



JunosE™ Software for E Series™ Broadband Services Routers

Policy Resources Management

Release

14.2.x



Published: 2013-04-01

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

Copyright © 2013, Juniper Networks, Inc. All rights reserved.

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

JunosE™ Software for E Series™ Broadband Services Routers Policy Resources Management
Release 14.2.x
Copyright © 2013, Juniper Networks, Inc.
All rights reserved.

Revision History
April 2013—FRS JunosE 14.2.x

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

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. Junos OS has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

SOFTWARE LICENSE

The terms and conditions for using this software are described in the software license contained in the acknowledgment to your purchase order or, to the extent applicable, to any reseller agreement or end-user purchase agreement executed between you and Juniper Networks. By using this software, you indicate that you understand and agree to be bound by those terms and conditions.

Generally speaking, the software license restricts the manner in which you are permitted to use the software and may contain prohibitions against certain uses. The software license may state conditions under which the license is automatically terminated. You should consult the license for further details.

For complete product documentation, please see the Juniper Networks Web site at www.juniper.net/techpubs.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <http://www.juniper.net/support/eula.html>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

	About the Documentation	vii
	E Series and JunosE Documentation and Release Notes	vii
	Audience	vii
	E Series and JunosE Text and Syntax Conventions	vii
	Obtaining Documentation	ix
	Documentation Feedback	ix
	Requesting Technical Support	ix
	Self-Help Online Tools and Resources	x
	Opening a Case with JTAC	x
Part 1	Overview	
Chapter 1	Policy Resources	3
	Policy Resources Overview	3
	FPGA Hardware Classifiers	6
	Detection of Corruption in the FPGA Statistics for Policies of Subscribers	
	Managed by the SRC Software	7
	Guidelines for Configuring the Capability to Detect Corruption in the FPGA	
	Statistics	8
	Detection of Corruption in the Statistics FPGA for AAA-Based Policy	
	Accounting	10
	Guidelines for Configuring the Capability to Detect Corruption in the Statistics	
	FPGA	11
	Computation of the Interface and Policy Counters for the Detection of Corruption	
	in the FPGA Statistics	13
	Processing the Extra Header in Policy Counters	13
	Processing the Egress Policy Counters	13
	Processing the Received Multicast Packets with Applied Policies	14
	Comparing the Interface and Policy Counters Over Two Polling Intervals	14
	Scenarios for the Detection of Corruption in the FPGA Statistics and the	
	Determination of the Threshold	15
	Bit Flip in Policy Counters	15
	Bit Flip in Interface Counters	16
	Reattachment of a Policy to an Interface	16
	Bit Flip in Policy Counters When a Policy is Attached to an Interface for the	
	First Time	17
	System Operations When Corrupted FPGA Statistics Is Detected	17
	CAM Hardware Classifiers Overview	18
	Size Limit for IP and IPv6 CAM Hardware Classifiers	19
	IP Classifiers and Size Limits	20
	IPv6 Classifiers and Size Limits	22

	CAM Hardware Classifiers and Interface Attachment Resources	25
	Range Vector Hardware Classifiers and Interface Attachment Resources	25
	Performance Impact and Scalability Considerations	25
	Performance Impact	25
	Scalability Considerations	26
	CAM Device Block Size and CAM Entry Allocation	26
	Number of CAM Entries Per Allocation and Free Entries	26
	Software Classifiers Overview	29
	Interface Attachment Resources Overview	30
Part 2	Configuration	
Chapter 2	Configuration Tasks for Policy Resources Management	35
	Creating and Attaching a Policy with IP Classifiers	35
	Enabling the Policy Resources Exhaustion Trap	37
	Configuring the Capability to Detect Corruption in the FPGA Statistics for Policies Managed by the SRC Software	38
Chapter 3	Examples	41
	Examples: Variable-Sized CAM Classification for IPv6 Policies	41
	144-bit IPv6 Classification Example	42
	288-bit IPv6 Classification Example	43
	576-bit IPv6 Classification Example	43
	Example: Computation of the Threshold Value by Using Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics	45
	Statistics Calculation for Incoming Packets	45
	Statistics Calculation for Outgoing Packets	46
Part 3	Administration	
Chapter 4	Monitoring Task for Policy Resources Management	51
	Monitoring the Detection of Corrupted FPGA Statistics Settings	51
	Displaying the Slot Numbers with Corrupted Statistics FPGA for AAA-Based Policy Accounting	52
	Monitoring the Utilization of Interface Attachment Resources	52
	Monitoring the Status of a Policy Resources Trap	54
Part 4	Index	
	Index	59

