet mirroring request is denied.

The calling-station-id trigger is externally visible only for tunneled users (if there are no RADIUS overrides). If a case-sensitive user name does not match a subscriber's name or if the dynamic IP interface UID does not exist, the subscriber is disregarded.

Related Documentation

- Avoiding Conflicts Between Multiple Packet Mirroring Configurations on page 271
- Configuring Triggers for CLI-Based Mirroring on page 253

Optimizing Packet Mirroring Performance

Packet mirroring operations require some system resources. As a general rule, to avoid performance degradation, limit the amount of mirrored traffic to a maximum of 5 percent of the E Series router's total traffic.

For many packet mirroring environments, using the 5-percent guideline is sufficient. However, if you want to more closely manage packet mirroring's use of your router's resources, this section provides guidelines and equations to help you determine your packet mirroring requirements.

The guidelines for packet mirroring requirements use the following assumptions for a specific line module:

- A = Total input traffic at the line module
- B = Total output traffic at the line module
- X = Amount of traffic mirrored at input in the line module
- Y = Amount of traffic mirrored at output in the line module

Determine Traffic Loads

Using the previous assumptions, you can determine traffic loads for a given line module:

- A = Load at ingress side of the line module
- (B + X) = Load at egress side of the line module
- (A + 2X + Y) = Load at ingress to fabric from the line module

Establish Resource Guidelines

Next, using the traffic loads that you determined for the line module, you can establish guidelines for the amount of packet mirroring traffic for your router.

If you exceed these guidelines, regular (non-packet mirroring) packets from all subscribers, including nonmirrored subscribers, will be dropped. If the fabric bandwidth is not exceeded, then the performance penalties are contained within the slot where the packet mirroring activity occurs. However, if the fabric bandwidth is exceeded, traffic from other line modules might also be dropped.

- (A + 2X + Y) must be less than the maximum fabric bandwidth supported from this line module.
- (2X + Y) must be less than 100Mbps (the enforced queue limit).

  The 100 Mbps limit does not apply to the following line modules:

  - GE-2 line module (Juniper Networks ERX310 and ERX1440 Broadband Services Routers)
  - GE-HDE line module (ERX310 and ERX1440 router)
  - OC48 Frame APS I/O module (ERX1440 router only)
  - ES2 4G LM (E120 router and E320 Broadband Services Routers)
  - ES2 10G LM (E120/E320)
- (B + X) must be less than the maximum supported egress bandwidth.
- The number of mirrored interfaces per line module must be less than 1023 (the configuration enforced for secure policy attachments).
- The number of interfaces mirrored per chassis must be less than 2400 (the configuration enforced for secure policy attachments).

NOTE: **Packet mirroring can also affect the forwarding controller's packet handling performance.**

Related Documentation

- Packet Mirroring Overview on page 239

Logging Packet Mirroring Information

The JunosE Software's packet mirroring feature provides two secure methods of capturing and displaying packet mirroring-related information. Both methods ensure security by requiring the **mirror-enable** command to be enabled.

- Secure logging—Captures packet mirroring information to a local secure log on the router.
- SNMP secure packet mirroring traps—Captures and reports packet mirroring information to an external device; you can then use the privileged **show mirror trap** and **show snmp traps** CLI commands to view secure trap configuration information.

  SNMP agent also implements a secure audit logging facility for the debugging of packet mirroring traps and packet Mirror-MIB accesses. When secure audit logging is enabled, SNMP agent logs reported mirror traps and packet Mirror-MIB get/set operations to local volatile memory on the router.

By default, the JunosE Software captures packet mirroring-related activity to a secure local mirror log. No action is required on your part to enable or disable the logging process; however, only authorized users can access the secure log.

The secure logging feature includes the **clear mirror log** and **show mirror log** commands. The **mirror-enable** command must be enabled to make the commands visible in the CLI.

Related Documentation

- Packet Mirroring Overview on page 239
- clear mirror log
- show mirror log

Using SNMP Secure Packet Mirroring Traps

SNMP secure packet mirroring traps enable you to capture and report packet mirroring information to an external device; you can then view the secure information on the remote device. The secure packet mirroring traps feature is an extension of the router's standard SNMP implementation, and is only available to SNMPv3 users who are authorized to use packet mirroring.

You can also log mirror traps to local volatile memory for debugging purposes by enabling the SNMP secure log feature. See "Capturing SNMP Secure Audit Logs" on page 285 for details of secure audit logging. Normal console and syslog audit logs for packet mirroring traps and packet Mirror-MIB accesses are suppressed due to security concerns.

NOTE: **The contents of secure logs are not preserved across a reboot.**

The **mirror-enable** command must be enabled to make packet mirroring-related commands, command options, and **show** command output visible.

NOTE: You must use the CLI to configure the secure packet mirroring trap category to allow transmission of secure packet mirroring traps through the router—you cannot use SNMP to configure the secure packet mirroring trap category. However, after you have configured the secure packet mirroring trap category using the CLI, you can then use SNMP (juniPacketMirrorMIB.mi2) to enable and disable secure packet mirroring traps.

Table 62 on page 281 indicates the events that trigger secure packet-mirroring traps and lists the information sent in the trap for each event.

