Configuration — QoS and IP Filtering Avaya Ethernet Routing Switch 8800/8600

7.1.3
NN46205-507, 07.01
January 2012

© 2012 Avaya Inc.

All Rights Reserved.

Notice

While reasonable efforts have been made to ensure that the information in this document is complete and accurate at the time of printing, Avaya assumes no liability for any errors. Avaya reserves the right to make changes and corrections to the information in this document without the obligation to notify any person or organization of such changes.

Documentation disclaimer

“Documentation” means information published by Avaya in varying mediums which may include product information, operating instructions and performance specifications that Avaya generally makes available to users of its products. Documentation does not include marketing materials. Avaya shall not be responsible for any modifications, additions, or deletions to the original published version of documentation unless such modifications, additions, or deletions were performed by Avaya. End User agrees to indemnify and hold harmless Avaya, Avaya's agents, servants and employees against all claims, lawsuits, demands and judgments arising out of, or in connection with, subsequent modifications, additions or deletions to this documentation, to the extent made by End User.

Link disclaimer

Avaya is not responsible for the contents or reliability of any linked Web sites referenced within this site or documentation provided by Avaya. Avaya is not responsible for the accuracy of any information, statement or content provided on these sites and does not necessarily endorse the products, services, or information described or offered within them. Avaya does not guarantee that these links will work all the time and has no control over the availability of the linked pages.

Warranty

Avaya provides a limited warranty on its Hardware and Software (“Product(s)”). Refer to your sales agreement to establish the terms of the limited warranty. In addition, Avaya’s standard warranty language, as well as information regarding support for this Product while under warranty is available to Avaya customers and other parties through the Avaya Support Web site: http://support.avaya.com. Please note that if you acquired the Product(s) from an authorized Avaya reseller outside of the United States and Canada, the warranty is provided to you by said Avaya reseller and not by Avaya.

Licenses

THE SOFTWARE LICENSE TERMS AVAILABLE ON THE AVAYA WEBSITE, HTTP://SUPPORT.AVAYA.COM/LICENSEINFO/ ARE APPLICABLE TO ANYONE WHO DOWNLOADS, USES AND/OR INSTALLS AVAYA SOFTWARE, PURCHASED FROM AVAYA INC., ANY AVAYA AFFILIATE, OR AN AUTHORIZED AVAYA RESELLER (AS APPLICABLE) UNDER A COMMERCIAL AGREEMENT WITH AVAYA OR AN AUTHORIZED AVAYA RESELLER. UNLESS OTHERWISE AGREED TO BY AVAYA IN WRITING, AVAYA DOES NOT EXTEND THIS LICENSE IF THE SOFTWARE WAS OBTAINED FROM ANYONE OTHER THAN AVAYA, AN AVAYA AFFILIATE OR AN AVAYA AUTHORIZED RESELLER; AVAYA RESERVES THE RIGHT TO TAKE LEGAL ACTION AGAINST YOU AND ANYONE ELSE USING OR SELLING THE SOFTWARE WITHOUT A LICENSE. BY INSTALLING, DOWNLOADING OR USING THE SOFTWARE, OR AUTHORIZING OTHERS TO DO SO, YOU, ON BEHALF OF YOURSELF AND THE ENTITY FOR WHOM YOU ARE INSTALLING, DOWNLOADING OR USING THE SOFTWARE (HEREINAFTER REFERRED TO INTERCHANGEABLY AS “YOU” AND “END USER”), AGREE TO THESE TERMS AND CONDITIONS AND CREATE A BINDING CONTRACT BETWEEN YOU AND AVAYA INC. OR THE APPLICABLE AVAYA AFFILIATE (“AVAYA”).

Copyright

Except where expressly stated otherwise, no use should be made of materials on this site, the Documentation, Software, or Hardware provided by Avaya. All content on this site, the documentation and the Product provided by Avaya including the selection, arrangement and design of the content is owned either by Avaya or its licensors and is protected by copyright and other intellectual property laws including the sui generis rights relating to the protection of databases. You may not modify, copy, reproduce, republish, upload, post, transmit or distribute in any way any content, in whole or in part, including any code and software unless expressly authorized by Avaya. Unauthorized reproduction, transmission, dissemination, storage, and or use without the express written consent of Avaya can be a criminal, as well as a civil offense under the applicable law.

Third-party components

Certain software programs or portions thereof included in the Product may contain software distributed under third party agreements (“Third Party Components”), which may contain terms that expand or limit rights to use certain portions of the Product (“Third Party Terms”). Information regarding distributed Linux OS source code (for those Products that have distributed the Linux OS source code), and identifying the copyright holders of the Third Party Components and the Third Party Terms that apply to them is available on the Avaya Support Web site: http://support.avaya.com/Copyright.

Preventing Toll Fraud

“Toll fraud” is the unauthorized use of your telecommunications system by an unauthorized party (for example, a person who is not a corporate employee, agent, subcontractor, or is not working on your company's behalf). Be aware that there can be a risk of Toll Fraud associated with your system and that, if Toll Fraud occurs, it can result in substantial additional charges for your telecommunications services.

Avaya Toll Fraud Intervention

If you suspect that you are being victimized by Toll Fraud and you need technical assistance or support, call Technical Service Center Toll Fraud Intervention Hotline at +1-800-643-2353 for the United States and Canada. For additional support telephone numbers, see the Avaya Support Web site: http://support.avaya.com. Suspected security vulnerabilities with Avaya products should be reported to Avaya by sending mail to: securityalerts@avaya.com.

Trademarks

The trademarks, logos and service marks (“Marks”) displayed in this site, the Documentation and Product(s) provided by Avaya are the registered or unregistered Marks of Avaya, its affiliates, or other third parties. Users are not permitted to use such Marks without prior written consent from Avaya or such third party which may own the Mark. Nothing contained in this site, the Documentation and Product(s) should be construed as granting, by implication, estoppel, or otherwise, any license or right in and to the Marks without the express written permission of Avaya or the applicable third party.

Avaya is a registered trademark of Avaya Inc.

All non-Avaya trademarks are the property of their respective owners, and “Linux” is a registered trademark of Linus Torvalds.

Downloading Documentation

For the most current versions of Documentation, see the Avaya Support Web site: http://support.avaya.com.

Contact Avaya Support

Avaya provides a telephone number for you to use to report problems or to ask questions about your Product. The support telephone number is 1-800-242-2121 in the United States. For additional support telephone numbers, see the Avaya Web site: http://support.avaya.com.

Contents

Chapter 1: Purpose of this document 9
Chapter 2: New in this release 11
Features 11
8812XL SFP+ I/O module 11
Other changes 11
Chapter 3: QoS fundamentals 13
Introduction to QoS 13
QoS for R modules 14
QoS for RS and 8800 modules 15
QoS and filters 16
DiffServ networks 16
Packet classification, marking, and mapping 17
PHB 18
DiffServ and the Ethernet Routing Switch 8800/8600 19
QoS implementation 20
DiffServ and non-IP traffic 21
DiffServ configuration parameters 21
Layer 2 and Layer 3 trusted and untrusted ports 23
DiffServ and ACLs 31
Queueing 32
Critical or Network ADSSC 36
Egress queue packet assignment 43
Policing and shaping 51
Token buckets and policing 52
Policy-based policer versus shaper 53
Policy-based traffic policing 54
Port-based traffic policing 59
Queue-based traffic shaping 60
Port-based shaping 61
Broadcast and multicast traffic bandwidth limiters 61
QoS and MPLS 61
QoS and VoIP 62
Automatic QoS 62
802.1Q tagged packets 64
Chapter 4: Traffic filtering fundamentals 65
Overview 65
Traffic filters for R, RS, and 8800 series modules 65
Deep packet pattern match filters 66
R, RS, and 8800 series module filters and packet layer traversal 66
Access control templates 66
ACT attributes 67
ACT patterns for offset filtering 67
Predefined ACTs 70
ACT configuration guidelines 72

Access control lists ... 72
ACL priority ... 74
Access control entries ... 75
ACE overview ... 75
ACE actions ... 76
ACE priority ... 77
Common ACE uses and configurations ... 78
Example: ACE TCP Established flag filter ... 79
Port mirroring, ACLs, and ACEs ... 80
R modules and port mirroring ... 81
RS and 8800 modules and port mirroring ... 81
Traffic filter configuration ... 81
ACL, ACT, and ACE configuration guidelines ... 82
Secure Network Access ... 82
Chapter 5: QoS and IP filter configuration ... 85
Chapter 6: Basic DiffServ configuration using Enterprise Device Manager ... 87
Enabling DiffServ on a port ... 87
Procedure steps ... 87
Configuring Layer 3 trusted or untrusted ports ... 87
Procedure steps ... 88
Configuring Layer 2 trusted or untrusted ports ... 88
Procedure steps ... 88
Configuring the port QoS level ... 88
Procedure steps ... 89
Configuring the VLAN QoS level ... 89
Chapter 7: QoS configuration using Enterprise Device Manager ... 91
Broadcast and multicast bandwidth limiting ... 91
Configuring port-based shaping ... 91
Configuring a policy-based policer ... 92
Configuring an egress queue set ... 93
Configuring egress queue set queues ... 94
Modifying an egress queue set or queue ... 96
Modifying ingress 802.1p to QoS mappings ... 97
Modifying ingress DSCP to QoS mappings ... 97
Modifying ingress MPLS to QoS mappings ... 98
Modifying egress QoS to 802.1p mappings ... 99
Modifying egress QoS to DSCP mappings ... 100
Modifying egress QoS to MPLS mappings ... 100
Chapter 8: Traffic filter configuration using Enterprise Device Manager ... 103
Traffic filter configuration procedures ... 103
Configuring ACTs ... 103
Adding a user-defined pattern ... 106
Configuring an access control list ... 107
Chapter 9: Access control entry configuration using Enterprise Device Manager ... 111
Configuring ACEs ... 111
Configuring ACE actions ... 114
Modifying ACE parameters ... 115

Configuring ACE ARP entries ... 115
Viewing all ACE ARP entries for an ACL ... 116
Configuring an ACE Ethernet source address ... 117
Configuring an ACE Ethernet destination address ... 118
Configuring an ACE LAN traffic type ... 119
Configuring an ACE Ethernet VLAN tag priority ... 121
Configuring an ACE Ethernet port ... 122
Configuring an ACE Ethernet VLAN ID ... 124
Viewing all ACE Ethernet entries for an ACL ... 125
Configuring an ACE IP source address ... 126
Configuring an ACE IP destination address ... 128
Configuring an ACE IP DSCP ... 129
Configuring an ACE IP protocol ... 130
Configuring ACE IP options ... 132
Configuring ACE IP fragmentation ... 133
Viewing all ACE IP entries for an ACL ... 134
Configuring an ACE TCP source port ... 135
Configuring an ACE UDP source port ... 137
Configuring an ACE TCP destination port ... 138
Configuring an ACE UDP destination port ... 139
Configuring an ACE ICMP message type ... 141
Configuring an ACE TCP flag ... 142
Viewing all ACE Protocol entries for an ACL ... 144
Configuring an ACE Pattern 1 entry ... 145
Configuring an ACE Pattern 2 entry ... 146
Configuring an ACE Pattern 3 entry ... 147
Viewing all ACE Advanced pattern entries for an ACL ... 148
Configuring an ACE IPv6 source address ... 149
Configuring an ACE IPv6 destination address ... 150
Configuring an ACE IPv6 next header ... 151
Viewing IPv6 attributes for an ACL ... 153
Chapter 10: Basic DiffServ configuration using the CLI ... 155
Job aid ... 155
Enabling DiffServ on a port ... 155
Configuring Layer 3 trusted or untrusted ports ... 156
Configuring Layer 2 trusted or untrusted ports ... 157
Configuring the port QoS level ... 158
Configuring the VLAN QoS level ... 158
Configuring the QoS level for a MAC address ... 159
Example of configuring a QoS level for a MAC address ... 160
Chapter 11: QoS configuration using the CLI ... 161
Job aid ... 161
Configuring broadcast and multicast bandwidth limiting ... 163
Configuring the port-based shaper ... 164
Configuring a port-based policer for RS and 8800 modules ... 165
Configuring a policy-based policer ... 165
Job aid ... 167

Adding lanes to a policy-based policer ... 167
Configuring an egress queue set ... 168
Example of configuring an egress queue set ... 170
Job aid ... 171
Modifying an egress queue set ... 171
Configuring an egress queue set queue ... 173
Example of configuring an egress queue set queue ... 175
Job aid ... 176
Configuring ingress mappings ... 176
Configuring egress mappings ... 178
Configuring Avaya Automatic QoS ... 179
Chapter 12: Traffic filter configuration using the CLI ... 181
Traffic filter configuration using the CLI procedures ... 181
Job aid ... 182
Configuring an ACT ... 185
Adding a user-defined pattern ... 187
Configuring an ACL ... 189
Configuring global and default actions for an ACL ... 190
Associating VLANs with an ACL ... 191
Associating ports with an ACL ... 192
Viewing filter configuration information ... 193
Job aid ... 194
Chapter 13: Access control entry configuration using the CLI ... 195
Job aid ... 195
Configuring ACEs ... 198
Configuring ACE actions ... 200
Configuring ACE debug actions ... 202
Example of configuring R module TxFilter mode mirroring ... 204
Configuring ARP ACEs ... 205
Configuring an Ethernet ACE ... 206
Example of configuring an Ethernet ACE ... 208
Configuring an IP ACE ... 208
Example of configuring an IP ACE ... 209
Configuring a protocol ACE ... 210
Example of configuring a protocol ACE ... 211
Configuring a custom ACE ... 212
Example of configuring a custom ACE ... 213
Configuring an IPv6 ACE ... 213
Viewing ACL and ACE configuration data ... 215
Chapter 14: CLI configuration examples ... 217
Delivering subrate IP service using policy-based policers ... 217
Policing multiple flows using VLAN-based ACLs ... 219
Mirroring using ACLs ... 223
Asymmetric downlink and uplink using policy-based policers and port-based shapers ... 225
Chapter 15: Basic DiffServ configuration using the ACLI ... 227
Job aid ... 227
Enabling DiffServ on a port ... 228

Configuring Layer 3 trusted or untrusted ports ... 228
Configuring Layer 2 trusted or untrusted ports ... 229
Configuring the port QoS level ... 230
Configuring the VLAN QoS level ... 231
Configuring the QoS level for a MAC address ... 232
Example of setting a QoS level for a MAC address ... 233
Chapter 16: QoS configuration using the ACLI ... 235
Job aid ... 235
Configuring broadcast and multicast bandwidth limiting ... 237
Configuring the port-based shaper ... 239
Configuring a port-based policer for RS and 8800 modules ... 240
Configuring a policy-based policer ... 240
Job aid ... 241
Configuring an egress queue set ... 242
Job aid ... 244
Configuring an egress queue set queue ... 244
Modifying an egress queue set or egress queue set queue ... 246
Configuring ingress mappings ... 248
Configuring egress mappings ... 249
Configuring Avaya Automatic QoS ... 250
Chapter 17: Traffic filter configuration using the ACLI ... 253
Traffic filter configuration procedures ... 253
Job aid ... 254
Configuring an ACT ... 256
Adding a user-defined pattern ... 258
Configuring an ACL ... 259
Configuring global and default actions for an ACL ... 260
Associating VLANs with an ACL ... 262
Associating ports with an ACL ... 262
Viewing filter configuration information ... 263
Job aid ... 264
Chapter 18: Access control entry configuration using the ACLI ... 267
Job aid ... 267
Configuring ACEs ... 269
Configuring ACE actions ... 271
Example of configuring ACE actions ... 273
Configuring ACE debug actions ... 273
Configuring ARP ACEs ... 275
Configuring an Ethernet ACE ... 276
Example of configuring an Ethernet ACE ... 277
Configuring an IP ACE ... 278
Example of configuring an IP ACE ... 279
Configuring a protocol ACE ... 279
Example of configuring a protocol ACE ... 281
Configuring a custom ACE ... 281
Example of configuring a custom ACE ... 283
Configuring an IPv6 ACE ... 283

Example of configuring an IPv6 ACE 284
Viewing ACL and ACE configuration data 284
Chapter 19: Safety messages 287
Notices 287
Attention notice 287
Caution ESD notice 287
Caution notice 288
Chapter 20: Customer Service 291
Getting technical documentation 291
Getting product training 291
Getting help from a distributor or reseller 291
Getting technical support from the Avaya Web site 291
Appendix A: Advanced filter examples 293
ACE filters for secure networks 293
Appendix B: Egress queues and pages 349
Appendix C: Workaround for inVlan, srcIp ACL 351
Procedure steps 351
Glossary 353

Chapter 1: Purpose of this document

This document helps you to configure Quality of Service (QoS) and filtering operations on the Avaya Ethernet Routing Switch 8800/8600 using the Command Line Interface (CLI), the Avaya Command Line Interface (ACLI), and the Enterprise Device Manager (EDM).