List of Tables

	About the Documentation	vii
	Table 1: Notice Icons	viii
	Table 2: Text and Syntax Conventions	viii
Part 1	Overview	
Chapter 1	Policy Resources	3
	Table 3: Classifier Support (OC48/STM16, GE-2, GE-HDE, ES2 4G, ES2 10G, and ES2 10G Uplink Line Modules)	4
	Table 4: Classifier Support (All Line Modules Except OC48/STM16, GE-2, GE-HDE, ES2 4G, ES2 10G, and ES2 10G Uplink)	5
	Table 5: Interface and Policy Statistics When a Bit Flip Occurs in Policy Counters	15
	Table 6: Interface and Policy Statistics When a Bit Flip Occurs in Interface Counters	16
	Table 7: Interface and Policy Statistics When a Policy is Reapplied to an Interface	16
	Table 8: Interface and Policy Statistics When a Bit Flip Occurs when a Policy is Attached to an Interface for the First Time	17
	Table 9: Size Limit of Individual IP Classifiers	20
	Table 10: Size Limit of Combined IP Classifiers	21
	Table 11: Size Limit of Individual IPv6 Classifiers	22
	Table 12: Size Limit of Combined IPv6 Classifiers	23
	Table 13: Maximum Policies with One Classifier per Policy for GE-2 LMs	27
	Table 14: Maximum Policies with Four Classifiers per Policy for GE-2 LMs	28
	Table 15: Resource Consumption	30
Part 2	Configuration	
Chapter 2	Configuration Tasks for Policy Resources Management	35
	Table 16: Classification Fields for Example 1	36
	Table 17: Classification Fields for Example 2	37
Chapter 3	Examples	41
	Table 18: IPv6 Classification Fields for a 144-bit CAM Entry	42
	Table 19: IPv6 Classification Fields for a 288-bit CAM Entry	43
	Table 20: IPv6 Classification Fields for a 576-bit CAM Entry	44
Part 3	Administration	
Chapter 4	Monitoring Task for Policy Resources Management	51
	Table 21: show fpga-stats-monitoring Output Fields	51

Table 22: show policy-resources slot Output Fields	54
--	----

About the Documentation

- E Series and JunosE Documentation and Release Notes on page vii
- Audience on page vii
- E Series and JunosE Text and Syntax Conventions on page vii
- Obtaining Documentation on page ix
- Documentation Feedback on page ix
- Requesting Technical Support on page ix

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 viii 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 viii 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.	<ul style="list-style-type: none"> 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)
<i>Italic text like this</i>	<ul style="list-style-type: none"> Emphasizes words. Identifies variables. Identifies chapter, appendix, and book names. 	<ul style="list-style-type: none"> There are two levels of access: <i>user</i> and <i>privileged</i>. <i>clusterId</i>, <i>ipAddress</i>. <i>Appendix A, System Specifications</i>
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
<i>Italic text like this</i>	Represents variables.	<i>mask</i> , <i>accessListName</i>

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 at <http://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

Overview

- [Policy Resources on page 3](#)

CHAPTER 1

Policy Resources

- Policy Resources Overview on page 3
- FPGA Hardware Classifiers on page 6
- Detection of Corruption in the FPGA Statistics for Policies of Subscribers Managed by the SRC Software on page 7
- Detection of Corruption in the Statistics FPGA for AAA-Based Policy Accounting on page 10
- Computation of the Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 13
- Scenarios for the Detection of Corruption in the FPGA Statistics and the Determination of the Threshold on page 15
- System Operations When Corrupted FPGA Statistics Is Detected on page 17
- CAM Hardware Classifiers Overview on page 18
- Size Limit for IP and IPv6 CAM Hardware Classifiers on page 19
- CAM Hardware Classifiers and Interface Attachment Resources on page 25
- Range Vector Hardware Classifiers and Interface Attachment Resources on page 25
- Performance Impact and Scalability Considerations on page 25
- Software Classifiers Overview on page 29
- Interface Attachment Resources Overview on page 30

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 30](#) 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 3 on page 4](#) lists the classifiers supported on OC48/STM16, GE-2, GE-HDE, ES2 4G, ES2 10G, and ES2 10G Uplink line modules; [Table 4 on page 5](#) lists the classifiers supported on all other line modules.

Table 3: 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)	–	<ul style="list-style-type: none"> • Color • Traffic class • User packet class
Frame Relay	Not supported	<ul style="list-style-type: none"> • DE bit
GRE tunnels	Not supported	<ul style="list-style-type: none"> • ToS
IP	<ul style="list-style-type: none"> • 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 3: 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • EXP
VLAN	Not supported	<ul style="list-style-type: none"> • User priority

Table 4: 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	–	<ul style="list-style-type: none"> • Color • Traffic class • User packet class
Frame Relay	Not supported	<ul style="list-style-type: none"> • DE bit
GRE tunnels	Not supported	<ul style="list-style-type: none"> • ToS
IP	<ul style="list-style-type: none"> • Destination address • Destination port • ICMP type and code • IGMP type • Protocol • Source address • Source port 	<ul style="list-style-type: none"> • Destination route class • IP flags • IP fragmentation • Local • Source route class • TCP flags • ToS

Table 4: 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	<ul style="list-style-type: none"> • Destination address • Destination port • ICMPv6 type and code • Protocol • Source address • Source port 	<ul style="list-style-type: none"> • Destination route class • Local • Source route class • TC field • TCP flags
MPLS	Not supported	<ul style="list-style-type: none"> • EXP
VLAN	Not supported	<ul style="list-style-type: none"> • User priority

- Related Documentation**
- [CAM Hardware Classifiers and Interface Attachment Resources on page 25](#)
 - [FPGA Hardware Classifiers on page 6](#)
 - [Interface Attachment Resources Overview on page 30](#)

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 3](#) 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 30](#)
- [Policy Resources Overview on page 3](#)