Table 62: Packet-Mirroring SNMP Traps

Trap Information Sent	Event That Triggers the Trap			
	A secure policy failed during CoA-based or RADIUS-initiated packet mirroring	A secure policy failed during CLI trigger or CLI-based packet mirroring	An interface with secure policies attached is deleted	An analyzer is unreachable
Analyzer address	–	–	–	✓
Application name	✓	✓	–	–
Configuration source	✓	✓	✓	–
Date and time of event	–	✓	✓	✓
Error cause	✓	✓	–	–
Error string	✓	✓	–	–
Mirror ID	✓	–	✓	–
Mirroring direction	–	–	✓	–
Secure policy name	–	✓	✓	–
Secure policy UID	–	✓	✓	–
Session ID	✓	–	✓	–
Trigger event	✓	✓	✓	–
Trigger type	✓	✓	✓	–
Username	✓	–	–	–

Table 62: Packet-Mirroring SNMP Traps *(continued)*

Trap Information Sent	Event That Triggers the Trap			
	A secure policy failed during CoA-based or RADIUS-initiated packet mirroring	A secure policy failed during CLI trigger or CLI-based packet mirroring	An interface with secure policies attached is deleted	An analyzer is unreachable
Virtual router (0 for L2TP)	✓	✓	✓	✓

Additional Packet-Mirroring Traps for CALEA Compliance

You can use the packet-mirroring traps shown in Table 63 on page 282 to help support compliance with the Communications Assistance for Law Enforcement Act (CALEA), which defines electronic surveillance guidelines for telecommunications companies. For example, a third-party vendor of mediation devices might receive packet mirroring traps from the router and convert the traps to messages that comply with CALEA, such as Lawfully Authorized Electronic Surveillance (LAES) for IP Network Access, American Nation Standard For Telecommunications messages. Individual traps might map to multiple LAES messages to provide additional compliance-related information.

Table 63: Packet-Mirroring Traps for CALEA Compliance

Trap	Description
juniPacketMirrorSessionStart	A grant has been issued to a mirrored subscriber.
juniPacketMirrorSessionEnd	A mirrored session has been terminated; includes the termination reason.
juniPacketMirrorInterfaceSessionActivated	A secure policy has been attached to an existing interface or to an existing session.
juniPacketMirrorInterfaceSessionDeactivated	A secure policy has been detached from an interface, not including interface or session termination.
juniPacketMirrorSessionReject	A deny has been issued because the potential mirrored user was not allowed on the network for some reason. However, the user would have been mirrored if access to the network had been allowed.
juniPacketMirrorSessionFailed	The user session was terminated before the secure policy was attached. For example, no resources were available to create the interface. The termination reason is included.

Packet Mirroring Trap Severity Levels

Table 64 on page 283 lists the default severity levels for packet mirroring traps. See the *JunosE System Basics Configuration Guide* for descriptions of the severity levels.

Table 64: Packet Mirroring Trap Severity Levels

Trap	Default Severity Level
juniPacketMirrorAnalyzerUnreachable	Warning
juniPacketMirrorCliTriggerBasedMirroringFailure	Error
juniPacketMirrorInterfaceDeleted	Notice
juniPacketMirrorInterfaceSessionActivated	Info
juniPacketMirrorInterfaceSessionDeactivated	Info
juniPacketMirrorRadiusBasedMirroringFailure	Error
juniPacketMirrorSessionEnd	Info
juniPacketMirrorSessionFailed	Info
juniPacketMirrorSessionStart	Info
juniPacketMirrorSessionReject	Info

See Configuring SNMP in *JunosE System Basics Configuration Guide* for information about JunosE Software SNMP support.

Related Documentation

- Configuring SNMP Secure Packet Mirroring Traps on page 284
- Monitoring SNMP Secure Packet Mirroring Traps on page 299
- mirror trap-enable
- snmp-server clear secure-log
- snmp-server enable traps
- snmp-server host
- snmp-server secure-log
- show mirror trap
- show snmp secure-log

Configuring SNMP Secure Packet Mirroring Traps

To configure SNMP secure traps support, perform the following tasks on your E Series router:

1. Enable packet mirroring support.
2. Configure the packet mirroring application to generate traps.
3. (Optional) Verify the packet mirroring trap configuration.
4. (Optional) Configure the SNMP server to support secure logs.
5. Configure the SNMP server to generate packet mirroring traps.
6. Configure the SNMPv3 user for whom packet mirroring traps are generated.
7. Configure the SNMP server to report packet mirroring traps to a remote host.
8. (Optional) Verify the SNMP server packet mirroring configuration.

The following example illustrates the procedure to configure SNMP secure packet mirroring traps support:

```
host1#mirror-enable
host1#configure terminal
host1(config)#mirror trap-enable
host1(config)#show mirror trap
Traps are enabled
host1(config)#snmp-server secure-log
host1(config)#snmp-server user fredMirrorUser group mirror authentication md5
  fred-md5password privacy des fred-despassword
host1(config)#snmp-server enable traps packetMirror trapFilters notice
host1(config)#snmp-server host 192.168.57.103 version 3 fredMirrorUser  cliSecurityAlert
  packetMirror trapFilters notice
host1(config)#show snmp trap
```

```
Enabled Categories: CliSecurity, PacketMirror, Sonet
SNMP authentication failure trap is disabled
Trap Source: FastEthernet 6/0, Trap Source Address:192.168.120.78
Trap Proxy: enabled
Global Trap Severity Level: 6 - informational
Address          Security String                     Ver  Port   Trap Categories
---------------  --------------------------------    ---  -----  ----------------
192.168.1.1      host1                               v1   162    Cli
192.168.57.103   fredMirrorUser                      v3   162    CliPacketMirror
192.168.57.162   host2                               v3   162    Sonet
Address         TrapSeverityFilter  Ping    Maximum    Queue     Queue Full
                                   TimeOut QueueSize DrainRate discrd methd
--------------- ------------------ ------- --------- --------- -------------
192.168.1.1     5 - notice         1       32        0         dropLastIn
192.168.57.103  5 - notice         1       32        0         dropLastIn
192.168.57.162  2 - critical       1       32        0         dropLastIn
```

See Configuring SNMP in *JunosE System Basics Configuration Guide* for information about JunosE Software SNMP support.