Comments? infodev@avaya.com

Chapter 2: New in this release

The following sections detail what's new in *Avaya Ethernet Routing Switch 8800/8600 Configuration — QoS and IP Filtering,* (NN46205-507) for Release 7.1.3.

- Features on page 11
- Other changes on page 11

Features

See the following section for information about changes that are feature-related.

8812XL SFP+ I/O module

Release 7.1.3 introduces a new Ethernet Routing Switch 8800 interface module — the 8812XL SFP+ I/O module. This module supports 12 SFP+ ports at 10Gbps and provides the same functionality as its RS module equivalent, the 8612XLRS.

All 8800 series modules including the 8812XL SFP+ I/O module use the new enhanced network processor, the RSP 2.7.

For information on the supported R, RS and 8800 modules in this release, and their installation, see *Avaya Ethernet Routing Switch 8800/8600 Installation — Modules*, (NN46205–304).

For information on SFP+ transceivers, see *Avaya Ethernet Routing Switch 8800/8600 Installation — SFP, SFP+, XFP, and OADM Hardware Components*, (NN46205–320).

Other changes

There are no other changes to this document for release 7.1.3.

Chapter 3: QoS fundamentals

Use the information in this chapter to help you understand Quality of Service (QoS).

This chapter describes a range of features that you can use with the Avaya Ethernet Routing Switch 8800/8600 to allocate network resources to critical applications. You can configure your network to prioritize specific types of traffic to ensure traffic receives the appropriate QoS level. Allocate priority to protocol and application data depending on required parameters, for example, minimum data rate or minimum time delay.

For information about how to use the command line interface (CLI), the Avaya Command Line Interface (ACLI), and Enterprise Device Manager (EDM), see *Avaya Ethernet Routing Switch 8800/8600 Fundamentals — User Interfaces,* (NN46205-308).

Introduction to QoS

QoS is the extent to which a service delivery meets user expectations. In a QoS-aware network, a user can expect the network to meet certain performance levels. You specify these performance levels in terms of service availability, packet loss, packet delay, and packet delay variation.

By assigning QoS levels to traffic flows on your Local Area Network (LAN), you can allocate network resources where you need them most. For an effective QoS strategy, you must configure QoS functionality from end-to-end in the network: across various devices, such as routers, switches, and end stations; across platforms and media; and across link layers, such as an Ethernet.

The Ethernet Routing Switch 8800/8600 supports QoS classification for both L2 (802.1p bits) and L3 (Differentiated Services Code Point bits) parameters. Do not confuse the terminology L2 and L3 with Layer 2 (bridging) or Layer 3 (routed) operation. L2 represents an association with Q-tags, of which 802.1p bits is a portion. L3 represents an association with Differentiated Services Code Point (DSCP).

The Ethernet Routing Switch 8800/8600 provides QoS functionality that can differ for Layer 2 (bridged) and Layer 3 (routed) traffic flows. The Ethernet Routing Switch 8800/8600 can also assign QoS levels based on multiple criteria including (but not limited to) Transport Control Protocol (TCP) or User Datagram Protocol (UDP) ports used by an application.

To effectively use QoS functions in your network, you must perform the following tasks:

- Identify traffic sources and types.
- Determine the required QoS parameters based on the traffic.
- Perform traffic management (QoS) operations based on the required parameters.

Important:

The QoS value of unicast packets is retained when forwarded to the CP as exception packets. If enough packets with high QoS setting are received, this could negatively affect CP handling of other packets. In general, unicast packets being sent to CP is abnormal, and the root cause of that situation should be investigated and resolved as a first step.

The Ethernet Routing Switch 8800/8600 implements the QoS functionality for IP traffic through a Differentiated Services (DiffServ) network architecture.

QoS for R modules

This release contains two QoS implementations:

- From Release 4.0, an implementation that uses specific R module features and includes support for the 8630GBR, 8648GTR, 8683XLR, and 8683XZR modules.
- From Release 5.0, an implementation for RS modules that performs all features of R modules, and offers advanced policing capabilities. See QoS for RS and 8800 modules on page 15 and Port-based traffic policing on page 59.

The following table shows the level of support for Advanced QoS implementations.

In this table, E denotes enabled, D denotes disabled, NA denotes not applicable, and ADV denotes advanced. The mode 256 K denotes the number of records in kilobytes supported for each mode.

Table 1: Features supported for each operation mode for R series modules

Module type	Features supported on modules			
R	QoS	Filters	Policing	Shaping
E	ADV	ADV	ADV	ADV

An all-R module chassis configuration includes the following capabilities:

- Feedback Output Queueing (FOQ)
- high scaling; for more information, see the most recent Ethernet Routing Switch 8800/8600 release notes

You can configure up to 128 MultiLink Trunking (MLT) groups, and up to 8 Equal Cost Multipath (ECMP) routing paths.

Enhanced Operational mode increases virtual local area network (VLAN) MLT scalability. Use Enhanced Operational mode to provide up to 1980 MLT VLANs. For more information about Enhanced Operational mode, VLANs, and VLAN scalability, see *Avaya Ethernet Routing Switch 8800/8600 Configuration — VLANs and Spanning Tree,* (NN46205-517).

R series modules support both ingress and egress filtering by using ACLs.

R modules use many features, such as FOQ, shaping, and policing, to implement QoS functionality.

QoS for RS and 8800 modules

RS and 8800 module ports operate at up to 10 Gb/s. At high data rates, ensuring network stability is critical. The switch cannot drop network control protocol traffic. In addition, the switch must process high-priority traffic, such as VoIP traffic, even at the expense of lower-priority data traffic. To provide such performance, the RS or 8800 module performs frame classification and scheduling at the MAC layer (Layer 2).

You can oversubscribe RS and 8800 modules on ingress. The Ethernet Media Access Controller data transport device operates such that the switch continues to forward protocol and other high-priority traffic during congestion. Each RS and 8800 module port uses three ingress queues to handle priority traffic if ingress oversubscription occurs.

RS and 8800 modules support the same QoS features as R modules, and provide QoS functionality at the MAC layer by using port-based policers. For more information, see Port-based traffic policing on page 59. R, RS, and 8800 modules use Advanced (ACL-based) filters.

RS and 8800 modules use three strict-priority queues for each port. These queues are ingress queues on the Ethernet Media Access Controller data transport device.

RS modules include the 8648GTRS, the 8612XLRS, the 8634XGRS, and the 8648GBRS. 8800 modules include the 8848GT, the 8812XL, the 8834XG, and the 8848GB. The 8648GBRS, 8848GB, 8648GTRS, 8848GT, and 10/100/1000 Mb/s ports of the 8634XGRS and the 8834XG support eight queues for each egress port. The 8612XLRS, the 8812XL, and the 10 Gb/s Ethernet ports of the 8634XGRS and the 8834XG support up to 64 queues for each egress port.

QoS and filters

The Ethernet Routing Switch 8800/8600 has functions you can use to provide appropriate QoS levels to traffic for each customer, application, or packet. These functions include egress-queue-set-based shapers, port-based shapers, DiffServ access or core port settings, policy-based policers, and port-based policers. The Ethernet Routing Switch 8800/8600 also provides advanced ACL filters. You need not use filters to provide QoS; however, filters help prioritize customer traffic. Filters also provide protection by blocking unwanted traffic.

Policers apply at ingress; ACL-based filters and shapers apply at egress.

DiffServ networks

DiffServ divides traffic into various classes (behavior aggregates) to give each class differentiated treatment.

A DiffServ network provides either end-to-end or intradomain QoS functionality by implementing classification and mapping functions at the network boundary or access points. Within a core network, DiffServ regulates packet behavior by this classification and mapping.

DiffServ, as defined by RFC 2475, provides QoS for aggregate traffic flows (as opposed to individual traffic flows, which use an Integrated Services architecture [IntServ—RFC 1633]). DiffServ provides QoS by using traffic management and conditioning functions (packet classification, marking, policing, and shaping) on network edge devices, and by using Per-Hop Behaviors (PHB), which includes queueing and dropping traffic on network core devices. The Ethernet Routing Switch can perform all these QoS functions. The order of DiffServ operations for a packet is as follows:

- packet classification: IEEE 802.1p, EXP-bit, and DSCP markings classify (map) the packet to the appropriate PHB and QoS level.

  For more information, see Packet classification, marking, and mapping on page 17.

- policing: The switch rate-limits and colors packets; the switch drops or re-marks excessive traffic.

  For more information, see Policy-based traffic policing on page 54and Port-based traffic policing on page 59.

- re-marking: The switch can re-mark packets according to QoS actions you configure into the switch (internal QoS mappings).

  For more information, see Internal QoS level on page 48.

- shaping: The Ethernet Routing Switch 8800/8600 provides both queue-based and port-based shaping. Egress queue shaping provides shaping for each queue; port-based shaping shapes all outgoing traffic to a specific rate.

For more information, see Queue-based traffic shaping on page 60 and Port-based shaping on page 61.

Although you do not require filters for QoS operation, you can use filters to provide traffic management actions.

For more information about Advanced filters, see Traffic filtering fundamentals on page 65.

Packet classification, marking, and mapping

Traffic classification includes functions that examine a packet to determine further actions according to defined rules. Classification involves identifying flows so that the router can modify the packet contents or PHB, apply conditioning treatments to the packet, and determine how to forward the packet to the egress interface. Packet classification depends on the service type of the packet and the point in the traffic management process where the classification occurs.

The device classifies traffic as it enters the DiffServ network, and assigns the appropriate PHB based on the classification. To differentiate between classes of service, the device marks the DiffServ (DS) parameter in the IP packet header, as defined in RFC 2474 and RFC 2475. The DSCP marking defines the forwarding treatment of the packet at each network hop. This marking (or classification) occurs at the edge of the DiffServ domain, and is based on the policy (or filter) associated with a microflow or aggregate flow.

You can configure the mapping of DSCP-to-forwarding behaviors and DSCP re-markings. Re-marking the DSCP resets the treatment of packets based on new network specifications or desired levels of service.

Layer 3 marking uses the DSCP parameter. Layer 2 (Ethernet) marking uses the 802.1p-bit parameter.

For Layer 2 packets, priority bits (or 802.1p bits) define the traffic priority of the Ethernet packet. You can configure an interface to map DSCP, 802.1p, or EXP bits to internal QoS levels on ingress. You can configure an interface to map internal QoS levels to DSCP, 802.1p, or EXP bits at egress. 802.1p bit mapping, which assesses the 802.1p bit and derives an appropriate DSCP, meets the Ethernet VLAN QoS requirements.

Within the network, a packet PHB associated with the DSCP determines how a device forwards the packet to the next hop—if at all. Consequently, nodes can allocate buffer and bandwidth resources to each competing traffic stream. The initial DSCP setting is based on network policies for the type of service required. The objective of DSCP-to-NSC mapping is to translate the QoS characteristics defined by the packet DSCP marker to an Networks Service Class (NSC). The DSCP-to-NSC mapping occurs at ingress. For each received packet, the mapping function assigns an NSC.

The Ethernet Routing Switch maintains six mapping tables. These tables translate the ingress 802.1p-bit, EXP-bit, or DSCP markings to an internal QoS level, and then retranslate the internal QoS level to an egress DSCP, EXP-bit, or 802.1p-bit markings as follows:

- Ingress 802.1p-bit to QoS level
- Ingress DSCP to QoS level
- Ingress MultiProtocol Label Switching (MPLS) EXP-bit to QoS level
- QoS level to egress 802.1p-bit
- QoS level to egress DSCP
- QoS level to egress MPLS EXP-bit

For more information about mappings, see Egress queue packet assignment on page 43.

PHB

When traffic enters the DiffServ network, packets enter a queue according to the marking, which determines the PHB of the packets. For example, if the system marks a video stream to receive the highest priority, it enters a high-priority queue. As these packets traverse the DiffServ network, the system forwards the video stream before other packets.

RFC 2597 and RFC 2598 define two standard PHBs: the Assured Forwarding PHB group and the Expedited Forwarding PHB group. The Avaya Ethernet Routing Switch 8800/8600 also uses the Default (DF) and Class Selector (CS) groups. Class Selector in a DiffServ network provides backward compatibility with IP precedence.

Assured Forwarding PHB group

RFC 2597 describes the Assured Forwarding PHB group, which divides delivery of IP packets into four independent classes. The Assured Forwarding PHB group offers different levels of forwarding resources in each DiffServ node. Within each Assured Forwarding PHB group, the system marks IP packets with one of three possible drop precedence values. During network congestion, the drop precedence of a packet determines the relative importance within the Assured Forwarding PHB group.

Expedited Forwarding PHB group

RFC 2598 describes the Expedited Forwarding PHB group as the Premium service: the best service the network can offer. Expedited Forwarding PHB is a forwarding treatment for a DiffServ microflow when the transmission rate ensures that it is the highest priority and it experiences no packet loss for in-profile traffic.

DiffServ and the Ethernet Routing Switch 8800/8600

The Avaya Ethernet Routing Switch 8800/8600 implements a DiffServ architecture as defined in RFC 2474 and RFC 2475. The IEEE 802.1p and the DSCP markings in virtual local area networks (VLAN) classify the packet to the appropriate PHB and QoS level to provide Layer 2 and Layer 3 QoS functionality, respectively.

You can use Ethernet Routing Switch 8800/8600s in the network core. The switches can perform classification, marking, policing, or shaping; they perform the actions defined by the PHB of the packet. To determine whether a port is an edge (access) or a core device, configure each port as access or core. The default is core.

The following figure illustrates DiffServ network operations. Ethernet Routing Switch 8800/8600s exist on the network edge where they perform classification, marking, policing, and shaping functions.

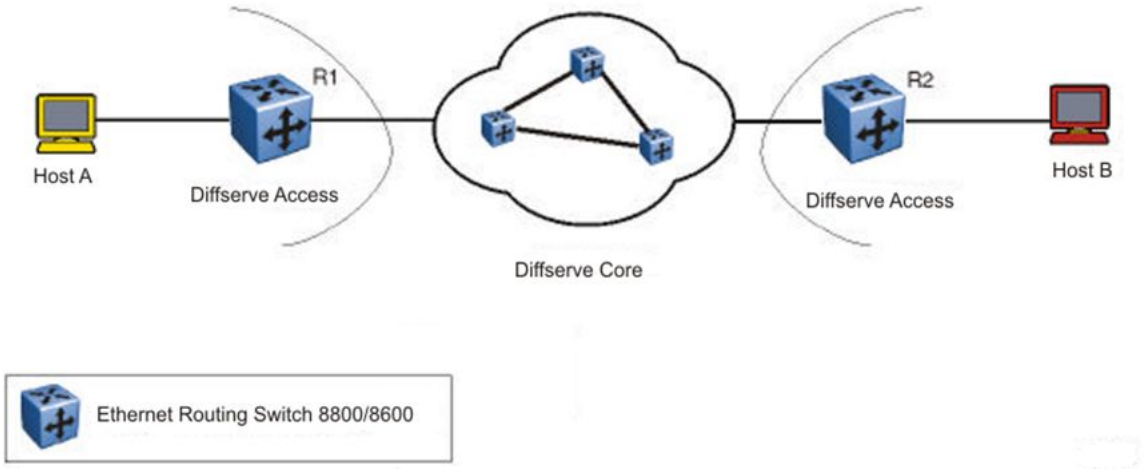


Figure 1: DiffServ network core and edge devices

When you configure a port as a core port, packet markings are trusted. When you configure a port as an access port, packet markings are not trusted.