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 13](#)
- [Example: Computation of the Threshold Value by Using Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 45](#)
- [Scenarios for the Detection of Corruption in the FPGA Statistics and the Determination of the Threshold on page 15](#)
- [Configuring the Capability to Detect Corruption in the FPGA Statistics for Policies Managed by the SRC Software on page 38](#)
- [Monitoring the Detection of Corrupted FPGA Statistics Settings on page 51](#)

Detection of Corruption in the Statistics FPGA for AAA-Based Policy Accounting

When a bit flip occurs in the RAM of the statistics field-programmable gate array (FPGA) of a router, the router may transmit erroneous and inconsistent policy accounting details for IPv4 and IPv6 subscribers to a RADIUS server through authentication, authorization, and accounting (AAA). You can resolve the problem of inconsistent policy accounting details only by replacing the defective hardware.

To avoid accounting wrong statistics before replacing the defective hardware, you can configure a software-detection mechanism that identifies corruption in the statistics FPGA and prevents the router from sending erroneous policy accounting details to the RADIUS server. To detect incorrect policy accounting details, the software-detection mechanism compares the following statistical counters against a threshold value:

- Packets processed by an interface
- Packets for which policies are attached to the interface

The detection mechanism performs various computations in the policy and interface counters. For more information about the computations performed in the counters for detecting the corruption in the FPGA, see [“Computation of the Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics” on page 13](#).

If the difference between the interface counters and policy counters for ingress or egress policies collected over two polling intervals is equal to or greater than the specified threshold value, a corruption is detected in the statistics FPGA and the policy accounting details are not forwarded to the RADIUS 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 statistics FPGA and the policy accounting details are sent to the RADIUS server. For more information about the computation of difference between the interface counters and policy counters, see [“Example: Computation of the Threshold Value by Using Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics” on page 45](#)

You can use the **fpga-stats-monitoring-enable** command in Privileged Exec mode to enable the capability to detect corruption in the statistics FPGA and prevent the transmission of incorrect policy accounting details to the RADIUS server. You can 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 statistics FPGA. The threshold value is compared with the difference of the interface and policy counter values (for ingress and egress policies) collected over two consecutive polling periods. For more information about the configuration steps to be performed for detecting corruption in the statistics FPGA, see [“Configuring the Capability to Detect Corruption in the FPGA Statistics for Policies Managed by the SRC Software” on page 38](#).



NOTE: The **fpga-stats-monitoring-enable**, **fpga-stats-monitoring threshold**, and **fpga-stats-monitoring trap enable** commands are commonly used for configuring the capability to detect corruption in the statistics FPGA for policies managed by the SRC software and for AAA-based policy accounting.

When AAA polls for policy accounting statistics to update RADIUS accounting messages, the software-detection mechanism is triggered before the router sends the policy accounting details to the RADIUS server. If the accounting interim is enabled on the router, AAA informs Point-to-Point Protocol (PPP) to poll accounting requests at the frequency of the configured accounting interim period when a subscriber logs in. The detection mechanism is triggered when the router receives the following requests from the RADIUS server:

- Accounting start request
- Interim accounting request
- Accounting stop request

When corruption is detected in the statistics FPGA, AAA logs out the affected subscribers, prevents new subscribers from establishing sessions on the defective slot, and generates a system log message and SNMP trap containing information about all affected subscribers—such as interface names, usernames, and details about defective slots. The SNMP trap is sent to an external device. AAA maintains the defective slot information until the defective hardware is replaced.

Guidelines for Configuring the Capability to Detect Corruption in the Statistics FPGA

Keep the following points in mind when you configure the capability to detect corruption in the FPGA statistics. You must specify a threshold value to determine discrepancies in the statistics:

- When a subscriber attempts to establish a session over a defective slot where corruption in the statistics FPGA is detected, the subscriber is not allowed to log in.
- The configuration settings related to the detection of corruption in the statistics FPGA 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 statistics FPGA is detected on ES2 10G ADV line modules (LMs) or ES2 4G LMs with Service input/output adapters (IOAs), and this validation of the state of the statistics FPGA is not performed on the access interfaces. This method of detection occurs because the interface statistics are maintained only in the Service IOA.
- When corruption is detected on ES2 4G LMs or ES2 10G ADV LMs with ES2-ES1 Service IOAs, 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 the subscribers log out. You must remove stateful switchover configuration on such LMs to enable the secondary line module to handle new subscriber logins.
- If Line Card Redundancy (LCR) and Line Card High Availability (LCHA) are not enabled on the defective hardware, AAA does not disconnect the affected subscribers but prevents any new subscribers from being connected to the defective slot when corruption is detected.

- If LCR is enabled on the defective hardware, AAA terminates all affected subscribers and the standby slot takes over when corruption is detected.
- If LCHA is enabled on the defective hardware, the router unconfigures the LCHA group when corruption is detected:
 - If the configured primary Tunnel Service line module (TSM) is active, the standby TSM is reloaded to accept new subscribers while maintaining the subscribers existing on the primary TSM.
 - If the configured secondary TSM is active, the primary and standby TSM cards are reloaded to terminate all subscribers.



NOTE: The maximum number of supported interfaces should be set as zero on the current active TSM card to achieve load balancing.