Related Documentation

- Using SNMP Secure Packet Mirroring Traps on page 280
- Monitoring SNMP Secure Packet Mirroring Traps on page 299
- mirror trap-enable
- snmp-server clear secure-log
- snmp-server enable traps
- snmp-server host
- snmp-server secure-log
- show mirror trap
- show snmp secure-log

Capturing SNMP Secure Audit Logs

SNMP secure audit logging enables administrators to collect the SNMP audit logs for mirror traps and Mirror-MIB get/set operations with the protection of the mirror enabling feature. Secure audit logging facilitates the debugging of issues related to SNMP packet mirror traps.

All normal SNMP console and syslog audit logs (including snmpTrap, snmpPduAudit, and snmpSetPduAudit) for secure traps and Mirror-MIB are suppressed due to security concerns. When you have issued the **mirror enable** command, you can issue the **snmp secure-log** command to capture secure audit logs. Configuration, storage, and display of the SNMP secure logging is on global basis rather than a per-VR basis.

The SNMP agent captures and stores the audit logs for secure traps. The SNMP agent also captures PDU audit logs for Mirror-MIB operations. Configure the snmpTrap, snmpPduAudit, and snmpSetPduAudit logs at the proper severity level to capture the secure audit logs.

You can use the **show snmp secure-log** command to display the captured secure logs. Secure logs are stored in a string format similar to SNMP trap audit logs. You can use the **snmp-server clear secure-log** command to reset the secure logs.

The secure log data is not persistent. Secure audit logs are not available after a warm or cold restart of the SNMP agent, because the SNMP agent does not store the secure logs in NVS. The SNMP agent can store a maximum of 100 secure logs before overwriting the logs.

The secure log configuration is persistent. The configuration is available after a warm restart operation because it is stored in the nonvolatile memory. Based on the configuration, data is logged for the packet mirrors that are automatically applied during subscriber login for the newly attached secure policy after the restart operation.

To enhance security, you can configure and display the secure audit logs only through the CLI. You cannot use SNMP to configure and display the logs. Secure trap logs are not populated in the notification logs MIB. From the perspective of the notification log MIB, secure traps do not exist.

Related Documentation

- Monitoring SNMP Secure Audit Logs on page 301
- snmp-server clear secure-log
- snmp-server secure-log
- show snmp secure-log
- show snmp trap

CHAPTER 14

Monitoring Packet Mirroring

Packet mirroring enables you to send a copy of a packet to an external host for analysis. Packet mirroring has many uses, including traffic debugging and troubleshooting user networking problems.

This chapter contains the following topics:

Monitoring Packet Mirroring Overview

This topic describes the commands you can use to view your CLI-based and RADIUS-based packet mirroring environments.

Use the **baseline radius dynamic-request** command in RADIUS-based packet mirroring to set a statistics baseline for packet mirroring–related RADIUS statistics. The E Series router implements the baseline by reading and storing the statistics at the time the baseline is set and then subtracting this baseline when you retrieve baseline-relative statistics. Use the **delta** keyword with the **show radius statistics** command to show baselined statistics.

Related Documentation

- Packet Mirroring Overview on page 239
- baseline radius dynamic-request

- clear mirror log

Monitoring CLI-Based Packet Mirroring

Purpose Display brief or default (normal) information about your CLI-based packet mirroring environment, including interface analyzer information. To display secure packet mirroring information you must enable the **mirror-enable** command before using this command. This command displays a maximum of two secure policy attachments and statistics, if configured.

Action To display the default (normal) format for a specific interface, which is used as the default analyzer interface:

```
host1#show ip interface atm 5/0.1
ATM5/0.1 line protocol Atm1483 is up, ip is analyzer (default)
  Network Protocols: IP
  Internet address is 10.10.3.4/255.255.255.0
  Broadcast address is 255.255.255.255
  Operational MTU = 0  Administrative MTU = 0
  Operational speed = 100000000  Administrative speed = 0
  Discontinuity Time = 0
  Router advertisement = disabled
  Proxy Arp = disabled
  Administrative debounce-time = disabled
  Operational debounce-time    = disabled
  Access routing = disabled
  Multipath mode = hashed

  In Received Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Packets 0, Bytes 0
  In Policed Packets 0, Bytes 0
  In Error Packets 0
  In Invalid Source Address Packets 0
  In Discarded Packets 0
  Out Forwarded Packets 0, Bytes 0
    Unicast Packets 0, Bytes 0
    Multicast Routed Packets 0, Bytes 0
  Out Scheduler Dropped Packets 0, Bytes 0
  Out Policed Packets 0, Bytes 0
  Out Discarded Packets 0
```