DiffServ access port (untrusted)

Use a DiffServ access port, as shown in Figure 1: DiffServ network core and edge devices on page 19, at the edge of a DS network. The access port classifies traffic by re-marking the L3 DSCP parameter to zero (it does not trust the traffic markings) or by ignoring the 802.1p bits within a Dot1Q-tagged packet. The system adds Dot1Q headers at ingress, and adds them back at egress only when you configure the egress port as a tagged or trunk port.

DiffServ core port (trusted)

A DiffServ core port does not change packet classification or markings; the port trusts the incoming traffic markings. A core port preserves the DSCP marking of all incoming packets, and uses these markings to assign the packet to an internal QoS level. For tagged packets, the port honors the 802.1p bits within a Dot1Q header, and uses these bits to classify ingress traffic. Use the 802.1p override command to honor (or not) 802.1p bits.

QoS operations for IPv4 and IPv6 are the same. You can associate all traffic with MAC, port, and VLAN QoS levels rather than with 802.1p bits or the DSCP parameter.

QoS implementation

The following figure shows how the Avaya Ethernet Routing Switch 8800/8600 provides QoS functionality. The order of operations is as follows:

- ingress classification of the packet
- mapping of ingress classification to an internal QoS value
- placement of the packet into an egress queue based on the internal QoS-to-egress queue mapping
- egress servicing of the packet by a scheduler

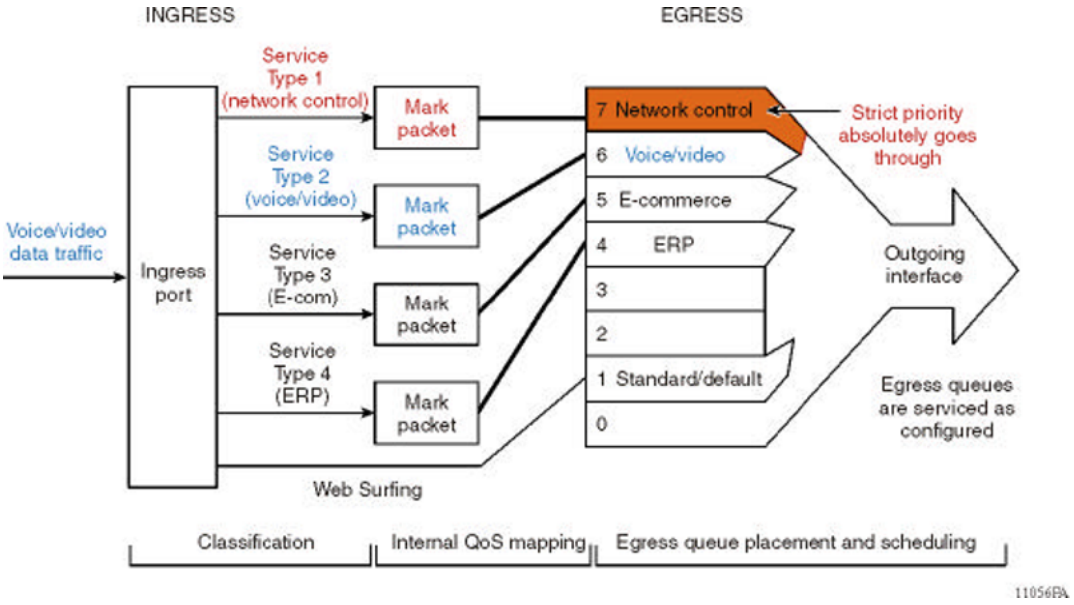


Figure 2: Overview of Avaya Ethernet Routing Switch 8800/8600 QoS operations

Ingress QoS configuration parameters determine traffic classification. Classification creates a mapping to an internal QoS level (0 to 7) that maps to an egress queue. The egress queue

mapping determines the output packet DSCP, EXP-bit, or 802.1p markings. Whether a packet is part of a Layer 2 (bridged) or a Layer 3 (routed) traffic flow can affect QoS operations.

At ingress, you can modify traffic classification with filters (Access Control Lists—ACL); however, QoS deployment does not require the use of traffic filters. You can use traffic filters to configure criteria to identify a microflow or an aggregate flow. The filters can match multiple parameters in the IP packet and can assign actions that match the criteria you specify. Filters override the standard ingress QoS or DiffServ operations.

Implement a DiffServ network on the Avaya Ethernet Routing Switch 8800/8600 by configuring a port as trusted or untrusted.

DiffServ and non-IP traffic

DiffServ applies only to IP packets. The system maps non-IP traffic to a source MAC, port, or VLAN QoS level. For R, RS, and 8800 module ports, the system first maps traffic to the MAC QoS level. With no MAC QoS level setting or match, the Avaya Ethernet Routing Switch 8800/8600 chooses between port and VLAN QoS levels by selecting the highest QoS level setting. Normal egress QoS operation then occurs, although egress mapping tables associated with DSCP do not apply—DSCP is an IP-only parameter.

DiffServ configuration parameters

You can use a number of parameters to configure DiffServ and QoS. All packets receive QoS operation handling. The following sections describe these parameters using Enterprise Device Manager terms.

In the following sections, do not confuse the terminology L2 and L3 with Layer 2 (bridging) or Layer 3 (routed) operation. L2 represents an association with Q-tags, of which 802.1p bits is a portion. L3 represents an association with DSCP.

- DiffServ—true or false on page 21
- Layer3Trust—core or access on page 22
- Layer2 8021p Override on page 22
- Port-based QoS level on page 22
- VLAN-based QoS level on page 23

DiffServ—true or false

You can configure the DiffServ parameter to true or false; false is the default. This parameter works with the Layer3Trust parameter. The DiffServ parameter is a global parameter that affects QoS L3 DSCP operations.

If the DiffServ parameter is false (DiffServ disabled), the L3 DSCP parameter is not used for classification or modified. When the DiffServ parameter is true, it activates the Layer3Trust parameter.

Layer3Trust—core or access

You can configure the Layer3Trust parameter to core or access; core is the default. Core configures the port to a trusted state and access configures the port to an untrusted state

The DiffServ parameter determines the operation of this parameter. The operation depends on whether the port is tagged or untagged. Tagged packet operation depends on the Layer2 8021p Override parameter (described next). If DiffServ is false, Layer3Trust has no effect; no modification of the DSCP or TOS bits occurs. If DiffServ is true, the core and access settings take affect as described in DiffServ access port (untrusted) on page 19 and DiffServ core port (trusted) on page 20.

Layer2 8021p Override

You can configure the Layer2 8021p Override parameter to true or false; false is the default.

This parameter primarily affects L2 tagged packet treatment, but can also affect the treatment of the L3 DSCP parameter.

If Layer2 8021p Override is false, the port trusts the 802.1p-bit portion of a Q-tagged packet. The port trusts the 802.1p-bit marking regardless of the port setting (tagged or untagged); however, if the discard tagged packets parameter (DiscardTaggedFrames) on an untagged port is true, the port discards the packet.

If Layer2 8021p Override is true, the port does not trust the 802.1p bit marking. No re-marking occurs because the system strips 802.1p bits at ingress. In this case, the QoS operation depends on other parameters, such as DiffServ and Layer3Trust settings, or the MAC, port, or VLAN QoS level.

Port-based QoS level

Use the port-based QoS level to configure the default QoS level for a port. You can configure the QoS level from 0 to 6 (level 7 is reserved for internal switch use—network control traffic). The default value is 1.

For VoIP traffic, Avaya recommends that you use QoS level 6.

If you configure port QoS levels, Layer 2 and Layer 3 traffic from the same port has the same QoS level.

VLAN-based QoS level

Use the VLAN-based QoS level to configure a default QoS level for a VLAN. You can configure a QoS level from 0 to 6 (level 7 is reserved for internal switch use— network control traffic). The default value is 1.

Use VLAN-based QoS levels to customize VLANs for traffic applications. For example, add a Voice VLAN to an edge switch to carry VoIP traffic. Then you can apply a QoS level to the Voice VLAN to ensure proper handling of time-sensitive VoIP traffic without using filters. For VoIP traffic, Avaya recommends that use you QoS level 6.

Layer 2 and Layer 3 trusted and untrusted ports

This section contains a series of traffic processing flowcharts. The flowcharts show QoS operations that result from various configuration options. You can configure ports as trusted or untrusted at both Layer 2 (802.1p) or Layer 3 (DSCP) for ingress packet classification. The following section describes the configuration combinations:

- Layer 2 untrusted and Layer 3 untrusted on page 24
- Layer 2 untrusted and Layer 3 trusted on page 25
- Layer 2 trusted and Layer 3 trusted on page 27
- Layer 2 trusted and Layer 3 untrusted on page 28

The Avaya Ethernet Routing Switch 8800/8600 provides eight internal QoS levels. These eight levels, numbered zero to seven, map to the egress queues (see Ingress mappings and queues on page 44) through

- the MAC, port, or VLAN QoS level settings (also numbered zero to seven)
- the ingress 8021p to (internal) QoS mapping table
- the ingress DSCP to (internal) QoS mapping table
- the ingress MPLS EXP bit to (internal) QoS mapping table

If the default number of egress queues changes by using a custom queue set, you can alter the mapping tables as required.

The default number of queues for either the 8 max-queue-set or the 64 max-queue-set is 8.

The following sections and flowcharts include no MPLS QoS operations. For information about MPLS actions, see QoS and MPLS on page 61.

Layer 2 untrusted and Layer 3 untrusted

To configure a port as Layer 2 untrusted and Layer 3 untrusted, assign the following parameter values:

- DiffServ = true
- Layer3Trust = access
- Layer2 8021p Override = true

Use this configuration to classify packets through either MAC, port, or VLAN QoS levels. Use VLAN QoS for a VLAN that carries traffic for a single application. For example, directly connected voice traffic can use VLAN QoS to give the same ingress classification to all packets (all ingress packets are voice packets). You can use MAC-based QoS for all packets from a single device. You can use a port-based QoS level for all packets that enter a port within a VLAN, rather than a VLAN-based QoS level, which applies to all ports within the VLAN.

For details about Layer 2 untrusted, Layer 3 untrusted QoS operations, see Figure 3: DiffServ access mode with 802.1p override enabled on page 25.

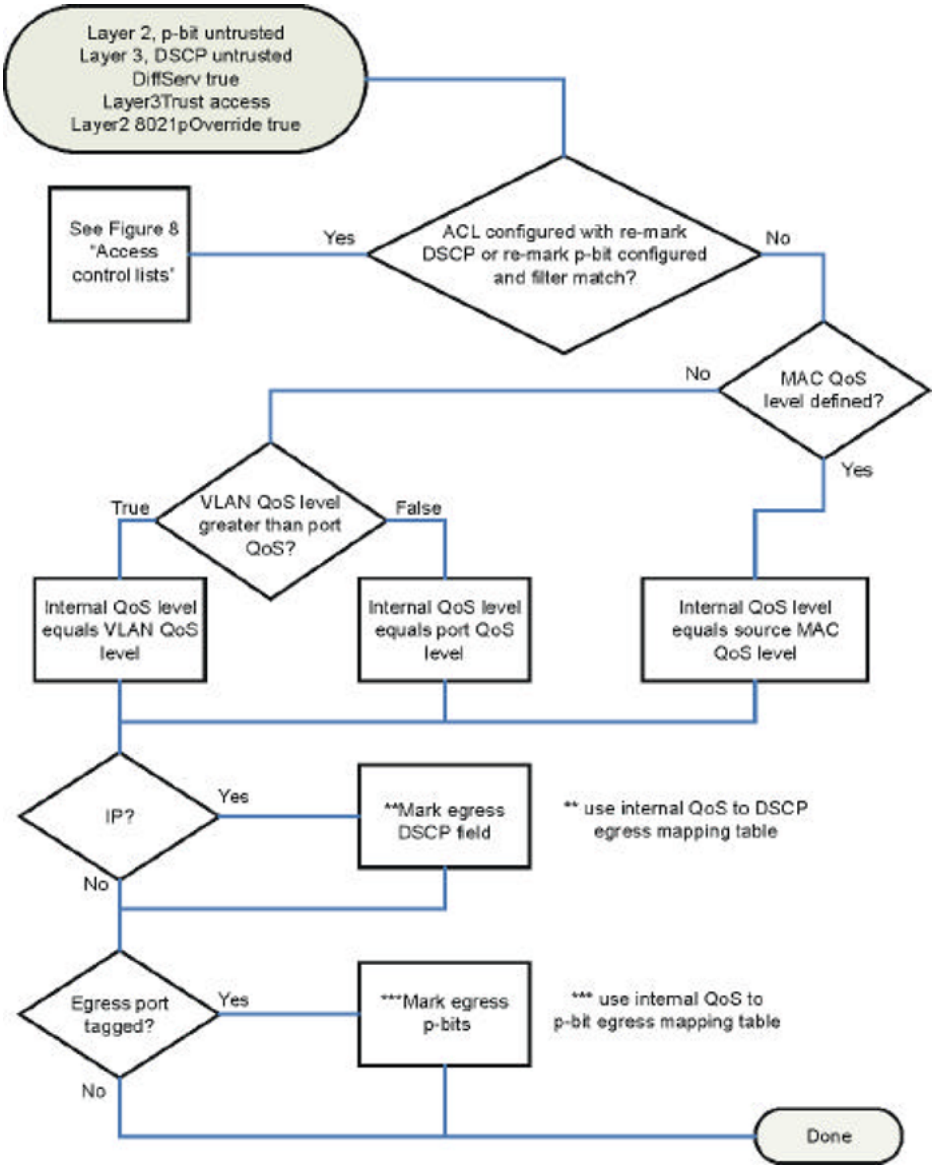


Figure 3: DiffServ access mode with 802.1p override enabled

Layer 2 untrusted and Layer 3 trusted

To configure a port as Layer 2 untrusted and Layer 3 trusted, assign the following parameter values:

- DiffServ = true
- Layer3Trust = core
- Layer2 8021p Override = true

Use these configuration options to classify packet QoS through the DSCP parameter for all IP packets, whether tagged or untagged. This configuration is typical when another QoS or

DiffServ-enabled and configured switch marks IP packets at the edge. These already marked packets arrive L3 trusted, and the Avaya Ethernet Routing Switch 8800/8600 continues with the trust (DiffServ core port operation). For tagged packets, 802.1p bits are not examined. For non-IP packets, this configuration causes classification by one of MAC, port, or VLAN QoS settings.

For details about Layer 2 untrusted, Layer 3 trusted QoS operations, see Figure 4: DiffServ core mode with 802.1p override enabled on page 26.