- Information about defective slots is unavailable across stateful SRP switchover operations. Therefore, if subscribers attempt to log in to a slot, which was determined to be corrupted before the reset, they are permitted to log in until the detection mechanism again classifies the slot as defective after the reset.
- The detection mechanism cannot accurately calculate policy statistics when the ingress or egress traffic does not match any of the classified rules configured within a policy. To avoid this discrepancy, the unaccounted traffic should be matched with a default classifier group.
- The detection mechanism cannot detect the statistics corruption caused by bit flips in the least significant bits, which causes the difference between the interface and policy counters for ingress or egress policies collected over two polling intervals to be lower than the configured threshold value.
- The detection mechanism is not supported for the Hierarchical Rate Limit Profile (H-RLP).

**Related
Documentation**

- [Scenarios for the Detection of Corruption in the FPGA Statistics and the Determination of the Threshold on page 15](#)
- [Monitoring the Detection of Corrupted FPGA Statistics Settings on page 51](#)
- [Displaying the Slot Numbers with Corrupted Statistics FPGA for AAA-Based Policy Accounting on page 52](#)
- `fpga-stats-monitoring-enable`
- `fpga-stats-monitoring threshold`
- `fpga-stats-monitoring trap enable`

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

When a bit flip occurs and causes a negative value for the egress policy counter in bytes and the ingress policy counter in bytes, a system log message is generated at the error severity level.

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.

When a bit flip occurs and causes a negative value for the interface counters, which are calculated as a measure of the number of packets and bytes of traffic arriving on an interface, a system log message is generated at the error severity level.

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 7](#)
- [Detection of Corruption in the Statistics FPGA for AAA-Based Policy Accounting on page 10](#)
- [Example: Computation of the Threshold Value by Using Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 45](#)
- [Scenarios for the Detection of Corruption in the FPGA Statistics and the Determination of the Threshold on page 15](#)
- [Configuring the Capability to Detect Corruption in the FPGA Statistics for Policies Managed by the SRC Software on page 38](#)
- [Monitoring the Detection of Corrupted FPGA Statistics Settings on page 51](#)

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 5: 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 6: 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 7: 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 8: 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 7](#)
- [Detection of Corruption in the Statistics FPGA for AAA-Based Policy Accounting on page 10](#)
- [Computation of the Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 13](#)
- [Example: Computation of the Threshold Value by Using Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 45](#)
- [Configuring the Capability to Detect Corruption in the FPGA Statistics for Policies Managed by the SRC Software on page 38](#)
- [Monitoring the Detection of Corrupted FPGA Statistics Settings on page 51](#)

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 7](#)
- [Computation of the Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 13](#)
- [Configuring the Capability to Detect Corruption in the FPGA Statistics for Policies Managed by the SRC Software on page 38](#)
- [Monitoring the Detection of Corrupted FPGA Statistics Settings on page 51](#)
- `fpga-stats-monitoring trap enable`

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 3](#) 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 25](#)
- [Interface Attachment Resources Overview on page 30](#)
- [Policy Resources Overview on page 3](#)
- [Size Limit for IP and IPv6 CAM Hardware Classifiers on page 19](#)

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 9 on page 20 lists all IP classifiers and the size limit of each classifier entry.

Table 9: 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

Table 9: Size Limit of Individual IP Classifiers (*continued*)

IP Classifier	Size Limit (Bits)
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 10 on page 21](#) 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 10 on page 21](#) 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 10: 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	—

Table 10: Size Limit of Combined IP Classifiers (*continued*)

IP Classifier Entry Combination	Size Limit (Bits)	Rule
[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.
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 11 on page 22 lists all IPv6 and the size limit of each classifier entry.

Table 11: Size Limit of Individual IPv6 Classifiers

IPv6 Classifier Entry	Size Limit (Bits)
Color	2

Table 11: Size Limit of Individual IPv6 Classifiers (*continued*)

IPv6 Classifier Entry	Size Limit (Bits)
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 12 on page 23](#) 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 12 on page 23](#) 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 12: 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	—

Table 12: Size Limit of Combined IPv6 Classifiers (*continued*)

IPv6 Classifier Entry Combination	Size Limit (Bits)	Rule
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	–
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 25](#)
 - [CAM Hardware Classifiers Overview on page 18](#)

[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 18](#)
 - [Interface Attachment Resources Overview on page 30](#)

[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 30](#)

[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 41](#), 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 26](#)
- [Number of CAM Entries Per Allocation and Free Entries on page 26](#)

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 13 on page 27 lists the maximum policies supported with variable length IPv6 CAM classification and one classifier per policy. The following note is referred to in Table 13 on page 27.

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 13: 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 13: 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 14 on page 28 lists the maximum policies supported with variable length IPv6 CAM classification and four classifiers per policy.

Table 14: 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 CLACLs	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 41](#)
- [Size Limit for IP and IPv6 CAM Hardware Classifiers on page 19](#)

Software Classifiers Overview

An E Series router supports a variety of software classifiers, depending on the type of interface. “[Policy Resources Overview](#)” on [page 3](#) 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 15 on page 30](#).