To display the format for a specific interface, showing secure policy attachments:

```
host1#show ip interface atm 4/1.1
ATM5/0.1 line protocol Atm1483 is up
  Network Protocols: IP
  Internet address is 10.10.7.14/255.255.255.0
  Broadcast address is 255.255.255.255
  Operational MTU = 0  Administrative MTU = 0
  Operational speed = 100000000  Administrative speed = 0
  Discontinuity Time = 0
  Router advertisement = disabled
  Proxy Arp = disabled
  Administrative debounce-time = disabled
  Operational debounce-time    = disabled
  Access routing = disabled
  Multipath mode = hashed
```



```
In Received Packets 0, Bytes 0
  Unicast Packets 0, Bytes 0
  Multicast Packets 0, Bytes 0
In Policed Packets 0, Bytes 0
In Error Packets 0
In Invalid Source Address Packets 0
In Discarded Packets 0
Out Forwarded Packets 0, Bytes 0
  Unicast Packets 0, Bytes 0
  Multicast Routed Packets 0, Bytes 0
Out Scheduler Dropped Packets 0, Bytes 0
Out Policed Packets 0, Bytes 0
Out Discarded Packets 0

IP policy secure-input ipSecureIn
  classifier-group secClassA entry 1
    0 packets, 0 bytes
    mirror analyzer-ip-address 10.10.3.14, analyzer-virtual-router default
  classifier-group secClassB entry 2
    0 packets, 0 bytes
    mirror analyzer-ip-address 10.10.3.14, analyzer-virtual-router vr200
IP policy secure-output ipSecureOut
  classifier-group secClassC entry 1
    0 packets, 0 bytes
    mirror analyzer-ip-address 10.10.7.104, analyzer-virtual-router vr300
```

Meaning Table 65 on page 289 lists the secure packet mirroring-related fields.

Table 65: show ip interface Output Fields

Field Name	Field Description
IP Policy	Type (secure-input, secure-output) and name of the secure policy
classifier-group	Name of a CLACL attached to the interface and number of entry
packets	Number of packets classified by the CLACL
bytes	Number of bytes classified by the CLACL
mirror analyzer-ip-address	IP address of analyzer device
analyzer-virtual-router	Name of analyzer interface virtual router

Related Documentation

- Configuring CLI-Based Packet Mirroring on page 251
- show ip interface

Monitoring the Packet Mirroring Configuration of IP Interfaces

Purpose Display CLI-based packet mirroring configuration information for a specific interface or for all interfaces on which mirroring is enabled.

NOTE: This command is deprecated and might be removed completely in a future release. The function provided by this command has been replaced by the **show secure policy-list** command.

Action To display information about a specific interface or for all interfaces:

host1#**show ip mirror interface atm 5/0.1**

```
    Interface          Analyzer Port      Analyzer next-hop
------------------- ------------------- ---------------------
ATM5/0.1             FastEthernet3/0       192.168.1.1
```

Meaning Table 66 on page 290 lists the **show ip mirror interface** command output fields.

Table 66: show ip mirror interface Output Fields

Field Name	Field Description
Interface	Interface being mirrored
Analyzer Port	Interface to which the mirrored traffic is sent, and that then sends the traffic to the analyzer device
Analyzer next-hop	IP address of the next hop to the analyzer device; displayed when the analyzer interface is a shared medium

Related Documentation

- Configuring CLI-Based Packet Mirroring on page 251
- show ip mirror interface

Monitoring Failure Messages for Secure Policies

Purpose Display failure messages and information for secure policies. This command and the output are visible only to authorized users—the **mirror-enable** command must be enabled before using this command. All normal E Series system log messages are suppressed for packet mirroring-related policy operations.

Action To display information for secure policies:

host1#**show mirror log**

```
 Time              Mirror-ID   Session-ID   User        Error Status
-----------      ------------ ----------- ------------ --------------
TUE SEP 15         8976        1923     123@abc.com  no secure policies available

2009 18:35:43 UTC
```

Meaning Table 67 on page 291 lists the **show mirror log** command output fields.

Table 67: show mirror log Output Fields

Field Name	Field Description
Time	Day, date, and time of failure
Mirror-ID	Unique identifier of the mirrored session
Session-ID	Unique identifier of the user session
User	User login name
Error Status	Description of error condition

Related Documentation

- Configuring CLI-Based Packet Mirroring on page 251
- show mirror log

Monitoring Packet Mirroring Triggers

Purpose Display CLI-based packet mirroring information about all packet mirroring triggers (active and inactive) that are configured on the router. This command and the output are visible only to authorized users—the **mirror-enable** command must be enabled before using this command.

Action To display information about all packet mirroring triggers:

```
host1# show mirror rules
 Mirror Trigger Rules : 11 Total
Subscriber ID                                       ID             Secure       Secure      Sessions
                                                   Method         Policy Type Policy List Mirrored

------------------                              ------------------  ------------ ----------  --------
default:1.2.3.4                                     IP address      IP          spl_88000001  0
52:11:02:0F:12:4F:87:3A:72:65:6D:6F:74:65:00:69:64 dhcp-option-82   IP          spl_88000002  1
01:0D:61:74:6D:20:34:2F:32:3A:30:2E:31:30:35       agent-circuit-id IPV6        spl_88000003  1
02:0F:67:68:69:31:40:64:6F:6D:61:69:6E:2E:63:6F:6D agent-remote-id  IPV6        spl_88000004  1
52:10:01:0E:12:4F:87:3A:61:67:65:6E:74:00:69:64    dhcp-option-82   IP        op82hex_policy  1
01:0D:61:74:6D:20:34:2F:32:3A:30:2E:31:30:31       agent-circuit-id IPV6      cidhex_policy   1
02:0E:61:62:63:40:64:6F:6D:61:69:6E:2E:63:6F:6D    agent-remote-id  IPV6      ridhex_policy   1
01:0E:63:69:64:40:64:6F:6D:61:69:6E:2E:63:6F:6D    agent-remote-id  L2TP      l2tphex_pol     1
atm 4/1:0.101.abc@domain.com                        dhcp-option-82   IP        op82string_plcy 0
atm 4/2:0.101                                       agent-circuit-id IP        cidstring_plcy  0
user@juniper.com                                    agent-remote-id  IP        ridstring_plcy  0
```

Meaning Table 68 on page 291 lists **show mirror rules** command output fields.