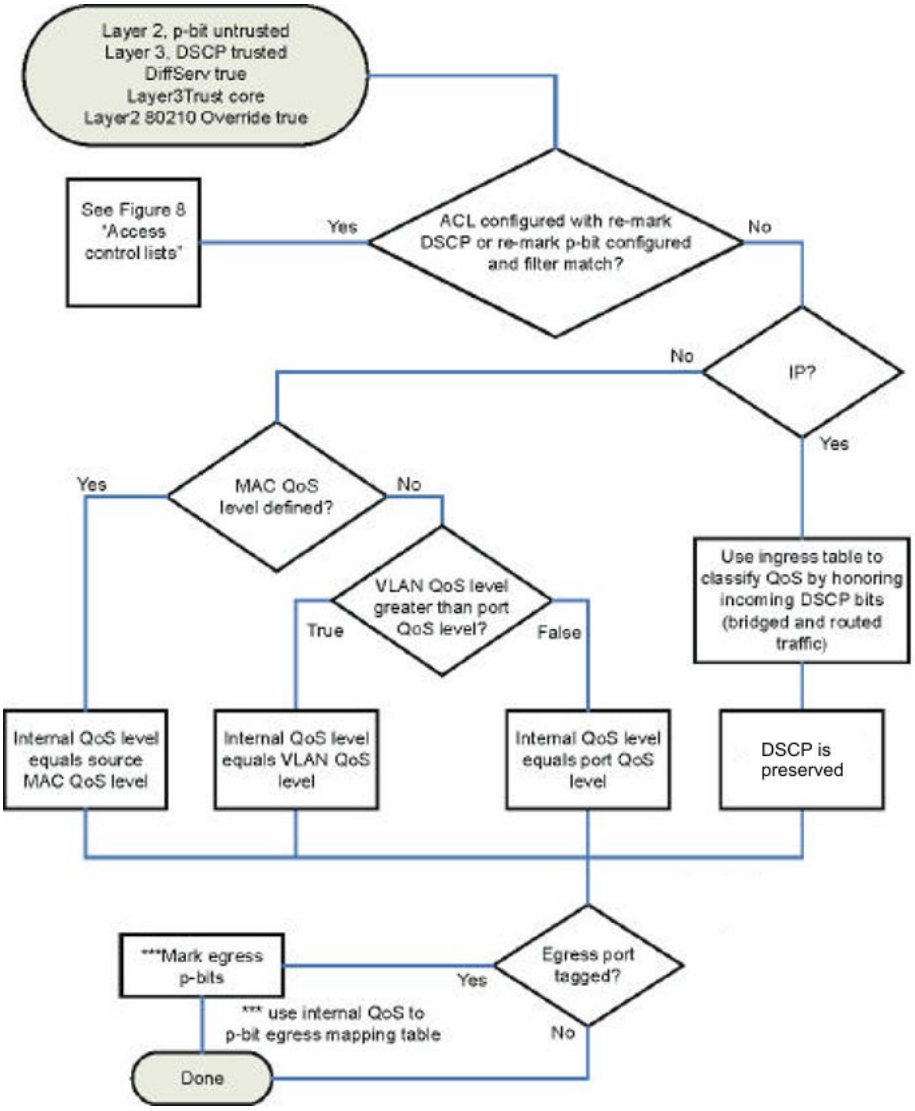


Figure 4: DiffServ core mode with 802.1p override enabled

Comments? infodev@avaya.com

Layer 2 trusted and Layer 3 trusted

To configure a port as Layer 2 trusted and Layer 3 trusted, assign the following parameter values:

- DiffServ = true
- Layer3Trust = core
- Layer2 8021p Override = false

Use these configuration options to classify packet QoS through 802.1p for all IP tagged packets, and through DSCP for all untagged routed IP packets. If the packet is non-IP or bridged IP, the system uses the MAC, port, or VLAN QoS level. This action is independent of tagged (trunk) or untagged (access) port settings. An exception is an untagged port with a DiscardTaggedFrames parameter of true (nondefault); the port discards the packet rather than classifies it for QoS treatment.

For details about Layer 2 trusted, Layer 3 trusted QoS operations, see Figure 5: DiffServ core mode with 802.1p override disabled on page 28.

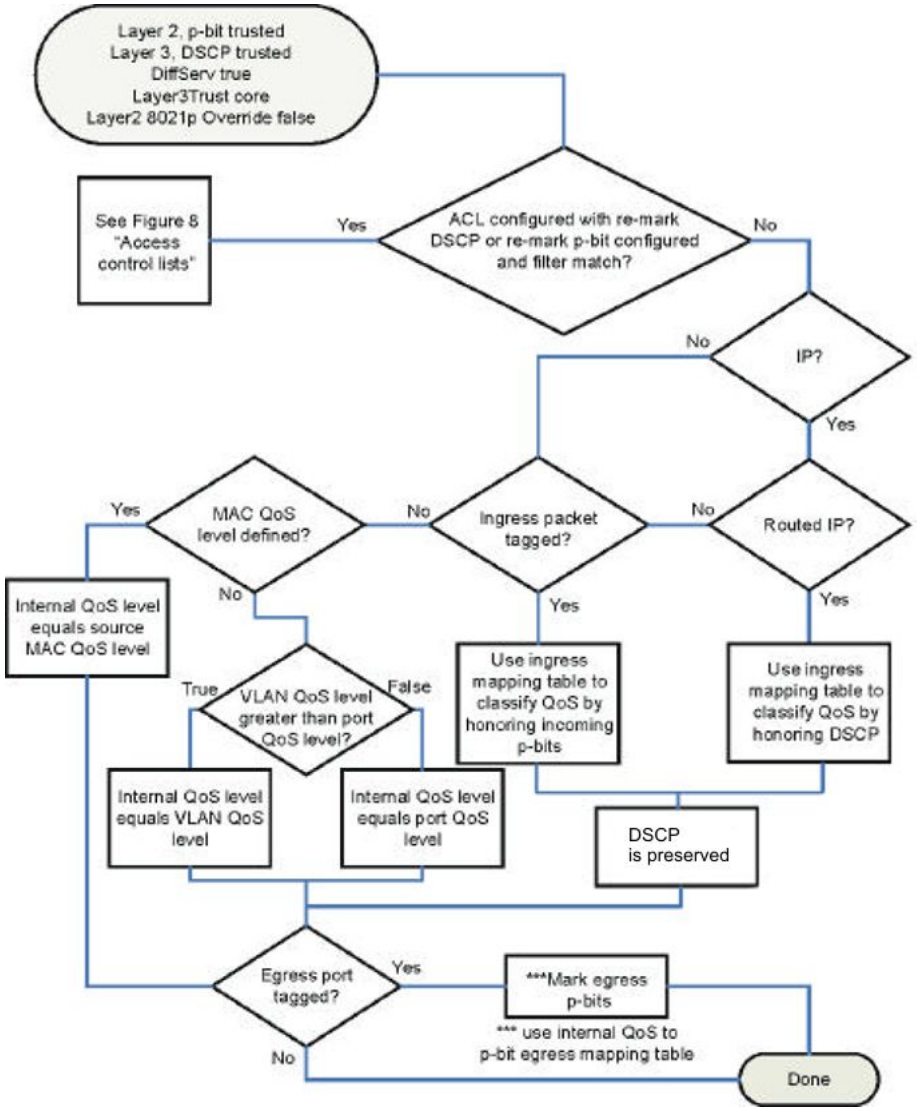


Figure 5: DiffServ core mode with 802.1p override disabled

Layer 2 trusted and Layer 3 untrusted

To configure a port as Layer 2 trusted and Layer 3 untrusted, assign the following parameter values:

- DiffServ = True
- Layer3Trust = Access
- Layer2 8021p Override = false

Use these configuration options to classify packet QoS through 802.1p for all tagged packets, and MAC, port, or VLAN QoS levels for all untagged packets. One MAC, port, or VLAN QoS

level setting handles all untagged (IP or non-IP) packets. If the packet is an IP packet, the DSCP parameter bits are not modified or examined.

For details about Layer 2 trusted, Layer 3 untrusted QoS operations, see Figure 6: DiffServ access mode with 802.1p override disabled on page 29.

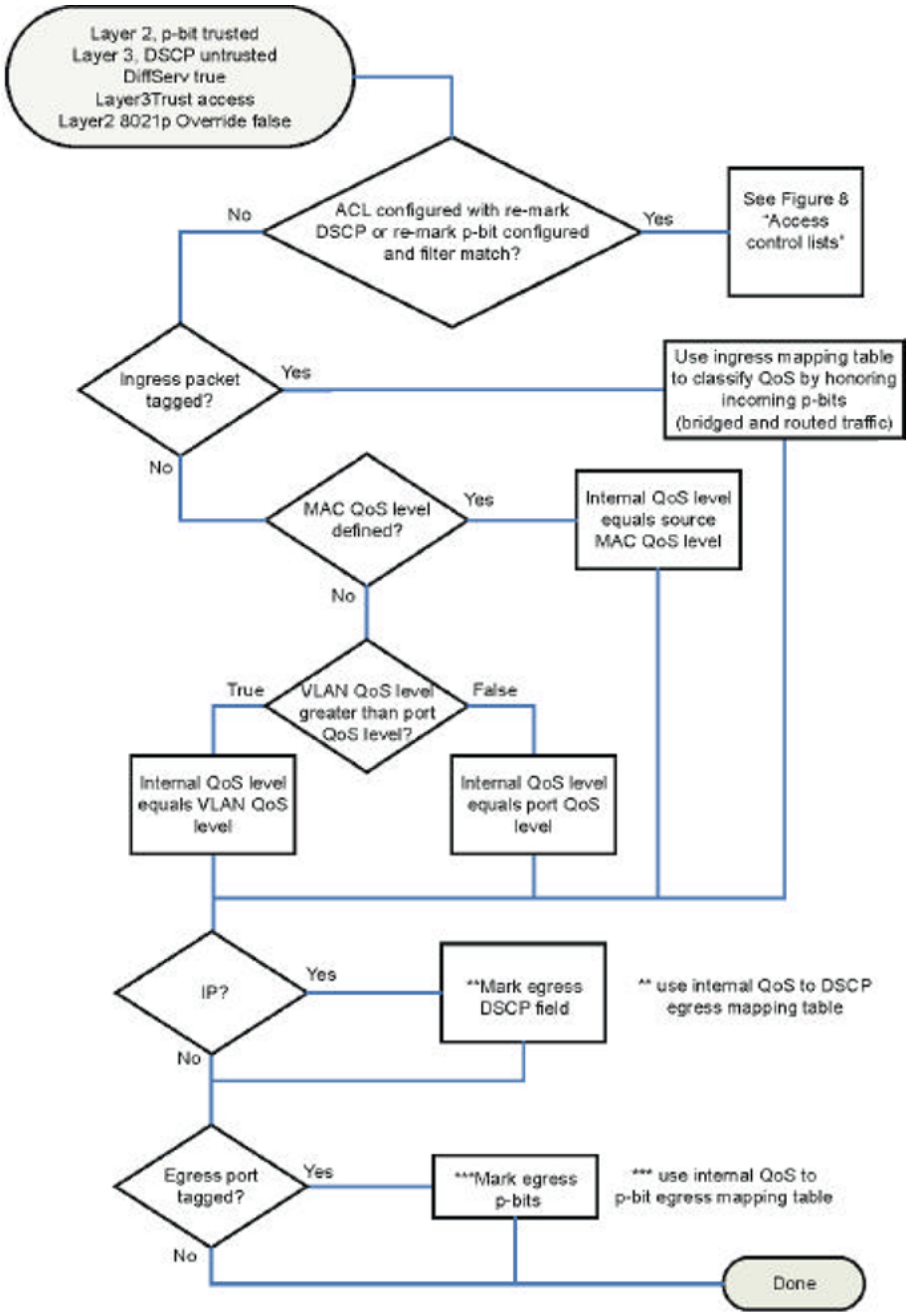


Figure 6: DiffServ access mode with 802.1p override disabled

DiffServ disabled

If you assign the DiffServ parameter the default of false (disabled), the L3 DSCP parameter is ignored. For more information about QoS operations when DiffServ is false, see Figure 7: DiffServ disabled on page 30.

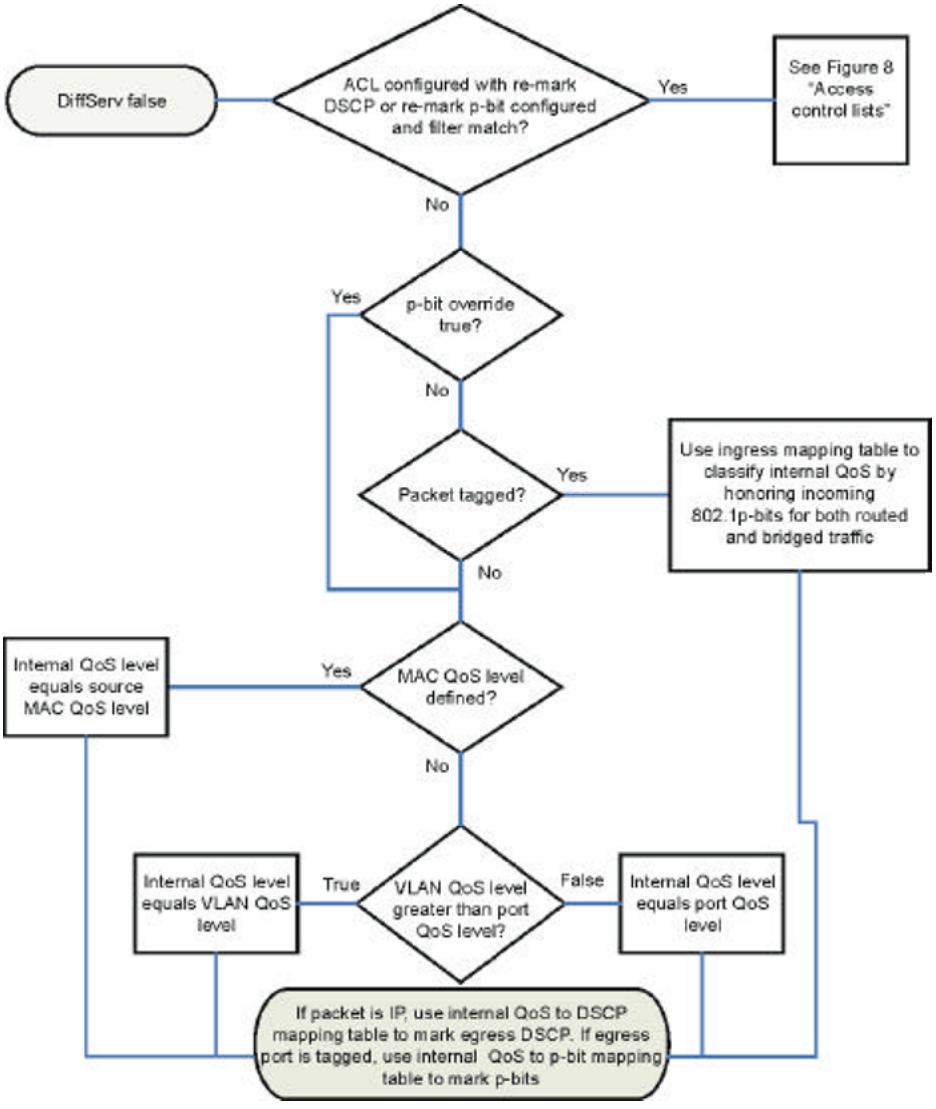


Figure 7: DiffServ disabled

DiffServ and ACLs

QoS (DiffServ) and filters operate independently; you need not use filters to provide QoS. However, filters can override QoS operations. The following figure shows how you can use ACLs to change packet QoS characteristics.

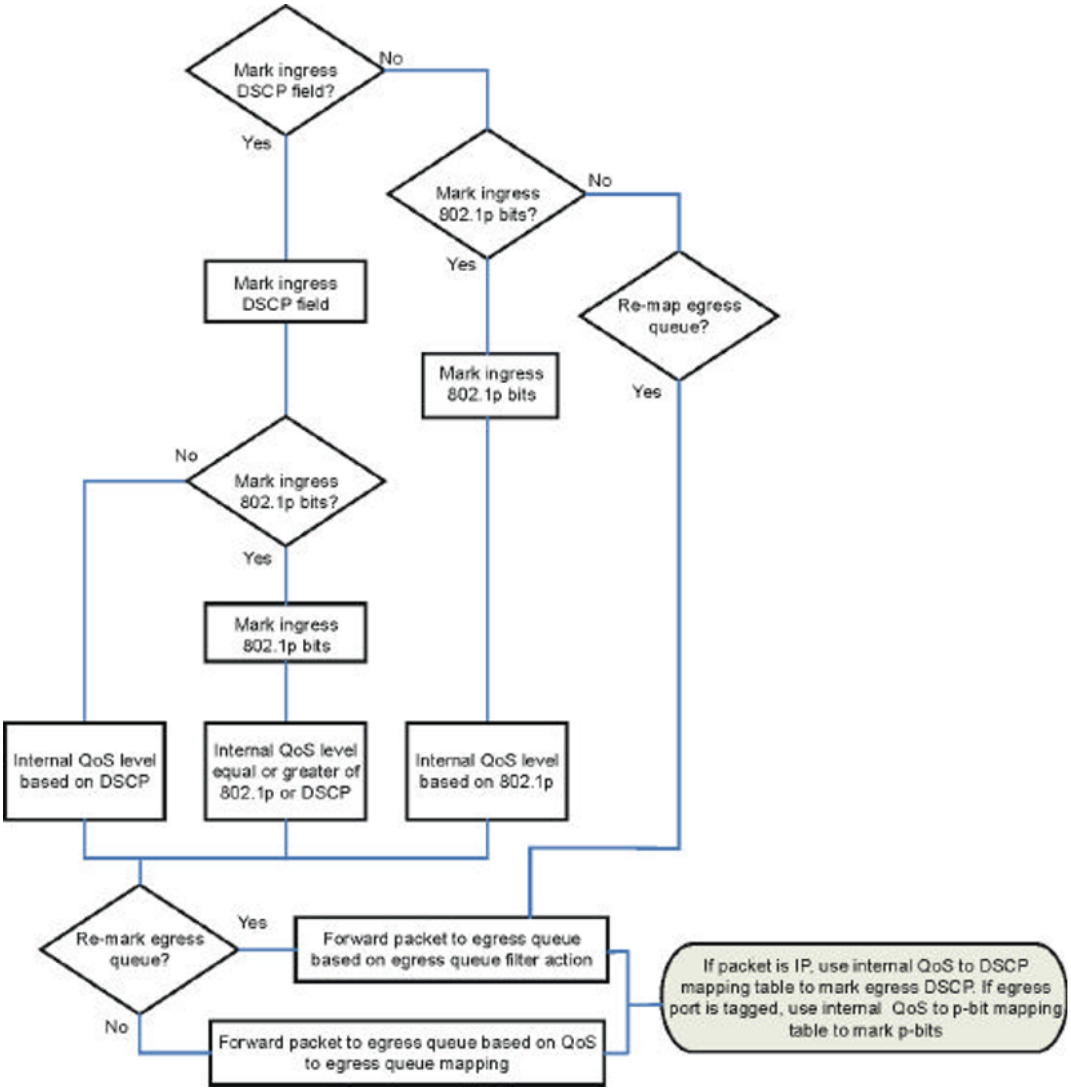


Figure 8: Access control lists

Queueing

Queuing is a congestion-avoidance function that prioritizes packet delivery. Queuing ensures discriminate packet discard during network congestion and can delay a packet in memory until the scheduled transmission.