Table 15: Resource Consumption

Number of Resources Consumed	Classifier Category
1 hardware	<ul style="list-style-type: none"> • Protocol • Destination address • Source address
1 software	Color
1 software	User-packet-class

Related Documentation

- [Policy Resources Overview on page 3](#)

Interface Attachment Resources Overview

JunosE Software allocates interface attachment resources when policies are attached to interfaces—that is, 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 IPv4 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 policy 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.

You can use the **show policy-resources slot** command to monitor the policy resource consumption on a specific slot or on all slots to which the policies are attached. The **show policy-resources slot** command displays the type of policy resource, number of policy resources supported, number of policy resources consumed, number of policy resources that are free, and direction type for each resource type.



NOTE: The current system maximums of the E Series router are displayed in the **show policy-resources slot** command output as the number of policy resources supported by the line module. For the current system maximums, see *Appendix A, System Maximums* in the *JunosE Release Notes*.

**Related
Documentation**

- [Policy Resources Overview on page 3](#)
- [Enabling the Policy Resources Exhaustion Trap on page 37](#)
- [Monitoring the Utilization of Interface Attachment Resources on page 52](#)
- `show policy-resources slot`

PART 2

Configuration

- [Configuration Tasks for Policy Resources Management on page 35](#)
- [Examples on page 41](#)

CHAPTER 2

Configuration Tasks for Policy Resources Management

- [Creating and Attaching a Policy with IP Classifiers on page 35](#)
- [Enabling the Policy Resources Exhaustion Trap on page 37](#)
- [Configuring the Capability to Detect Corruption in the FPGA Statistics for Policies Managed by the SRC Software on page 38](#)

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 16 on page 36 lists the active classifiers in the policy named ipPol and the size of each classifier.

Table 16: 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 17 on page 37 lists the active classifiers in the policy named ipPol and the size of each classifier.

Table 17: 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 30](#)

Enabling the Policy Resources Exhaustion Trap

You can use the **policy-resource-exhaustion trap enable** command to enable the policy resource exhaustion trap to send an SNMP trap notification to the SNMP manager when the policy resources are exhausted. An SNMP trap notification is sent for the following policy resource types when they exceed their maximum limits as per the system maximums:

- ipPolicyDescriptor (0)
- l2PolicyDescriptor (1)
- rateLimiter (2)
- statsBlock (3)

- camTotalEntries (4)
- softwareLookupRuleSet (5)
- parentGroup (6)

For more information about the system maximums, see *Appendix A, System Maximums* in the *JunosE Release Notes*.

An SNMP trap notification contains the following information:

- Slot number in which the policy resource is exhausted
- Type of the exhausted policy resource
- Direction of the exhausted policy resource: ingress, egress, or both.

To enable the policy resource exhaustion trap to send an SNMP trap notification:

- Issue the **policy-resource-exhaustion trap enable** command in Privileged Exec mode.

host1#policy-resource-exhaustion trap enable

Use the **no** version to disable the policy resource exhaustion trap from sending an SNMP trap notification when the policy resources are exhausted.

**Related
Documentation**

- [Policy Resources Overview on page 3](#)
- [Interface Attachment Resources Overview on page 30](#)
- [Monitoring the Status of a Policy Resources Trap on page 54](#)
- `policy-resource-exhaustion-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#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 7](#)
- [Detection of Corruption in the Statistics FPGA for AAA-Based Policy Accounting on page 10](#)
- [Computation of the Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 13](#)
- [Example: Computation of the Threshold Value by Using Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 45](#)
- [Scenarios for the Detection of Corruption in the FPGA Statistics and the Determination of the Threshold on page 15](#)
- [Monitoring the Detection of Corrupted FPGA Statistics Settings on page 51](#)

CHAPTER 3

Examples

- [Examples: Variable-Sized CAM Classification for IPv6 Policies on page 41](#)
- [Example: Computation of the Threshold Value by Using Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 45](#)

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 18 on page 42](#) lists the active classifiers in the policy named ipv6Pol and the size of each classifier.