Table 68: show mirror rules Output Fields

Field Name	Field Description
Subscriber ID	Identification of the subscriber

Table 68: show mirror rules Output Fields *(continued)*

Field Name	Field Description
ID Method	Method used to identify the subscriber
Secure Policy Type	Type of secure policy; IP, IPv6, or L2TP
Secure Policy List	Name of secure policy list used for packet mirroring
Sessions Mirrored	Number of sessions currently being mirrored

Related Documentation

- Configuring Triggers for CLI-Based Mirroring on page 253
- Using Multiple Triggers for CLI-Based Packet Mirroring on page 277
- show mirror rules

Monitoring Packet Mirroring Subscriber Information

Purpose Display CLI-based packet mirroring information about the subscribers for whom packet mirroring is currently active. This command and the output are visible only to authorized users—the **mirror-enable** command must be enabled before using this command.

Action To display information about subscribers for whom packet mirroring is active:

```
host1# show mirror subscribers
                               Mirror Subscribers:  7 Total

Subscriber ID                                      ID            Secure    Secure        Mirrored
                                                 Method        PolicyType Policy Name    Sessions
------------------                               --------------- --------- -------------- --------
52:10:01:0E:12:4F:87:3A:61:67:65:6E:74:00:69:64    dhcp-option-82     IP     op82hex_pol     1
01:0D:61:74:6D:20:34:2F:32:3A:30:2E:31:30:31       agent-circuit-id IPV6     cidhex_poli     1
02:0E:61:62:63:40:64:6F:6D:61:69:6E:2E:63:6F:6D    agent-remote-id  IPV6     ridhex_poli     1
52:11:02:0F:12:4F:87:3A:72:65:6D:6F:74:65:00:69:64 dhcp-option-82   IP       spl_88000002    1
01:0D:61:74:6D:20:34:2F:32:3A:30:2E:31:30:35       agent-circuit-id IPV6     spl_88000003    1
02:0E:61:62:63:40:64:6F:6D:61:69:6E:2E:63:6F:6E    agent-remote-id  IPV6     spl_88000004    1
01:0E:63:69:64:40:64:6F:6D:61:69:6E:2E:63:6F:6D    agent-remote-id  L2TP     l2tphex_pol     1
```

Meaning Table 69 on page 292 lists **show mirror subscribers** command output fields.

Table 69: show mirror subscribers Output Fields

Field Name	Field Description
Subscriber ID	Subscriber being mirrored
Subscriber ID Method	Method used to identify the subscriber
Secure Policy Type	Type of secure policy; IP, IPv6, or L2TP

Table 69: show mirror subscribers Output Fields *(continued)*

Field Name	Field Description
Secure Policy List	Name of secure policy list used for packet mirroring
Sessions Mirrored	Number of sessions being mirrored

Related Documentation

- Configuring CLI-Based Packet Mirroring on page 251
- Example: Configuring CLI-Based User-Specific Packet Mirroring on page 257
- show mirror subscribers

Monitoring RADIUS Dynamic-Request Server Information

Purpose Display RADIUS dynamic-request server configuration information and statistics.

Action To display RADIUS dynamic-request server configuration information:

host1#**show radius dynamic-request servers**

```
               RADIUS Request Configuration
                       ----------------------------
                                       Change
               Udp                      Of
 IP Address    Port   Disconnect   Authorization   Secret
-------------  ----   ----------   -------------   ------
192.168.2.3    1700   disabled     disabled        <NULL>
10.10.120.104  1700   disabled     disabled        mysecret
```

host1#**show radius dynamic-request statistics**

```
RADIUS Request Statistics
       -------------------------
          Statistic              10.10.3.4
-----------------------------   -------
UDP Port                         1700
Disconnect Requests              0
Disconnect Accepts               0
Disconnect Rejects               0
Disconnect No Session ID         0
Disconnect Bad Authenticators    0
Disconnect Packets Dropped       0
CoA Requests                     0
CoA Accepts                      0
CoA Rejects                      0
CoA No Session ID                0
CoA Bad Authenticators           0
CoA Packets Dropped              0
No Secret                        0
Unknown Request                  0

Invalid Addresses Received  :0
```

Meaning Table 70 on page 294 lists **show radius dynamic-request statistics** command output fields.