You can use queuing to manage congestion. Queueing determines the order in which an interface sends packets based on priorities assigned to those packets. Congestion management activities include the creation of queues, the assignment of packets to the queues based on packet classification, and the scheduling of packets in a queue for transmission.

When no congestion exists (periods of low traffic volume), an interface sends packets after they arrive. During periods of transmission congestion at the outgoing interface, packets arrive faster than the interface can send them. If you use congestion management features, packets that accumulate at an interface form a queue until the interface can send them. The packets follow a transmission schedule according to the assigned priority and the queuing mechanism configured for the interface. The Avaya Ethernet Routing Switch 8800/8600 scheduler determines the order of packet transmission by controlling how queues are handled with respect to each other.

Feedback output queueing

The FOQ mechanism helps the Avaya Ethernet Routing Switch 8800/8600 avoid switch fabric congestion. The Ethernet Routing Switch 8800/8600 monitors and reports congestion for individual egress queues. The FOQ mechanism notifies the ingress ports of possible future switch fabric congestion. If an egress queue becomes congested, FOQ restricts the packet flow to that queue. The switch fabric does not waste resources forwarding packets that will be dropped.

FOQ avoids packet drops indiscriminate of QoS flows, which provides fair congestion management. Old switches base congestion management on the Class of Service (CoS) and cannot distinguish offending traffic from correctly functioning traffic if they both have the same CoS level. Switches based on CoS congestion management also cannot distinguish offending traffic from well-behaved traffic on the lane (fabric PID) level. Thus, in old systems, all queues of the same PID can suffer from packet drops because of congestion. The switch uses FOQ for fine control over congestion; it can manage congestion for each queue. In FOQ systems, congestion in an egress queue only affects that queue; it does not affect packets destined for noncongested queues.

Egress queue sets

The egress queue set is a logical bundle of configuration queues; it is a template that you use to apply the same queue configuration to a group (set) of ports available on multiple input and output (I/O) modules. All ports that you add to an egress queue set use identical configuration queues.

You can use the following two templates to create an egress queue set:

- An eight-queue template: Configure up to eight queues on the 8648GTR, the 8648GBRS, the 8848GB, the 8648GTRS, the 8848GT, and the 10/100/1000 Mb/s ports of the 8634XGRS and 8834XG.
- A 64-queue template: Configure up to 64 queues on Gigabit and 10 Gigabit modules. These modules include the 8630GBR, the 8683XLR, the 8683XZR, the 8612XLRS, the 8812XL, and the 10 Gb/s Ethernet ports of the 8634XGRS and the 8834XG.

The Avaya Ethernet Routing Switch 8800/8600 I/O modules can use up to 8 or 64 queues.

Queues within the egress queue set use three queuing styles (see the following figure):

- high-priority group
- balanced-queuing group
- low-priority group

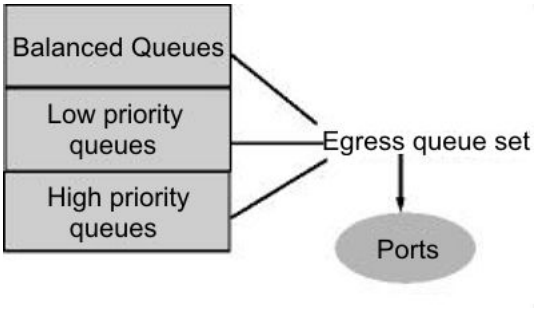


Figure 9: Queuing styles

For more information about queuing styles, see Queuing styles on page 38.

Avaya Data Solutions Service Classes

Avaya Data Solutions Service Classes (ADSSC) define a standard architecture to provide end-to-end QoS on a range of Avaya Ethernet switching and voice products. ADSSCs function as default QoS policies built in to a product. The ADSSCs incorporate the various QoS technologies to provide a complete end-to-end QoS behavioral treatment. The Avaya Ethernet Routing Switch 8800/8600 includes a built-in QoS implementation for ADSSCs.

Default egress queue sets (ADSSC templates)

ADSSCs provide default recommended settings and behaviors for queues on an output port. With the Avaya Ethernet Routing Switch 8800/8600, you can modify some of the default settings for each of these queues and create custom queues based on your specific needs.

The Ethernet Routing Switch 8800/8600 includes the following two reserved and preconfigured egress queue sets based on the ADSSCs model:

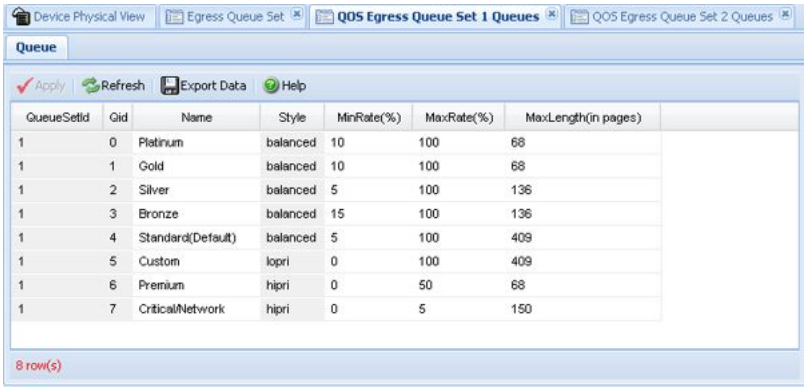
- Egress queue set 1 (eight-queue template)—used for modules with more than 10 ports for each lane.
- Egress queue set 2 (64-queue template)—used for modules with 10 ports or less for each lane.

For information about modules and lanes, see the following table.

Table 2: Modules and lanes

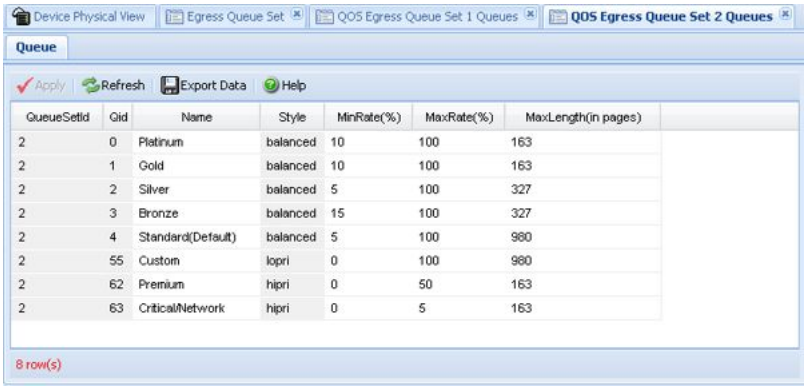
Module	Number of lanes
8612XLRS	3—each lane supports 4 XFP ports
8630GBR	3—each lane supports 10 SFP ports
8634XGRS	3—Lane 1 supports 4 RJ-45 ports and 12 SFP ports; Lane 2 supports 4 RJ-45 and 12 SFP ports, and Lane 3 supports 2 XFP ports
8648GBRS	3—each lane supports 16 SFP ports
8648GTR	2—one lane supports ports 1 to 24; the other supports ports 25 to 48
8648GTRS	2—one lane supports ports 1 to 24; the other supports ports 25 to 48
8683XLR and 8683XZR	3—each lane supports 1 XFP port
8812XL	3—each lane supports 4 SFP+ ports
8834XG	3—Lane 1 supports 4 RJ-45 ports and 12 SFP ports; Lane 2 supports 4 RJ-45 and 12 SFP ports, and Lane 3 supports 2 XFP ports
8848GB	3—each lane supports 16 SFP ports
8848GT	2—one lane supports ports 1 to 24; the other supports ports 25 to 48

The Ethernet Routing Switch 8800/8600 includes eight preconfigured queues (corresponding to the eight ADSSCs) on each port of a module. Figure 10: Preconfigured egress queue set 1 on page 35 shows the eight preconfigured queues of the eight-queue template. Figure 11: Preconfigured egress queue set 2 on page 35 shows the eight preconfigured queues of the 64 queue template. You can also use the CLI command **`show qos config egress-queue-set`** to view the queue sets.



QueueSetId	Qid	Name	Style	MinRate(%)	MaxRate(%)	MaxLength(in pages)
1	0	Platinum	balanced	10	100	68
1	1	Gold	balanced	10	100	68
1	2	Silver	balanced	5	100	136
1	3	Bronze	balanced	15	100	136
1	4	Standard(Default)	balanced	5	100	409
1	5	Custom	lopri	0	100	409
1	6	Premium	hipri	0	50	68
1	7	Critical/Network	hipri	0	5	150

Figure 10: Preconfigured egress queue set 1



QueueSetId	Qid	Name	Style	MinRate(%)	MaxRate(%)	MaxLength(in pages)
2	0	Platinum	balanced	10	100	163
2	1	Gold	balanced	10	100	163
2	2	Silver	balanced	5	100	327
2	3	Bronze	balanced	15	100	327
2	4	Standard(Default)	balanced	5	100	980
2	55	Custom	lopri	0	100	980
2	62	Premium	hipri	0	50	163
2	63	Critical/Network	hipri	0	5	163

Figure 11: Preconfigured egress queue set 2

The Queue IDs (Qid) for R, RS, and 8800 modules support 64 queues, numbered from 0 to 63.

The Ethernet Routing Switch 8800/8600 R, RS and 8800 series modules support up to 8 or 64 queues. You can use the eight preconfigured queues, or you can create custom queues. On R, RS, and 8800 modules, you can configure the minimum rate, maximum rate, and maximum queue length parameters for the queues.

The minimum rate parameter does not apply to the preconfigured high- or low-priority queues. On the 64 queue set modules, you cannot change the minimum rate for queues 55, 62, and 63. On the eight queue set modules, you cannot change the minimum rate for queues 5, 6, and 7.

If you choose to use custom queues, adhere to the following guidelines:

- Avaya recommends that you always use at least eight queues for a module to avoid possible issues with the DSCP to QoS mappings.
- You must include at least one balanced queue in each set.
- You must have at least one high-priority queue to handle network or critical traffic.

- Each set must include a balanced queue with a Qid of 0.
- You cannot configure the Qid; you can configure the number of queues for each queueing style. The switch automatically assigns the Qid based on the number of each queueing style you choose.

For a VLAN traffic shaping configuration example using egress queue sets, see *VLAN Traffic Shaping for ERS 8800/8600 Technical Brief, NN48500-557*, available on the Avaya Technical Support Web site.

ADSSC types in the egress queue set

In the ADSSC domain, the egress queue set uses the following traffic classifications:

- network control traffic (Critical or Network)
- subscriber traffic (Premium, Metal, or Standard)

Critical or Network ADSSC

The switch uses the Critical or Network ADSSC for traffic within a single administrative network domain. If such traffic does not get through, the network cannot function. Examples of such types of traffic are heartbeats between core network switches or routers. The Spanning Tree Bridge Protocol Data Units (BPDU) use the Critical ADSSC to enter and exit the Avaya Ethernet Routing Switch 8800/8600. ADSSCs include network control traffic packets for OSPF, BGP, STP, and other protocols.

Premium ADSSC

The switch uses the Premium ADSSC for IP telephony services, and provides the low latency and low jitter required to support the services. IP telephony services include Voice over IP (VoIP), voice signaling, Fax over IP (FoIP), and voice-band data services over IP (for example, analog modem). The switch can also use the Premium ADSSC for Circuit Emulation Services over IP (CESoIP).

Metal ADSSCs

The Platinum, Gold, Silver, and Bronze ADSSCs are collectively referred to as the metal classes. The metalADSSCs provide a minimum bandwidth guarantee and are useful for variable bit rate or bursty types of traffic. Applications that use the metal ADSSCs support mechanisms that dynamically adjust their transmit rate and burst size based on congestion (packet loss) detected in the network.

Comments? infodev@avaya.com

Platinum ADSSC

The switch uses the Platinum ADSSC for applications that require low latency, for example, real-time services such as video conferencing and interactive gaming. Platinum ADSSC traffic provides the low latency required for interhuman (interactive) communications. The Platinum ADSSC provides a minimum bandwidth assurance for Assured Forwarding 41 (AF41) and Class Selector 4 (CS4)-marked flows. When the network experiences congestion, DiffServ nodes use drop precedence to control variable bit rates that exceed the minimum assured bandwidth.

Gold ADSSC

The switch uses the Gold ADSSC for applications that require near-real-time service and are not as delay-sensitive as applications that use the Platinum service. Such applications include streaming audio and video, video on demand, and surveillance video.

The Gold ADSSC is based on the assumption that the source and destination buffer traffic and, therefore, the traffic is less sensitive to delay and jitter. By default, the Gold ADSSC provides a minimum bandwidth assurance for AF31, AF32, AF33, and CS3-marked flows. When the network experiences congestion, DiffServ nodes use drop precedence to control variable bit rates and burst sizes that exceed the minimum assured bandwidth.

Silver ADSSC

The switch uses the Silver ADSSC for responsive (typically client- and server-based) applications. Such applications include Systems Network Architecture (SNA) terminals (for example, a PC or Automatic Teller Machine) to mainframe (host) transactions that use Data Link Switching (SNA over IP), Telnet sessions, Web-based ordering and credit card processing, financial wire transfers, and Enterprise Resource Planning applications.

Silver ADSSC applications require a fast response and have asymmetrical bandwidth needs. The client sends a short message to the server and the server responds with a much larger data flow to the client. For example, after a user clicks a hyperlink (that sends a few dozen bytes) on a Web page, the Web browser loads a new Web page (that downloads kilobytes of data). The Silver ADSSC provides a minimum bandwidth assurance for AF21- and CS2-marked flows.

The Silver ADSSC favors short-lived, low-bandwidth TCP-based flows. During network congestion, DiffServ nodes use drop precedence to control variable bit rates and burst sizes that exceed the minimum assured bandwidth.

Bronze ADSSC

The switch uses the Bronze ADSSC for long-lived TCP-based flows, such as file transfers, e-mail, or noncritical Operation, Administration, and Maintenance (OAM) traffic. The Bronze ADSSC provides a minimum bandwidth assurance for AF11- and CS1-marked flows. During network congestion, DiffServ nodes use drop precedence to control variable bit rates and burst sizes that exceed the minimum assured bandwidth. Avaya recommends that you use the Bronze ADSSC for noncritical OAM traffic with the CS1 DSCP marking.

Standard ADSSC

The switch uses the Standard ADSSC for best-effort services. Avaya does not specify delay, loss, or jitter guarantees for this ADSSC.

Queuing styles

The Avaya Ethernet Routing Switch 8800/8600 I/O modules can have up to 8 or 64 queues for each port. The switch bundles queues together based on queuing styles. The queue numbering order is as follows:

- high-priority queues
- low-priority queues
- balanced queues

High-priority queues have the highest priority. Queues that are members of this group take precedence over the queues in all other queuing groups. The strict (high) priority group is always guaranteed service first and has the lowest latency among the groups. The queuing scheduler immediately handles packets that enter the strict-priority queues to transmit those packets at the highest priority.