Table 18: 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 to any DA

```
host1(config)#ipv6 classifier-list sourceCLACL source-address 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 19 on page 43](#) lists the active classifiers in the policy named `ipv6Pol` and the size of each classifier.

Table 19: IPv6 Classification Fields for a 288-bit CAM Entry

Classifiers	Size (Bits)
Source address (first word)	32
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 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 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 20 on page 44](#) lists the active classifiers in the policy named ipv6Pol and the size of each classifier.

Table 20: IPv6 Classification Fields for a 576-bit CAM Entry

Classifiers	Size (Bits)
Source address (first word)	32
Source address (second word)	32
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).

Although each CAM block can contain 2000 576-bit CAM entries, the hardware considers the CAM block to contain 8000 144-bit entries that are clustered together as 4 sets each of 144-bit entries. In this example, although three 576-bit entries are used, one for each classifier in the policy, these entries are essentially three sets of 144-bit entries from the 576-bit CAM block. The sum of the unique classification fields in the policy determines the group from which the CAM resources are allocated for the entire policy (the 144-bit, the 288-bit, or the 576-bit group).

- Related Documentation**
- [CAM Hardware Classifiers Overview on page 18](#)
 - [Performance Impact and Scalability Considerations on page 25](#)
 - [Size Limit for IP and IPv6 CAM Hardware Classifiers on page 19](#)

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 = $\{(\text{Ingress Interface Packet Counter} - \text{Ingress Policy Packet Counter})\} = \{(775,132 - 775,132)\} = 0$

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 7](#)
- [Detection of Corruption in the Statistics FPGA for AAA-Based Policy Accounting on page 10](#)
- [Computation of the Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 13](#)
- [Scenarios for the Detection of Corruption in the FPGA Statistics and the Determination of the Threshold on page 15](#)

- [Configuring the Capability to Detect Corruption in the FPGA Statistics for Policies Managed by the SRC Software on page 38](#)
- [Monitoring the Detection of Corrupted FPGA Statistics Settings on page 51](#)

PART 3

Administration

- [Monitoring Task for Policy Resources Management on page 51](#)

CHAPTER 4

Monitoring Task for Policy Resources Management

- [Monitoring the Detection of Corrupted FPGA Statistics Settings on page 51](#)
- [Displaying the Slot Numbers with Corrupted Statistics FPGA for AAA-Based Policy Accounting on page 52](#)
- [Monitoring the Utilization of Interface Attachment Resources on page 52](#)
- [Monitoring the Status of a Policy Resources Trap on page 54](#)

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 21 on page 51](#) lists the **show fpga-stats-monitoring** command output fields.

**Table 21: 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 7](#)
  - [Computation of the Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 13](#)
  - [Example: Computation of the Threshold Value by Using Interface and Policy Counters for the Detection of Corruption in the FPGA Statistics on page 45](#)
  - [Scenarios for the Detection of Corruption in the FPGA Statistics and the Determination of the Threshold on page 15](#)
  - [Configuring the Capability to Detect Corruption in the FPGA Statistics for Policies Managed by the SRC Software on page 38](#)
  - [System Operations When Corrupted FPGA Statistics Is Detected on page 17](#)

## Displaying the Slot Numbers with Corrupted Statistics FPGA for AAA-Based Policy Accounting

**Purpose** Display the list of slot numbers in which statistics FPGA corruption is detected while reporting policy accounting to the RADIUS server through AAA.

**Action** To display corrupted slot numbers:

```
host1#show aaa-corrupted-slots
Corrupted Slot List :
1
2
```

- Related Documentation**
- [Detection of Corruption in the Statistics FPGA for AAA-Based Policy Accounting on page 10](#)
  - [show aaa-corrupted-slots](#)

## Monitoring the Utilization of Interface Attachment Resources

**Purpose** Display policy resource consumption information about a particular slot or all slots to which the policies are attached.

**Action** To display policy resource consumption information about all slots to which the policies are attached:

```
host1#show policy-resources slot all
```

Slot number 4

| Resource     | Supported | Used  | Remaining | Direction |
|--------------|-----------|-------|-----------|-----------|
| -----        | -----     | ----- | -----     | -----     |
| IpPolicyDesc | 16000     | 0     | 16000     | Ingress   |
| IpPolicyDesc | 8191      | 0     | 8191      | Egress    |
| L2PolicyDesc | 8191      | 0     | 8191      | Ingress   |
| L2PolicyDesc | 8191      | 0     | 8191      | Egress    |
| RateLimiter  | 24575     | 0     | 24575     | Ingress   |

|                 |        |   |        |         |
|-----------------|--------|---|--------|---------|
| RateLimiter     | 24575  | 0 | 24575  | Egress  |
| StatsBlock      | 262140 | 0 | 262140 | Ingress |
| StatsBlock      | 262140 | 0 | 262140 | Egress  |
| SwLookupRuleSet | 16383  | 0 | 16383  | NA      |
| ParentGroups    | 8191   | 0 | 8191   | Ingress |
| ParentGroups    | 8191   | 0 | 8191   | Egress  |

Slot number 5

| Resource        | Supported | Used  | Remaining | Direction |
|-----------------|-----------|-------|-----------|-----------|
| -----           | -----     | ----- | -----     | -----     |
| IpPolicyDesc    | 16000     | 0     | 16000     | Ingress   |
| IpPolicyDesc    | 8191      | 0     | 8191      | Egress    |
| L2PolicyDesc    | 8191      | 0     | 8191      | Ingress   |
| L2PolicyDesc    | 8191      | 0     | 8191      | Egress    |
| RateLimiter     | 24575     | 0     | 24575     | Ingress   |
| RateLimiter     | 24575     | 0     | 24575     | Egress    |
| StatsBlock      | 262140    | 0     | 262140    | Ingress   |
| StatsBlock      | 262140    | 0     | 262140    | Egress    |
| SwLookupRuleSet | 16383     | 0     | 16383     | NA        |
| ParentGroups    | 24575     | 0     | 24575     | Ingress   |
| ParentGroups    | 24575     | 0     | 24575     | Egress    |

Slot number 6

| Resource        | Supported | Used  | Remaining | Direction |
|-----------------|-----------|-------|-----------|-----------|
| -----           | -----     | ----- | -----     | -----     |
| IpPolicyDesc    | 16000     | 0     | 16000     | Ingress   |
| IpPolicyDesc    | 8191      | 0     | 8191      | Egress    |
| L2PolicyDesc    | 8191      | 0     | 8191      | Ingress   |
| L2PolicyDesc    | 8191      | 0     | 8191      | Egress    |
| RateLimiter     | 24575     | 0     | 24575     | Ingress   |
| RateLimiter     | 24575     | 0     | 24575     | Egress    |
| StatsBlock      | 262140    | 0     | 262140    | Ingress   |
| StatsBlock      | 262140    | 0     | 262140    | Egress    |
| SwLookupRuleSet | 16383     | 0     | 16383     | NA        |
| ParentGroups    | 8191      | 0     | 8191      | Ingress   |
| ParentGroups    | 8191      | 0     | 8191      | Egress    |

To display policy resource consumption information about a particular slot to which the policies are attached:

```
host1#show policy-resources slot 5
```

Slot number 5

| Resource        | Supported | Used  | Remaining | Direction |
|-----------------|-----------|-------|-----------|-----------|
| -----           | -----     | ----- | -----     | -----     |
| IpPolicyDesc    | 16000     | 0     | 16000     | Ingress   |
| IpPolicyDesc    | 8191      | 0     | 8191      | Egress    |
| L2PolicyDesc    | 8191      | 0     | 8191      | Ingress   |
| L2PolicyDesc    | 8191      | 0     | 8191      | Egress    |
| RateLimiter     | 24575     | 0     | 24575     | Ingress   |
| RateLimiter     | 24575     | 0     | 24575     | Egress    |
| StatsBlock      | 262140    | 0     | 262140    | Ingress   |
| StatsBlock      | 262140    | 0     | 262140    | Egress    |
| SwLookupRuleSet | 16383     | 0     | 16383     | NA        |

|              |       |   |       |         |
|--------------|-------|---|-------|---------|
| ParentGroups | 24575 | 0 | 24575 | Ingress |
| ParentGroups | 24575 | 0 | 24575 | Egress  |

**Meaning** Table 22 on page 54 lists the output fields for the **show policy-resources slot** command.

**Table 22: show policy-resources slot Output Fields**

| Field Name  | Field Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Slot number | Number of the chassis slot                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Resource    | Type of policy resources: <ul style="list-style-type: none"> <li>• IpPolicyDesc – L3 policy interface attachments per line module</li> <li>• L2PolicyDesc – L2 policy interface attachments per line module</li> <li>• RateLimiter – Rate limiters per line module</li> <li>• StatsBlock – Policy statistics blocks per line module</li> <li>• CamTotalEntries – Policy classification (CLACL) entries per line module</li> <li>• SwLookupRuleSet – Software lookup blocks per line module</li> <li>• ParentGroups – Parent groups per line module</li> </ul> |
| Supported   | Number of policy resources supported by the line module                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Used        | Number of policy resources consumed by the line module                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Remaining   | Number of policy resources that are free in the line module                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Direction   | Direction type of the policy resources: <ul style="list-style-type: none"> <li>• Ingress – Input policy resource attached to the interface</li> <li>• Egress – Output policy resource attached to the interface</li> <li>• Both – Input and output policy resources attached to the interface</li> </ul>                                                                                                                                                                                                                                                      |

**Related Documentation**

- [Interface Attachment Resources Overview on page 30](#)
- `show policy-resources slot`

## Monitoring the Status of a Policy Resources Trap

**Purpose** Display the status of a policy resources trap.



**Action** To display the status of a policy resources trap:

```
host1#show policy-resources trap
```

```
Policy Resource exhaustion trap is enabled
```

- Related Documentation**
- [Enabling the Policy Resources Exhaustion Trap on page 37](#)
  - `show policy-resources trap`



## PART 4

# Index

- [Index on page 59](#)



# Index

## Symbols

- 144-bit CAM entries
  - example of IPv6 classifiers in a policy supported by CAM classifiers.....42
- 288-bit CAM entries
  - example of IPv6 classifiers in a policy supported by CAM classifiers.....42
- 576-bit CAM entries
  - example of IPv6 classifiers in a policy supported by CAM classifiers.....42

## A

- attachment of IPv6 policies
  - to an interface
    - with CAM entries greater than 128 bits.....41

## C

- CAM blocks
  - containing equal-sized CAM entries.....26
  - support for variable-sized entries
    - .....41
  - utilization for CAM entries
    - in an IPv4 policy, example.....26
    - in an IPv6 policy, example.....26
- CAM device block size
  - division to hold CAM entries
    - example.....26
- CAM entries
  - configured on line modules
    - calculation of CAM bit size.....41
  - division into blocks to store policies
    - example for GE-2 LMS.....26
  - number per allocation
    - formula for scaling limits on GE-2 LMS.....26
    - formula for scaling limits on GE-HDE LMS.....26

- obtaining length dynamically, three types
  - between 128 and 272 bits.....41
  - between 272 and 336 bits.....41
  - less than 128 bits.....41
- using the optimum size
  - for IPv6 policies.....41
- CAM hardware classifiers
  - three types of CAM entries supported
    - on line modules of E Series Broadband Services Routers.....41
  - variables-sized entries
    - for IPv6 policies.....41
- CAM resources
  - and variable-sized CAM entries
    - factors used to determine availability.....26
- classifier
  - CAM hardware.....3, 18
  - consumption.....29
  - FPGA hardware.....3, 6
  - hardware.....3, 6
  - line module support.....3, 4, 5
  - policy consumption.....3, 29
  - software.....3, 29
- conventions
  - notice icons.....vii
  - text and syntax.....viii
- customer support.....ix
  - contacting JTAC.....ix
- D
  - default classifier
    - allocation of CAM blocks
      - determined by the CAM entry size.....26
    - consumption of CAM blocks
      - same as the size computed for the policy.....26
- documentation set
  - comments on.....ix
- F
  - FPGA Statistics
    - computation of interface and policy counters.....13
    - computation of threshold value.....45
    - detection of corruption.....15, 45

|                                                                                                 |  |
|-------------------------------------------------------------------------------------------------|--|
| detection of corruption and<br>handling of sessions for AAA<br>subscribers.....17               |  |
| line module redundancy.....17                                                                   |  |
| stateful line module switchover.....17                                                          |  |
| detection of hardware corruption.....7, 10, 13, 38                                              |  |
| system operations when<br>corruption is detected.....17                                         |  |
| FPGA Statistics settings                                                                        |  |
| detection of corruption.....51                                                                  |  |
| detection of corruption and<br>generation of SNMP traps.....17                                  |  |
| monitoring.....51                                                                               |  |
| <b>G</b>                                                                                        |  |
| GE-2 line modules                                                                               |  |
| formula for scaling numbers of<br>CAM entries.....26                                            |  |
| GE-HDE line modules                                                                             |  |
| formula for scaling limits<br>CAM entries.....26                                                |  |
| <b>I</b>                                                                                        |  |
| IP policies                                                                                     |  |
| scalability improvement for<br>using optimum CAM entry size.....41                              |  |
| IPv4 classifier                                                                                 |  |
| number of bits consumed<br>for CAM entries.....26                                               |  |
| IPv6 classification                                                                             |  |
| 144-bit CAM entries                                                                             |  |
| active classifiers in the example and size<br>of each.....42                                    |  |
| example for a policy attachment.....42                                                          |  |
| 288-bit CAM entries                                                                             |  |
| active classifiers in the example and size<br>of each.....42                                    |  |
| example for a policy attachment.....42                                                          |  |
| 576-bit CAM entries                                                                             |  |
| active classifiers in the example and size<br>of each.....42                                    |  |
| example for a policy attachment.....42                                                          |  |
| number of lookups to the CAM<br>hardware.....25                                                 |  |
| determination of fields in the classifier<br>using policy definition.....41                     |  |
| minimum and maximum entry sizes<br>for IPv6 header CAM entries.....41                           |  |
| support for traffic greater than 128 bits<br>.....41                                            |  |
| variable-sized CAM entries<br>and available CAM resources.....25                                |  |
| maximum policies supported with four<br>classifier per policy.....27                            |  |
| maximum policies supported with one<br>classifier per policy.....27                             |  |
| performance impact.....25                                                                       |  |
| IPv6 classifier See IPv6 classification                                                         |  |
| IPv6 policies                                                                                   |  |
| with four classifiers per policy<br>maximum supported with variable-sized<br>CAM entries.....27 |  |
| with one classifier per policy<br>maximum supported with variable-sized<br>CAM entries.....27   |  |
| IPv6 policy definition                                                                          |  |
| sum of classification fields in the<br>and variable-sized CAM entries.....26                    |  |
| <b>M</b>                                                                                        |  |
| manuals                                                                                         |  |
| comments on.....ix                                                                              |  |
| Monitoring Policy Management                                                                    |  |
| monitoring the status of policy resources<br>trap.....54                                        |  |
| monitoring the utilization of policy resources<br>for interface attachments.....52              |  |
| <b>N</b>                                                                                        |  |
| notice icons.....vii                                                                            |  |
| <b>P</b>                                                                                        |  |
| policy management                                                                               |  |
| classifier resources.....6                                                                      |  |
| policy management commands                                                                      |  |
| policy-resource-exhaustion trap enable.....37                                                   |  |
| show policy-resources slot.....52                                                               |  |
| show policy-resources trap.....54                                                               |  |
| <b>S</b>                                                                                        |  |
| show commands                                                                                   |  |
| show aaa-corrupted-slots.....52                                                                 |  |
| support, technical See technical support                                                        |  |
| <b>T</b>                                                                                        |  |
| technical support                                                                               |  |
| contacting JTAC.....ix                                                                          |  |

text and syntax conventions.....viii

## V

variable length IPv6 classifiers

examples of

supported CAM entries.....42

variable-sized CAM entries

144-bit size

active classifiers in the example

policy.....42

creation and attachment of an IPv6 policy,

example.....42

size of each classifier in the IPv6 policy

example.....42

288-bit size

active classifiers in the example

policy.....42

creation and attachment of an IPv6 policy,

example.....42

size of each classifier in the IPv6 policy

example.....42

576-bit size

active classifiers in the example

policy.....42

creation and attachment of an IPv6 policy,

example.....42

size of each classifier in the IPv6 policy

example.....42

factors to determine

available CAM resources.....25

for IPv6 classification

supported bit sizes.....41

maximum IPv6 policies supported

with four classifiers per policy.....27

with one classifier per policy.....27

performance impact

caused by CAM addressing.....25