Table 70: show radius dynamic-request statistics Output Fields

Field Name	Field Description
IP Address	IP address of the RADIUS server
Udp Port	Port on which the router listens for RADIUS server
Disconnect	Status of RADIUS-initiated disconnect feature, enabled or disabled
Change of Authorization	Status of change of authorization feature, enabled or disabled
Secret	Secret (key) used to connect to RADIUS server
Disconnect or CoA Requests	Number of RADIUS-initiated disconnect or CoA requests received
Disconnect or CoA Accepts	Number of RADIUS-initiated disconnect or CoA requests accepted
Disconnect or CoA Rejects	Number of RADIUS-initiated disconnect or CoA requests rejected
Disconnect or CoA No Session ID	Number of RADIUS-initiated disconnect or CoA messages rejected because the request did not include a session ID attribute
Disconnect or CoA Bad Authenticators	Number of RADIUS-initiated disconnect or CoA messages rejected because the calculated authenticator in the authenticator field of the request did not match
Disconnect or CoA Packets Dropped	Number of RADIUS-initiated disconnect or CoA packets dropped because of queue overflow
No Secret	Number of messages rejected because a secret was not present in the authenticator field
Unknown Request	Number of packets received with an invalid RADIUS code for RADIUS disconnect or change of authorization
Invalid Addresses Received	Number of invalid addresses received

Related Documentation

- Configuring RADIUS-Based Packet Mirroring on page 264
- show radius servers
- show radius statistics

Monitoring Secure CLACL Configurations

Purpose Display information about only secure CLACL configurations. This command and the output are visible only to authorized users—the **mirror-enable** command must be enabled before using this command. Use the **brief** or **detail** keywords with the **show secure classifier-list** command to display different levels of information.

Action To display a list of secure CLACLs

host1#**show secure classifier-list**

```
                         Classifier Control List Table
                         ---------- ------- ---- -----
Secure IP secClassA.1 ip any any
Secure IP secClassB.1 ip any not 10.10.10.1 255.255.255.255
Secure IP secClass25.1 user-packet-class 8 source-route-class 100 ip
192.168.44.103 255.255.255.255 any
```

Displays details of each secure CLACL

host1#**show secure classifier-list secClass25 detailed**

```
                         Classifier Control List Table
                         ---------- ------- ---- -----
Secure IP Classifier Control List secClass25
   Reference count:      0
   Entry count:          1

   Classifier-List secClass25 Entry 1
      User Packet Class:         8
      Source Route Class:        100
      Protocol:                  ip
      Not Protocol:              false
      Source IP Address:         192.168.44.103
      Source IP WildcardMask:    255.255.255.255
      Not Source Ip Address:     false
      Destination IP Address:    0.0.0.0
      Destination IP WildcardMask:255.255.255.255
      Not Destination Ip Address: false
```

Meaning Table 71 on page 295 lists **show secure classifier-list** command output fields.

Table 71: show secure classifier-list Output Fields

Field Name	Field Description
Reference count	Number of times the CLACL is referenced by policies
Entry count	Number of entries in the classifier list
Classifier-List	Name of the classifier list
Entry	Entry number of the classifier list rule
Color	Packet color to match: green, yellow, or red
Protocol	Protocol type

Table 71: show secure classifier-list Output Fields *(continued)*

Field Name	Field Description
Not Protocol	If true, matches any protocol except the preceding protocol; if false, matches the preceding protocol
Source IP Address	Address of the network or host from which the packet is sent
Source IP WildcardMask	Mask that indicates addresses to be matched when specific bits are set
Not Source Ip Address	If true, matches any source IP address and mask except the preceding source IP address and mask; if false, matches the preceding source IP address and mask
Destination IP Address	Number of the network or host from which the packet is sent
Destination IP WildcardMask	Mask that indicates addresses to be matched when specific bits are set
Not Destination Ip Address	If true, matches any destination IP address and mask except the preceding destination IP address and mask; if false, matches the preceding destination IP address and mask
Traffic Class	Name of the traffic class to match
User Packet Class	User packet value to match
DS Field	DS field value to match
TOS Byte	ToS value to match
Precedence	Precedence value to match
User Priority bits	User priority bits value to match
Traffic Class Field	Traffic class field value to match
EXP Bits	MPLS EXP bit value to match
EXP Mask	Mask applied to EXP bits before matching
DE Bit	Frame Relay DE bit value to match5.2.0b1 ID-1381
Destination Route Class	Route class used to classify packets based on the packet's destination address
Source Route Class	Route class used to classify packets based on the packet's source address

Table 71: show secure classifier-list Output Fields *(continued)*

Field Name	Field Description
Local	If true, matches packets destined to a local interface; if false, matches packets that are traversing the router

Related Documentation

- Configuring CLI-Based Packet Mirroring on page 251
- show secure classifier-list

Monitoring Secure Policy Lists

Purpose Display information about only secure policy lists. This command and the output are visible only to authorized users—the **mirror-enable** command must be enabled before using this command. Use the **name** keyword to display information for a specific secure policy list.

Action To display information about secure policy lists:

```
host1#show secure policy-list
                                  Policy Table
                                  ------ -----
Secure IP Policy secureIpPolicy
   Administrative state: enable
   Reference count:      2
   Classifier control list: secClassA
      mirror analyzer-ip-address 192.168.1.1 analyzer-virtual-router default
analyzer-udp-port 3000 mirror-id 6789 session-id 6543

   Referenced by interface(s):
      ATM5/0.1  secure-input policy, statistics disabled, virtual-router default

      ATM5/0.1  secure-output policy, statistics disabled, virtual-router default


Secure IPv6 Policy secure-ipv6-pol3
   Administrative state: enable
   Reference count:      2
   Classifier control list: *
      Mirror analyzer-ip-address 190.168.1.1 analyzer-virtual-router default
analyzer-udp-port 3000 mirror-id 6789 session-id 6543

   Referenced by interface(s):
      GigabitEthernet1/0/2.1.2  secure-input policy, statistics disabled,
virtual-router default
      GigabitEthernet1/0/2.1.2  secure-output policy, statistics disabled,
virtual-router default

   Referenced by merged policies:
      None

L2TP Secure Policy secureL2tpPolicy
   Administrative state: enable
   Reference count:      2
   Classifier control list: *
```



```
      mirror analyzer-ip-address 192.168.2.1 analyzer-virtual-router default
analyzer-udp-port 3000 mirror-id 6789 session-id 6543 (unreachable)

   Referenced by interface(s):
      TUNNEL l2tp:1/msn.pwh.com/1  secure-input policy, statistics disabled
      TUNNEL l2tp:1/msn.pwh.com/1  secure-output policy, statistics disabled
```