For 64 queue set queues, the strict-priority queues numbers start from queue index 63 and decrement. For 8 queue set queues, the strict-priority queues numbers start from queue index 7 and decrement. In Figure 12: High-priority queues 62 and 63 on page 39, queues 62 and 63 are members of a strict-priority group. The scheduler handles a packet that enters queue 63 at the highest priority. After the scheduler transmits packets in queue 63, it handles queue 62.

The scheduler handles queues within the high-priority queue group in priority order. A higher queue number corresponds to a higher priority.

Comments? infodev@avaya.com

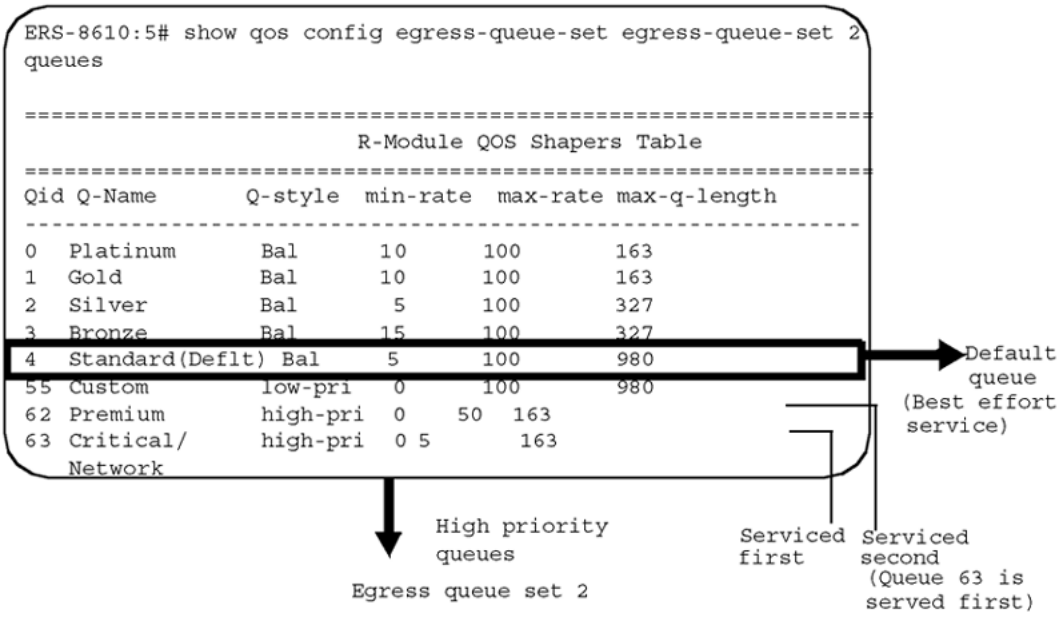


Figure 12: High-priority queues 62 and 63

Queue 63 is reserved for Critical or Network Control traffic. For example, Spanning Tree BPDUs and topology updates are placed in queue 63. Queue 62 is the next highest priority queue and carries latency-sensitive subscriber traffic. For example, VoIP and video conferencing applications use Premium queue 62.

By default on trusted ports, incoming packets with 802.1p equal to 6, or DSCP markings of CS5 or Expedited Forwarding (EF), are placed in queue 62 to ensure timely service.

You can configure the max-rate parameter to bind output traffic to the specified limit. The switch either delays (if the buffer is not full) or drops traffic that violates this limit; see Figure 13: Queues bounded by max-rate parameter on page 40). By default, high-priority queues use a maximum rate based on the ADSSC recommendations. Figure 10: Preconfigured egress queue set 1 on page 35 and Figure 11: Preconfigured egress queue set 2 on page 35 show the default max-rate parameters. For high-priority queues, a non-100-percent maximum rate ensures that a malfunctioning client application does not use the entire port bandwidth.

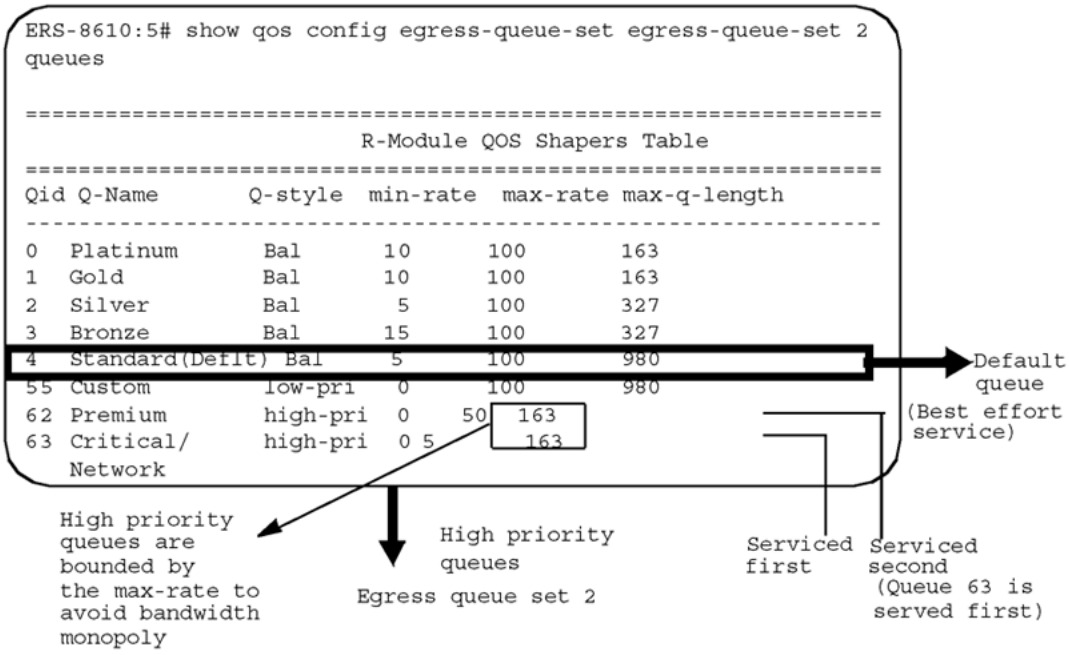


Figure 13: Queues bounded by max-rate parameter

By default, high-priority queues use a max-rate based on ADSSC recommendations. In the default ADSSC queuing template (egress-queue-set 2), high-priority queue 63 uses a max-rate of 5 percent, whereas queue 62 uses a max-rate of 50 percent.

Minimum rate values do not apply to high-priority queues. The following table shows examples of high-priority queues.

Table 3: High-priority queues in the 64-queue template

Queue	Name	Description
Queue 63	Network	Reserved for Critical or Network traffic
Queue 62	Subscriber	Recommended for latency-sensitive subscriber traffic, for example, VoIP

You can increase the max-rate on high-priority queues (see the following figure).

```
ERS-8610:5# config qos egress-queue-set 2 queue 62
ERS-8610:5/config/qos/egress-queue-set/2/queue/62# set max-rate 60
Available Maximum Rate is 50%, oversubscribed by 10%
Minimum rates are not guaranteed
ERS-8610:5/config/qos/egress-queue-set/2/queue/62#
```

Figure 14: Increase in maximum rate on high-priority queues

The warning message that appears can occur when you modify the default max-rate on high-priority queues. Because high-priority queues have precedence over balanced queues, you must follow this rule when you configure the max-rate on high-priority queues. The maximum

rate must be less than or equal to the available bandwidth minus the total minimum rate for the balanced queues.

To increase the max-rate on high-priority queues, decrease the minimum rate on the balanced queues as shown in Configuring an egress queue set on page 93. Then, increase the max-rate as described in Configuring an egress queue set on page 93. The following figure shows this configuration process.

```
ERS-8610:5# ERS-8610co:5# config qos egress-queue-set 2 queue 0
ERS-8610:5/config/qos/egress-queue-set/2/queue/0# set min-rate 5
ERS-8610:5/config/qos/egress-queue-set/2/queue/0# queue 1
ERS-8610:5/config/qos/egress-queue-set/2/queue/1# set min-rate 5
ERS-8610:5/config/qos/egress-queue-set/2/queue/1# queue 2
ERS-8610:5/config/qos/egress-queue-set/2/queue/2# set min-rate 5
ERS-8610:5/config/qos/egress-queue-set/2/queue/2# queue 62
ERS-8610:5/config/qos/egress-queue-set/2/queue/62# set max-rate 60
ERS-8610:5/config/qos/egress-queue-set/2/queue/62# top
ERS-8610:5# show qos config egress-queue-set egress-queue-set 2 queues

================================================================================
                         R-Module QOS Shapers Table
================================================================================
Qid Q-style  min-rate   max-rate   max-q-length
--------------------------------------------------------------------------------
0   Bal      10         100        163
1   Bal      10         100        163
2   Bal      5          100        327
3   Bal      15         100        327
4   Bal      5          100        980
55  low-pri  0          100        980
62  high-pri 0          50         163
63  high-pri 0          5          163
```

Figure 15: Decrease in minimum rate of balanced queues

Low-priority queues have the lowest priority, with a minimum rate of 0. High-priority and balanced queues take precedence over low-priority queues. This queue corresponds to best-effort traffic.

A weighted fair queueing (WFQ) scheduler handles balanced queues. A WFQ scheduler handles queues in a round-robin fashion (each queue in turn), where each queue receives bandwidth in proportion to the weight. The minimum rate you configure for the queue determines the weight and service time of the queue.

The minimum rate guarantees that the queues receive the configured bandwidth. The min-rate is a promise to the subscriber that the queue receives at least the percentage of bandwidth share configured for that queue. If no additional data exists on other queues, the rate on a queue can increase to the max-rate configured for the queue. For example, if you configure a queue for a 10 percent minimum rate on a 1 Gb/s port, the scheduler guarantees that the queue receives a fair share of 100 Mb/s from the available output port bandwidth.

To guarantee minimum configured rates, the sum of minimum rates for balanced queues and maximum rates for high-priority queues must not exceed 100 percent. Balanced queues permit oversubscription but do not guarantee minimum rates.

Minimum rates do not apply to high-priority groups. The switch handles high-priority traffic up to the max-rate limit. By default, minimum rates on balanced queues are based on the ADSSC recommendations; see Figure 16: Minimum rates on balanced queues on page 42. For more information, see Egress queue set minimum rate on page 60.

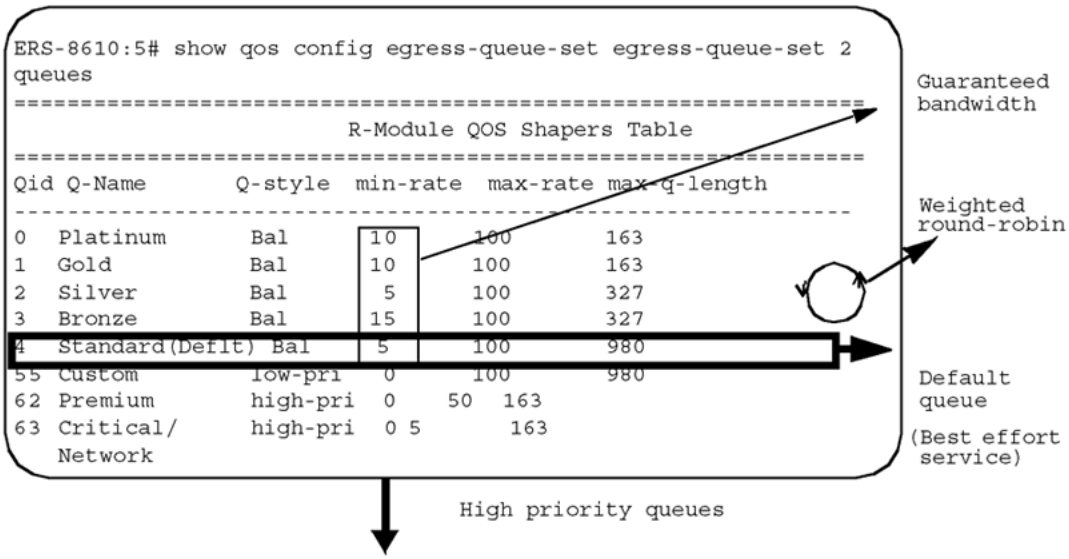


Figure 16: Minimum rates on balanced queues

You can configure the max-rate parameter to bind the output traffic to the specified limit. The system either delays (if the buffer is not full) or drops traffic that violates this limit. By default, high-priority queues use a maximum rate based on the ADSSC recommendations. Balanced and low-priority queues use a maximum rate of 100 percent. Figure 10: Preconfigured egress queue set 1 on page 35 and Figure 11: Preconfigured egress queue set 2 on page 35 show the default max-rate parameters. For high-priority queues, a non-100-percent maximum rate ensures that a malfunctioning client application does not use the entire port bandwidth.

You can modify the default max-rates on all queues. High-priority queues have precedence over balanced queues, and balanced queues take precedence over low-priority queues. To guarantee that balanced queues obtain the promised minimum rates, ensure that the maximum rate on high-priority queues is less than or equal to the available data rate minus the total minimum rate for the balanced queues.

The minimum rate guarantees that the queue receives the configured bandwidth. The min-rate is a promise to the subscriber that a queue receives at least the percentage of bandwidth share configured for that queue. If no data to service exists on other queues, the rate on a queue can increase to the max-rate configured on the queue.

For example, if you configure a balanced queue for a 10 percent min-rate on a 1 Gb/s port, the scheduler provides the queue with a fair share of at least 100 Mb/s from the available output port bandwidth. Minimum rates do not apply to high-priority or low-priority queueing styles. Incoming high-priority traffic is serviced at up to the max-rate limit. Low-priority queues always have a min-rate of 0; no guaranteed rates exist for low-priority traffic. By default, minimum rates for balanced queues are based on the ADSSC recommendations, see Figure 10:

Comments? infodev@avaya.com

Preconfigured egress queue set 1 on page 35 and Figure 11: Preconfigured egress queue set 2 on page 35.

The Avaya Ethernet Routing Switch 8800/8600 supports 32 000 memory pages (queues) for each forwarding lane. Each memory page is 512 bytes in length, except the first page, which is 144 bytes in length. For information about modules and lanes, see Table 2: Modules and lanes on page 34.

You can change the default maximum queue length (max-q-length) parameter. However, such changes can cause an oversubscription of available buffers, depending on module types and configurations. You can use leftover queue lengths from some queues to increase the buffer size of other queues. Use the **show port stats** command to view port queue statistics (see the following figure). Increase the max-q-length for any port with a queue that shows a nonzero value in the dropped pages parameter.

The default max-q-length settings are based on real-world (generalized) traffic patterns, and the traffic patterns and queue usage for a specific user can vary widely. Therefore, adjust the max-q-length parameter depending upon user traffic patterns and queue configurations.

```
ERS-8610:5# show port stats egress-queues 8/1 verbose

================================================================================
              R-Module QOS Egress Rate-Limit Stats Table
================================================================================
Port   Qid  Total pages  Dropped pages      Utilization
--------------------------------------------------------
8/1    0      10000         10                  1
8/1    1      20000         0                   2
8/1    2      5000          0                   0.3
8/1    3      80000         500                 5
8/1    4      600000        1000                37
8/1    5      0             0                   0
8/1    6      0             0                   0
8/1    7      0             2                   0
8/1    54     0             0                   0
8/1    63     1000          0                   0.1
8/1    64     500           0                   0.05

Page Size: 512 Bytes
```

Figure 17: show port stats egress-queues output

The utilization parameter is calculated for an individual port and for each queue.

For more information about QoS statistics, see *Avaya Ethernet Routing Switch 8800/8600 Performance Management,* (NN46205-704).

Egress queue packet assignment

The Avaya Ethernet Routing Switch 8800/8600 assigns packets to egress (transmit) queues based on the ingress mappings and the internal QoS level.

Ingress mappings and queues

The switch uses ingress maps to translate incoming packet QoS markings to the internal QoS level. The switch classifies packets based on the internal QoS level.

Ingress mappings are as follows:

- 802.1p to (internal) QoS level
- DSCP to (internal) QoS level
- EXP-bit to (internal) QoS level

The following tables show ingress mappings obtained using the CLI command **`show qos ingressmap`**. Table 5: Default ingress 802.1p to QoS to egress queue mappings on page 44 shows ingress IEEE 1p to QoS level mappings.