Meaning Table 72 on page 298 lists **show secure policy-list** command output fields.

Table 72: show secure policy-list Output Fields

Field Name	Field Description
Policy	Type (IP, IPv6, or L2TP) and name of the policy list
Administrative state	Status of administrative state, enable or disable; set to enable when the policy list is created
Reference count	Number of attachments to interfaces or profiles
Classifier control list	Name of the classifier control list
Mirror analyzer-ip-address	IP address of analyzer device
Analyzer-virtual-router	Analyzer interface virtual router
Analyzer-udp-port	UDP port used to communicate with analyzer device
Mirror-id	Unique identifier of the mirrored session
Session-id	Unique identifier of the user session
Referenced by interface(s)	List of interfaces to which the policy is attached; indicates whether the attachment is at secure input or secure output of interface
Referenced by profile(s)	Not currently supported: always null
Statistics	Not currently supported: always disabled

Related Documentation

- Configuring CLI-Based Packet Mirroring on page 251
- show secure policy-list

Monitoring Information for Secure Policies

Purpose Display failure messages and information for secure policies. This command and the output are visible only to authorized users—the **mirror-enable** command must be enabled before using this command. All normal E Series system log messages are suppressed for packet mirroring-related policy operations.

Action To display information for secure policies:

```
host1# show mirror log
  Time            Mirror-ID      Session-ID   User           Error Status
-----------     -------------- ---------- -------------- --------------
TUE SEP 15          8976           1923       123@abc.com  no secure policies available
2009 18:35:43 UTC
```

Meaning Table 73 on page 299 lists the **show mirror log** command output fields.

Table 73: show mirror log Output Fields

Field Name	Field Description
Time	Day, date, and time of failure
Mirror-ID	Unique identifier of the mirrored session
Session-ID	Unique identifier of the user session
User	User login name
Error Status	Description of the error condition

Related Documentation

- Configuring CLI-Based Packet Mirroring on page 251
- clear mirror log
- show mirror log

Monitoring SNMP Secure Packet Mirroring Traps

Purpose Display configuration information about SNMP traps and trap destinations. The PacketMirror trap category is displayed only when the **mirror enable** command has been configured. The Secure Trap Logging status is displayed only when the **mirror enable** command has been issued and secure audit logs have been configured. Text in bold indicates secure packet mirroring trap configuration information.

Action To display secure packet mirroring traps:

```
host1# show snmp trap
Enabled Categories: CliSecurity,  PacketMirror , Sonet
SNMP authentication failure trap is disabled
Trap Source: FastEthernet 6/0, Trap Source Address:192.168.120.78
Trap Proxy: enabled
Secure Trap Logging is enabled
Global Trap Severity Level: 6 - informational
Address          Security String                  Ver  Port   Trap Categories
---------------  -------------------------------- ---  -----  ----------------
10.1.1.1         host1                            v1     162  Cli
10.12.12.12      secureHost                       v3     162  CliOspf PacketMirror Sonet
```

```
192.168.57.162   host2                             v3     162   Sonet
Address         TrapSeverityFilter  Ping    Maximum    Queue    Queue  Full
                                   TimeOut QueueSize DrainRate discrd methd
--------------- ------------------ ------- --------- --------- -------------
10.1.1.1        5 - notice         1       32        0         dropLastIn
10.12.12.12     2 - critical       1       32        0         dropLastIn
192.168.57.162  2 - critical       1       32        0         dropLastIn
```

Meaning Table 74 on page 300 lists the **show snmp trap** command output fields.

Table 74: show snmp trap Output Fields

Field Name	Field Description
Enabled Categories	Trap categories that are enabled on the router
SNMP authentication failure trap	Enabled or disabled
Trap Source	Interface whose IP address is used as the source address for all SNMP traps
Trap Source Address	IP address used as the source address for all SNMP traps
Trap Proxy	Enabled or disabled
Secure Trap Logging	Enabled or disabled
Global Trap Severity Level	Global severity level filter; if a trap does not meet this severity level, it is discarded
Address	IP address of the trap recipient
Security String	Name of the SNMP community
Ver	SNMP version (v1 or v2) of the SNMP trap packet
Port	UDP port on which the trap recipient accepts traps
Trap Categories	Types of traps that the trap recipient can receive
TrapSeverityFilter	Severity level filter for this SNMP host
Ping TimeOut	Configured ping timeout in minutes
Maximum QueueSize	Maximum number of traps to be kept in the trap queue
Queue DrainRate	Maximum number of traps per second to be sent to the host
Queue Full discrd methd	Method used to discard traps when the queue is full:
dropFirstIn	Oldest trap in the queue is dropped

Table 74: show snmp trap Output Fields *(continued)*

Field Name	Field Description
dropLastIn	Most recent trap is dropped

NOTE: Secure packet-mirroring trap configuration information appears in the Enabled Categories and Trap Categories fields only if the **mirror-enable** command is enabled.