Table 6: Gigabit Ethernet default ingress DSCP to QoS to egress queue mapping on page 45 shows DSCP to internal QoS-level mappings.

The following table shows MPLS EXP-bit mappings.

Table 4: QoS ingress MPLS Exp bit to QoS-level map

MPLS Exp bit	QoS level
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7

The following tables describe default ingress and egress mappings.

Table 5: Default ingress 802.1p to QoS to egress queue mappings

Internal QoS	Egress queue		PHB	Queue name (Egress Queue Set 2)	Default 1p remarking on egress	Network Service Class (NSC)
	8 queue ports	64 queue ports				
0	5	55	Custom	Custom	1	Custom
1	4	4	CS0/DF	Standard	0	Standard

Internal QoS	Egress queue		PHB	Queue name (Egress Queue Set 2)	Default 1p remarking on egress	Network Service Class (NSC)
	8 queue ports	64 queue ports				
2	3	3	CS1/AF11	Bronze	2	Bronze
3	2	2	CS2/AF21	Silver	3	Silver
4	1	1	CS3/AF31	Gold	4	Gold
5	0	0	CS4/AF41	Platinum	5	Platinum
6	6	62	CS5/EF	Premium	6	Premium/EF
7	7	63	CS6/CS7	Network (or Critical)	7	Premium/EF

In the following table, TOS denotes Type of Service and Hex denotes hexadecimal.

Table 6: Gigabit Ethernet default ingress DSCP to QoS to egress queue mapping

Ingress				Internal QoS level	PHB level	Queue name (Egress Queue Set 2)
DSCP	DSCP (bin)	DSCP (Hex)	TOS			
00	000000	00	00	1	CS0	Custom
00	000000	00	00	1	DF	
01	000001	01	04	1	CS0	
02	000010	02	08	1	CS0	
03	000011	03	0C	1	CS0	
04	000100	04	10	1	CS0	
05	000101	05	14	1	CS0	
06	000110	06	18	1	CS0	
07	000111	07	1C	1	CS0	
08	001000	08	20	2	CS1	Bronze
09	001001	09	24	1	CS0	Custom
10	001010	0A	28	2	AF11	Bronze
11	001011	0B	2C	1	CS0	Custom
12	001100	0C	30	2	CS1	Bronze
13	001101	0D	34	1	CS0	Custom
14	001110	0E	38	2	CS1	Bronze
15	001111	0F	3C	1	CS0	Custom

Ingress				Internal QoS level	PHB level	Queue name (Egress Queue Set 2)
DSCP	DSCP (bin)	DSCP (Hex)	TOS			
16	010000	10	40	3	CS2	Silver
17	010001	11	44	1	CS0	Custom
18	010010	12	48	3	AF21	Silver
19	010011	13	4C	1	CS0	Custom
20	010100	14	50	3	CS2	Silver
21	010101	15	54	1	CS0	Custom
22	010110	16	58	3	CS2	Silver
23	010111	17	5C	1	CS0	Custom
24	011000	18	60	4	CS3	Gold
25	011001	19	64	1	CS0	Custom
26	011010	1A	68	4	AF31	Gold
27	011011	1B	6C	4	CS3	
28	011100	1C	70	4	CS3	
29	011101	1D	74	1	CS0	Custom
30	011110	1E	78	4	CS3	Gold
31	011111	1F	7C	1	CS0	Custom
32	100000	20	80	5	CS4	Platinum
33	100001	21	84	1	CS0	Custom
34	100010	22	88	5	AF41	Platinum
35	100011	23	8C	5	CS4	
36	100100	24	90	5	CS4	
37	100101	25	94	1	CS0	Custom
38	100110	26	98	5	CS4	Platinum
39	100111	27	9C	1	CS0	Custom
40	101000	28	A0	5	CS4	Platinum
41	101001	28	A4	5	CS4	Platinum
42	101010	2A	A8	1	CS0	Custom
43	101011	2B	AC	1	CS0	
44	101100	2C	B0	1	CS0	
45	101101	2D	B4	1	CS0	

Ingress				Internal QoS level	PHB level	Queue name (Egress Queue Set 2)
DSCP	DSCP (bin)	DSCP (Hex)	TOS			
46	101110	2E	B8	6	EF	Premium
47	101111	2F	BC	6	CS5	
48	110000	30	C0	7	CS6	Network (or Critical)
49	110001	31	C4	1	CS0	Custom
50	110010	32	C8	1	CS0	
51	110011	33	CC	1	CS0	
52	110100	34	D0	1	CS0	
53	110101	35	D4	1	CS0	
54	110110	36	D8	1	CS0	
55	110111	37	DC	1	CS0	
56	111000	38	E0	7	CS7	Network (or Critical)
57	111001	39	E4	1	CS0	Custom
58	111010	3A	E8	1	CS0	
59	111011	3B	EC	1	CS0	
60	111100	3C	F0	1	CS0	
61	111101	3D	F4	1	CS0	
62	111110	3E	F8	1	CS0	
63	111111	3F	FC	1	CS0	

The following table describes mappings for MPLS-based QoS.

Table 7: Default ingress EXP-bit to QoS to egress queue mappings

EXP-bit	Internal QoS	Egress queue	Queue name (Egress Queue Set 2)
0	0	55	Custom
1	1	4	Standard (or Default)
2	2	3	Bronze
3	3	2	Silver
4	4	1	Gold
5	5	0	Platinum
6	6	62	Premium

EXP-bit	Internal QoS	Egress queue	Queue name (Egress Queue Set 2)
7	7	63	Network (or Critical)

Internal QoS level

The internal QoS level or effective QoS level is a key element in the Ethernet Routing Switch 8800/8600 QoS architecture. The internal QoS level specifies the kind of treatment a packet receives and the transmit queue for the exit (egress) path. The Ethernet Routing Switch 8800/8600 classifies and assigns an internal QoS level to every packet that enters the switch.

Internal QoS levels map to the transmit or egress queues on a port. For example, for an access port, the highest value among the port QoS level, VLAN QoS level, and MAC QoS level becomes the internal QoS level (effective QoS level). For Layer 3 trusted (core) ports, the switch honors incoming DSCP and TOS bits. The ingress DSCP to QoS level map determines the internal QoS level assignment. If you configure a MAC QoS level on an untrusted port, it takes precedence over the VLAN QoS level and the port QoS level.

The following figure shows a i2002 VoIP phone that sends packets with a 802.1p value of 6 on a trusted Layer 2 port. The 802.1p-to-QoS level ingress map determines the internal QoS level of the packet and places the packet in the appropriate queue using the QoS level to queue mapping table.

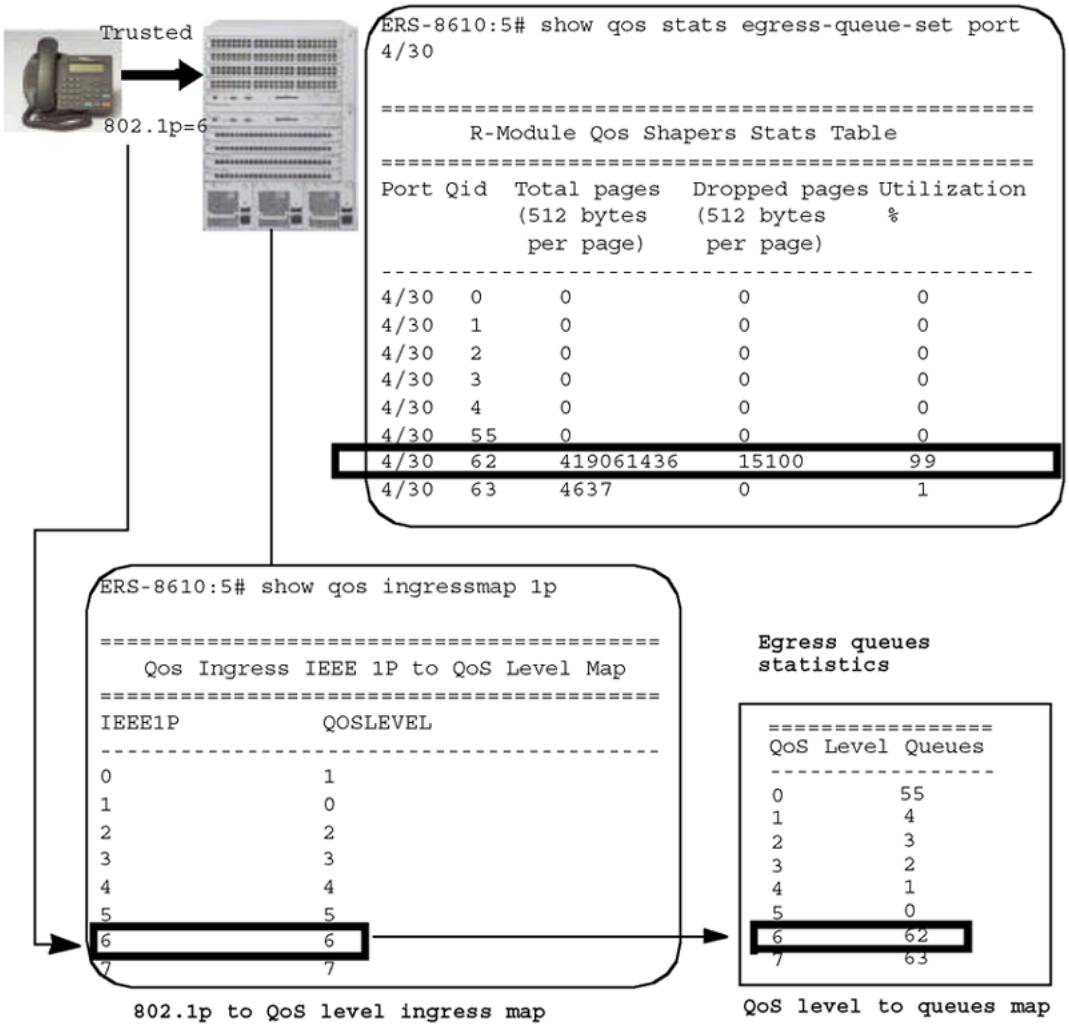


Figure 18: Path from input port to queues

The internal QoS level maps to the transmit queues. The following table shows the default mapping of internal QoS level to egress queue for the R, RS, and 8800 modules.

Table 8: QoS level to queue mapping for each module

	8683XLR, 8683XZR, 8630GBR, 8612XLRS, 8812XL, and 10 Gb/s ports of the 8634XGRS, and 8834XG	8648GTR, 8648GTRS, 8848GT, 8648GBRS, 8848GB, and 10/100/1000 Mb/s ports of the 8634XGRS and 8834XG
QoS level	**Queue**	**Queue**
0	55	5
1	4	4
2	3	3
3	2	2

	8683XLR, 8683XZR, 8630GBR, 8612XLRS, 8812XL, and 10 Gb/s ports of the 8634XGRS, and 8834XG	8648GTR, 8648GTRS, 8848GT, 8648GBRS, 8848GB, and 10/100/1000 Mb/s ports of the 8634XGRS and 8834XG
QoS level	**Queue**	**Queue**
4	1	1
5	0	0
6	62	6
7	63	7

Egress queueing and modules

Packets that egress from one module port can originate from another module port.

Although packets exit from the egress forward processing module, the ingress processor (the port processor of packet origin) determines the egress queue. The ingress forward processing module determines the egress queue ID based either on the packet DSCP or 802.1p markings or through the filter or port, VLAN, or MAC QoS levels (see the following table).

Table 9: Default QoS to egress queue mappings for each module

Internal QoS level and ADSSC	Ports with 8 queues for each port queue and style	Ports with 64 queues for each port queue and style	Classic queue
0, Custom (best effort)	5, Low priority	55, Low priority	0
1, Standard	4, Weighted	4, Weighted	1
2, Bronze	3, Weighted	3, Weighted	2
3, Silver	2, Weighted	2, Weighted	3
4, Gold	1, Weighted	1, Weighted	4
5, Platinum	0, Weighted	0, Weighted	6
6, Premium	6, High Priority	62, High Priority	5
7, Network	7, High Priority	63, High Priority	7

The internal QoS level determines the egress queue.

Queue numbers depend on module port types (ports with 8 queues for each port, or ports with 64 queues for each port). The central processor maintains the table that maps packet QoS level to egress queue, which depends on the port type.

If the packet on egress is tagged, the Avaya Ethernet Routing Switch 8800/8600 can remark the p-bits and the DSCP field as the packet leaves the port. The switch bases the remapping on either the default internal QOS to egress mappings as shown in the following table and

Comments? infodev@avaya.com

Table 5: Default ingress 802.1p to QoS to egress queue mappings on page 44, or through traffic filtering.

Table 10: Default egress internal QOS to DSCP

Internal QoS	Egress queue modules		PHB	Egress queue name	Default DSCP remarking on egress (decimal format)	Network Service Class (NSC)
	8 queue ports	64 queue ports				
0	5	55	Custom	Custom	0	Custom
1	4	4	CS0/DF	Standard	0	Standard
2	3	3	CS1/ AF11	Bronze	10	Bronze
3	2	2	CS2/ AF21	Silver	18	Silver
4	1	1	CS3/ AF31	Gold	26	Gold
5	0	0	CS4/ AF41	Platinum	34	Platinum
6	6	62	SC5/EF	Premium	46	Premium/EF
7	7	63	CS6/CS7	Network	46	Premium/EF

Policing and shaping

QoS for the Ethernet Routing Switch 8800/8600 R, RS and 8800 modules support the following two features for bandwidth management and traffic control:

- Ingress traffic policing—a mechanism that limits the number of packets in a stream that matches a classification
- Egress traffic shaping—the process that delays and transmits packets to produce an even and predictable flow rate

Each feature is important to deliver Differentiated Services (DiffServ) within a QoS network domain. Figure 19: Basic policer and shaper behavior on page 52 shows basic policing and shaping behavior.

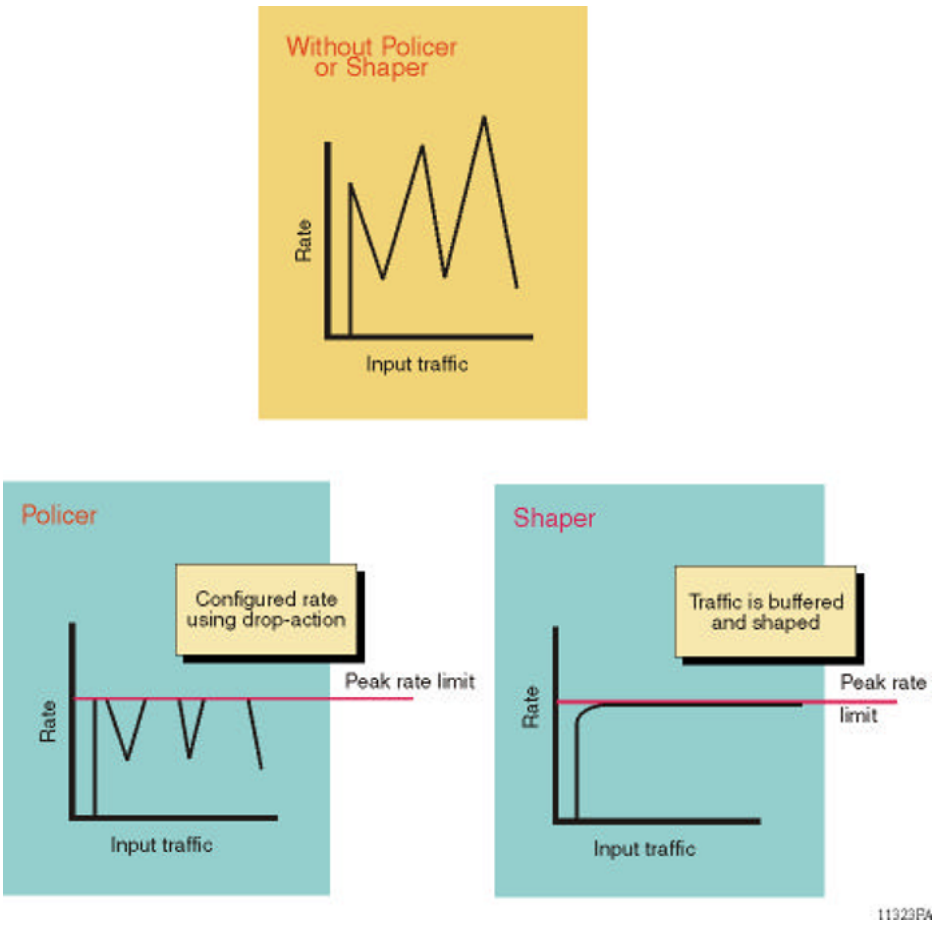


Figure 19: Basic policer and shaper behavior

Token buckets and policing

Tokens are a key concept in traffic control. A policer or shaper calculates the number of packets that pass and the data rate. Each packet corresponds to a token, and the policer or shaper transmits or passes the packet if the token is available (see Figure 20: Token flow on page 53).