Related Documentation

- Configuring SNMP Secure Packet Mirroring Traps on page 284
- mirror trap-enable
- snmp-server enable traps
- snmp-server host
- snmp-server secure-log
- show mirror trap
- show snmp trap

Monitoring SNMP Secure Audit Logs

Purpose Display output when the secure audit log data is available.

NOTE: The secure audit log data is not preserved across the reboot because secure logs are not stored in the nonvolatile memory. Only the **snmp-server secure-log** command configuration is stored in the nonvolatile memory.

Action To display the contents of the SNMP secure audit log:

```
host1# show snmp secure-log
Agent's Context    LogData
---------------    ---------------------------------------------------------
SnmpRouterAgent1   SNMP Trap, SNMPVer= 3, src=10.27.120.117, dest=10.1.1.1,
reqId=3, errSts=0, errIndx=0, msgID=2, msgMaxSize=1500, msgFlags=0,
msgSecurityModel=3,
contextEngineID=80:00:13:0a:05:00:90:1a:41:45:51:80:00:00:01 ,
securityName=jbond, engineBoots=0, engineTime=0, varCnt=13, Vars:
1.3.6.1.2.1.1.3.0 [1259], 1.3.6.1.6.3.1.1.4.1.0
1],3.6.1.4.1.4874.2.2.77.3.0.3], 1.3.6.1.4.1.4874.2.2.77.3.1.13 [?^K^B
1.3.6.1.4.1.4874.2.2.77.3.1.5 [0], 1.3.6.1.4.1.4874.2.2.77.3.1.4 [0],
1.3.6.1.4.1.4874.2.2.77.3.1.3 [f], 1.3.6.1.4.1.4874.2.2.77.3.1.14 [1],
1.3.6.1.4.1.4874.2.2.77.3.1.1 [1], 1.3.6.1.4.1.4874.2.2.77.3.1.2 [1],
1.3.6.1.4.1.4874.2.2.77.3.1.11 [f], 1.3.6.1.4.1.4874.2.2.77.3.1.12 [1],
1.3.6.1.4.1.4874.2.2.77.3.1.15 [0], 1.3.6.1.4.1.4874.2.2.16.1.3.5.0 [5],
SnmpRouterAgent44   SNMP Trap, SNMPVer= 3, src=10.27.120.117, dest=10.1.1.1,
```

```
reqId=5, errSts=0, errIndx=0, msgID=4, msgMaxSize=1500, msgFlags=0,
msgSecurityModel=3,
contextEngineID=80:00:13:0a:05:00:90:1a:41:45:51:80:00:00:01 ,
securityName=jbond, engineBoots=0, engineTime=0, varCnt=14, Vars:
1.3.6.1.2.1.1.3.0 [1259], 1.3.6.1.6.3.1.1.4.1.0
1],3.6.1.4.1.4874.2.2.77.3.0.1], 1.3.6.1.4.1.4874.2.2.77.3.1.13 [?^K^B
1.3.6.1.4.1.4874.2.2.77.3.1.5 [0], 1.3.6.1.4.1.4874.2.2.77.3.1.4 [0],
1.3.6.1.4.1.4874.2.2.77.3.1.3 [f], 1.3.6.1.4.1.4874.2.2.77.3.1.14 [1],
1.3.6.1.4.1.4874.2.2.77.3.1.10 [f], 1.3.6.1.4.1.4874.2.2.77.3.1.1 [1],
1.3.6.1.4.1.4874.2.2.77.3.1.2 [1], 1.3.6.1.4.1.4874.2.2.77.3.1.6 [0],
1.3.6.1.4.1.4874.2.2.77.3.1.8 [0], 1.3.6.1.4.1.4874.2.2.77.3.1.7 [f],
1.3.6.1.4.1.4874.2.2.16.1.3.5.0 [3],
SnmpRouterAgent22   SNMP Trap, SNMPVer= 3, src=10.27.120.117, dest=10.1.1.1,
reqId=8, errSts=0, errIndx=0, msgID=7, msgMaxSize=1500, msgFlags=3,
msgSecurityModel=3,
contextEngineID=80:00:13:0a:05:00:90:1a:41:45:51:80:00:00:01 ,
securityName=jbond, engineBoots=1, engineTime=8602, varCnt=6, Vars:
1.3.6.1.2.1.1.3.0 [1259], 1.3.6.1.6.3.1.1.4.1.0
1],3.6.1.4.1.4874.2.2.77.3.0.4], 1.3.6.1.4.1.4874.2.2.77.3.1.13 [?^K^B
1.3.6.1.4.1.4874.2.2.77.3.1.9 [192.168.7.120], 1.3.6.1.4.1.4874.2.2.77.3.1.14
[1], 1.3.6.1.4.1.4874.2.2.16.1.3.5.0 [4],
```

Meaning Table 75 on page 302 lists the **show snmp secure-log** command output fields.

Table 75: show snmp secure-log Output Fields

Field Name	Field Description
Agent's Context	Owner of the secure log entry
LogData	Contents of the secure audit log

Related Documentation

- Capturing SNMP Secure Audit Logs on page 285
- snmp-server clear secure-log
- show snmp secure-log

PART 3

Index

Index

Symbols

A

B

C

R