The token container is like a bucket. In this view, the bucket represents both the number of tokens available for use instantaneously (the depth of the bucket) and the rate of token replenishment (how fast the bucket refills). The following figure shows the flow of tokens.

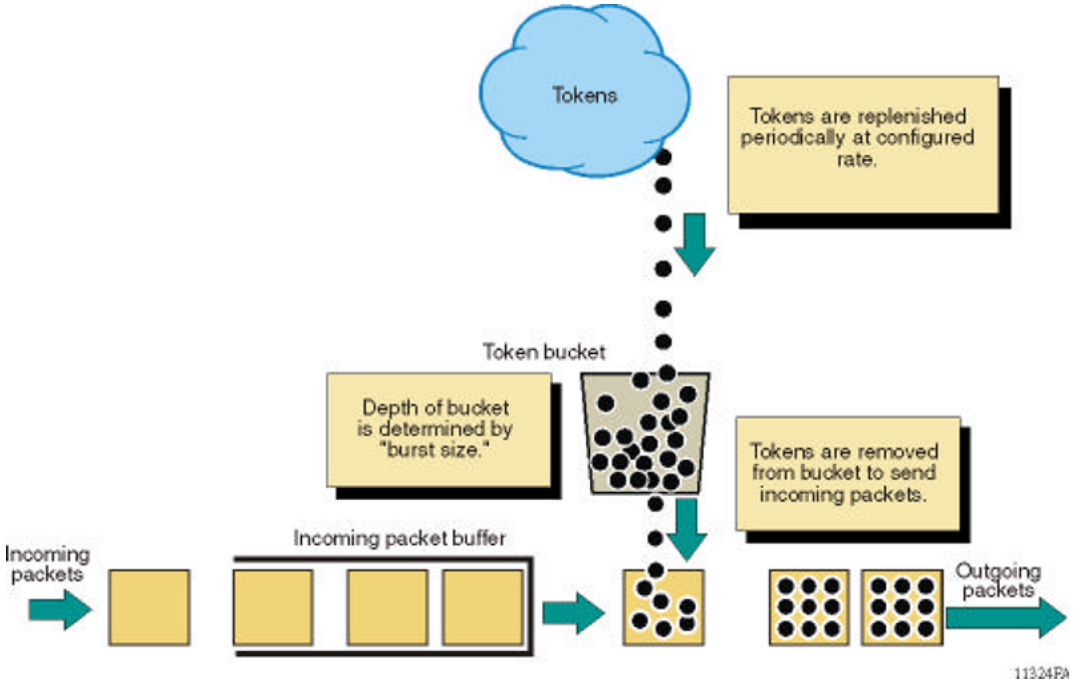


Figure 20: Token flow

In the Ethernet Routing Switch 8800/8600, each policer has two token buckets. One token bucket is for the peak rate and the other is for the service rate.

A token bucket permits bursty traffic and binds it. A bursty flow can use several tokens to sent the bursty transmission through. Hosts can save tokens to transmit, but never more tokens than the bucket can hold. When the bucket is full, the host discards the additional tokens. If no tokens are available, the sender must wait until one is available.

Policy-based policer versus shaper

Policy-based traffic policers and traffic shapers identify traffic by using a policy (a contract). Traffic that conforms to this policy (a service contract) is guaranteed transmission, and nonconforming traffic is considered in violation.

Policy-based policers and shapers differ in how they treat violations:

- Traffic shapers buffer and delay traffic that violates the contract.

  If no tokens are available in the token bucket, the traffic shaper delays packets until a token is available. Queueing buffers excessive packets and shapes the flow when the source data rate is higher than expected. The Avaya Ethernet Routing Switch 8800/8600 supports traffic shaping at the port level and for each transmit-queue (egress queue) level for outgoing (egress) traffic.

For more information about traffic shaping, see Queue-based traffic shaping on page 60.

- Traffic policers drop packets when traffic is excessive or re-mark the DSCP or 802.1p markings by using filter actions. Policing occurs at ingress.

  With the Ethernet Routing Switch 8800/8600, you can define multiple actions in case of traffic violation. For more information about traffic policing, see Policy-based traffic policing on page 54.

The following table summarizes the key differences between policing and shaping functions supported on the Ethernet Routing Switch 8800/8600.

Table 11: Policy-based policing versus shaping

Policing	Shaping
Apply at the ingress port.	Apply at the egress port.
Filter action can drop or re-mark excessive traffic. No buffering available.	Buffers excessive traffic and shapes the flow.
No individual queue policing.	Configure on each transmit queue level.
Supports RFC 2698—Two Rate Three Color Marker (trTCM). The RFC defines two rates: • Peak information rate (PIR) • Service rate Useful for policing of a service in which you must enforce a peak rate separately from a committed rate.	Supports one rate only.
You can perform traffic classification using filters.	Applies to egress queue. You can select egress queues through ingress filters. You cannot perform classification using filters.

Policy-based traffic policing

The Ethernet Routing Switch 8800/8600 R, RS and 8800 series modules support up to 450 policers, with 50 reserved internally for each lane. The 8683XLR, 8683XZR, or 8630GBR modules each support up to 1200 (1350 total) policy-based policers. For more information about modules and lanes, see Table 2: Modules and lanes on page 34.

The switch supports the following options:

- service rate limiting
- peak Information Rate limiting

Comments? infodev@avaya.com

- three internal colors to which to re-mark packets
  - red (discard right away)
  - yellow (discard if the network is congested)
  - green (forward)
- drop precedence during internal congestion

The switch supports ingress policing on port ACLs or VLAN ACLs. Port ACLs apply to individual port-based policers that are members of individual lanes. VLAN ACLs apply to global policers that are members of all lanes.

Policy-based policing in the Ethernet Routing Switch 8800/8600 offers three primary functions:

- rate limiting based on peak and service rates
- dropping packets in excess of the peak rate
- packet coloring as green, yellow, and red

Figure 21: Layer 2 to Layer 7 ingress policing on page 55 shows ingress policing operations. In this figure, the switch forwards packets classified as Expedited (E), colors them green, and does not drop a packet. The switch colors packets classified as Assured Forwarding (AF) as green, yellow, or red. The switch drops red packets immediately and drops yellow packets during congestion.

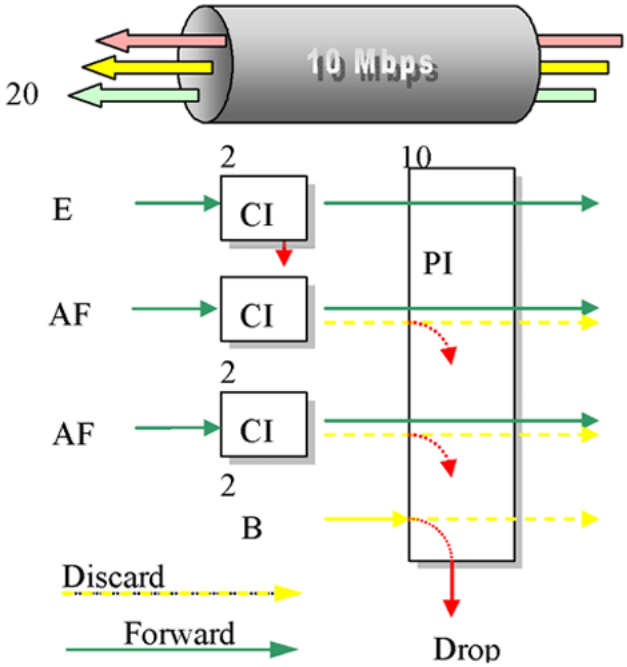


Figure 21: Layer 2 to Layer 7 ingress policing

In the preceding figure, CI denotes committed information (or service) rate, and PI denotes peak information rate. For more information about packet coloring, see Two Rate Three Color Marking on page 56.

Two Rate Three Color Marking

Ethernet Routing Switch 8800/8600 traffic policing supports RFC 2698 (Two Rate Three Color Marker—trTCM). The traffic policer meters a packet stream and marks packets either green, yellow, or red. The policer marks a packet red if it exceeds the peak rate. The policer marks a packet yellow if it exceeds the service rate, and green if it falls below that rate.

The policer assigns drop probabilities to packets in the red, yellow, and green zones. The switch is more likely to drop yellow packets during congestion than green packets.

The following figure shows that three color marking is useful for ingress policing of a service in which you must enforce a peak rate separately from a committed (service) rate.

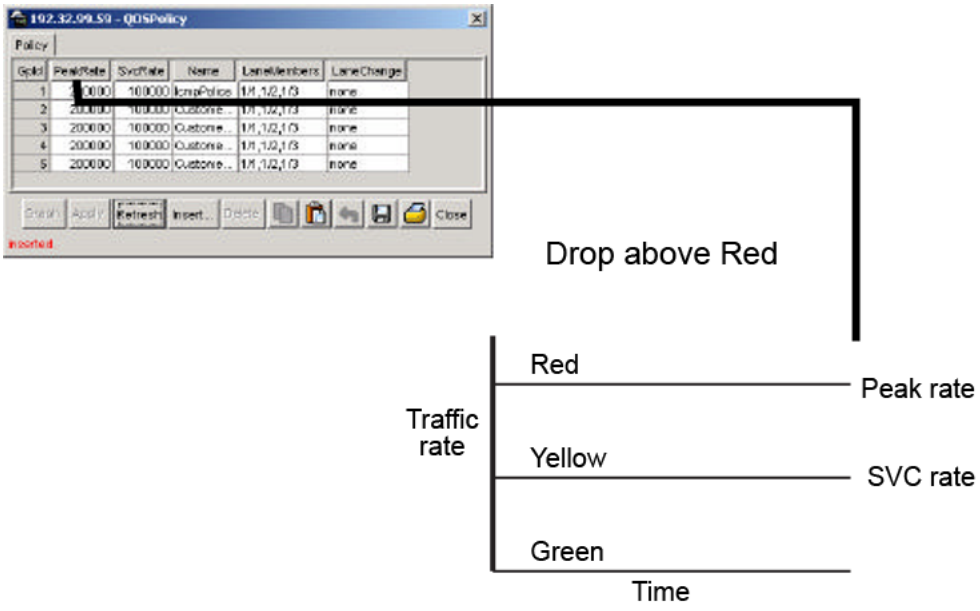


Figure 22: trTCM peak and service rates

Traffic policies

Policing ensures flow conformance with the rate metrics of configured policy. The policer drops the packets above the peak rate and recolors the packets above the service rate. When configuring traffic policies, you must define the peak and service rates.

For more information about how to configure traffic policies, see Configuring a policy-based policer on page 165 or Configuring a policy-based policer on page 92.

A policy is a template that defines policing characteristics. You can reference a policy by the global policy ID (GPID) or by the name. You can apply the policy to an individual port or to an

entire VLAN using an access control list (ACL). For more information, see Access control lists on page 72.

Lanes for policy-based policing

Traffic policies are global on the Ethernet Routing Switch 8800/8600. An individual port can use a single policy, or a group of ports can share the policy (an aggregate policer). For example, if a traffic policy specifies a peak rate of 500 Mb/s, and this traffic policy applies to ports 1/1 to 1/4, then the sum of the permitted input traffic from these ports cannot exceed the 500 Mb/s peak rate. You can implement aggregate policers on I/O modules by using lanes.

The following figure shows three lanes on an 8630GBR module, each consisting of ten 1 Gb/s ports. You configure a traffic policy for one lane or multiple lanes. All members of the lane can use this policy. A policer requires at least one configured lane to function. You must configure a policer on a lane for a lane port to use it. You can configure up to 450 policies (policers) for each lane.

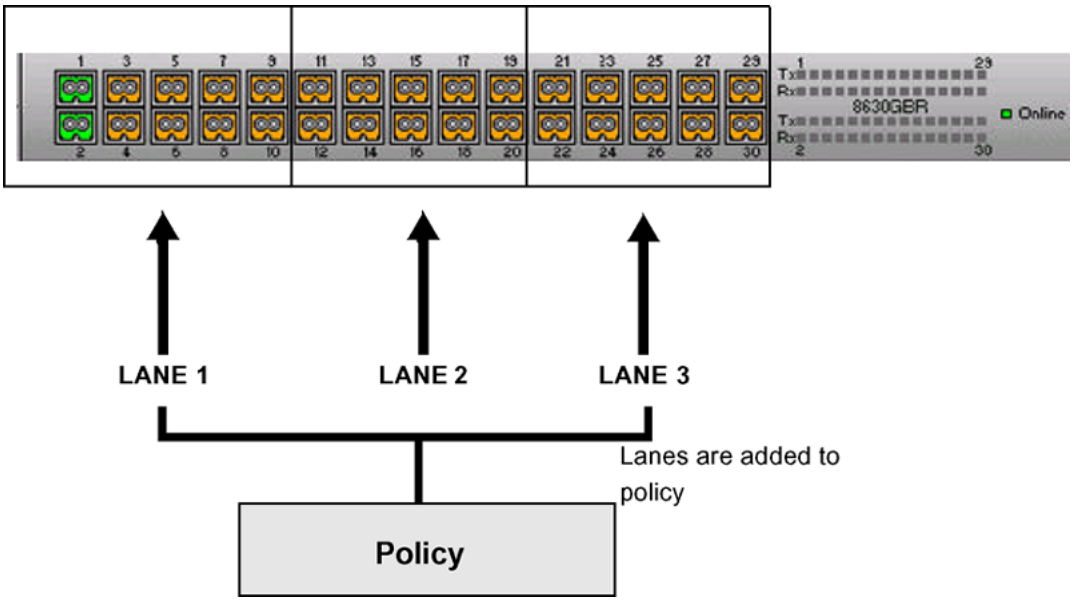


Figure 23: 8630GBR lanes

For more information about modules and lanes, see Table 2: Modules and lanes on page 34.

Policies and access control entries

You must bind a policy with a filter (access control entry—ACE). The filter classifies the packet from the input stream and applies the appropriate traffic policy based on the flow classification criteria configured in the filter. The following figure shows the building blocks for traffic policing.

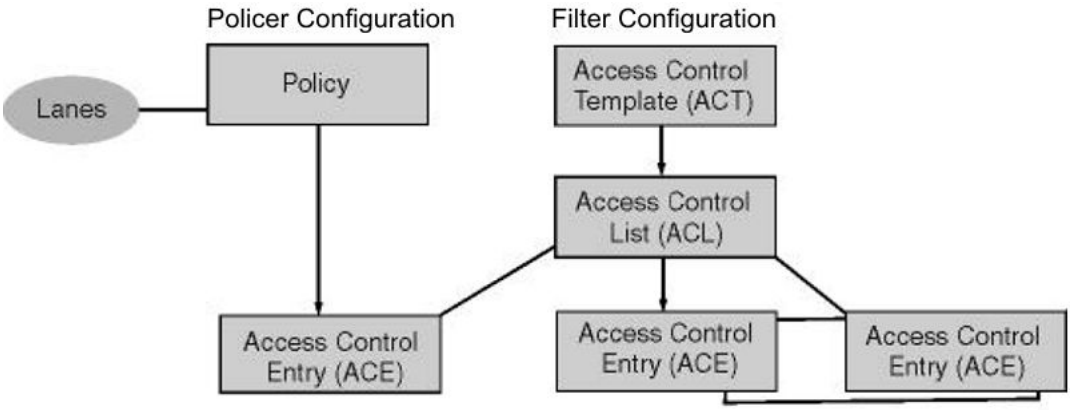


Figure 24: QoS traffic policing configuration building blocks

Policy-based policing actions

The following figure depicts policing actions. Packet coloring and drop actions depend on the peak and service rates. The policer drops packets transmitted greater than the configured peak rate; the policer recolors packets transmitted greater than the committed service rate.

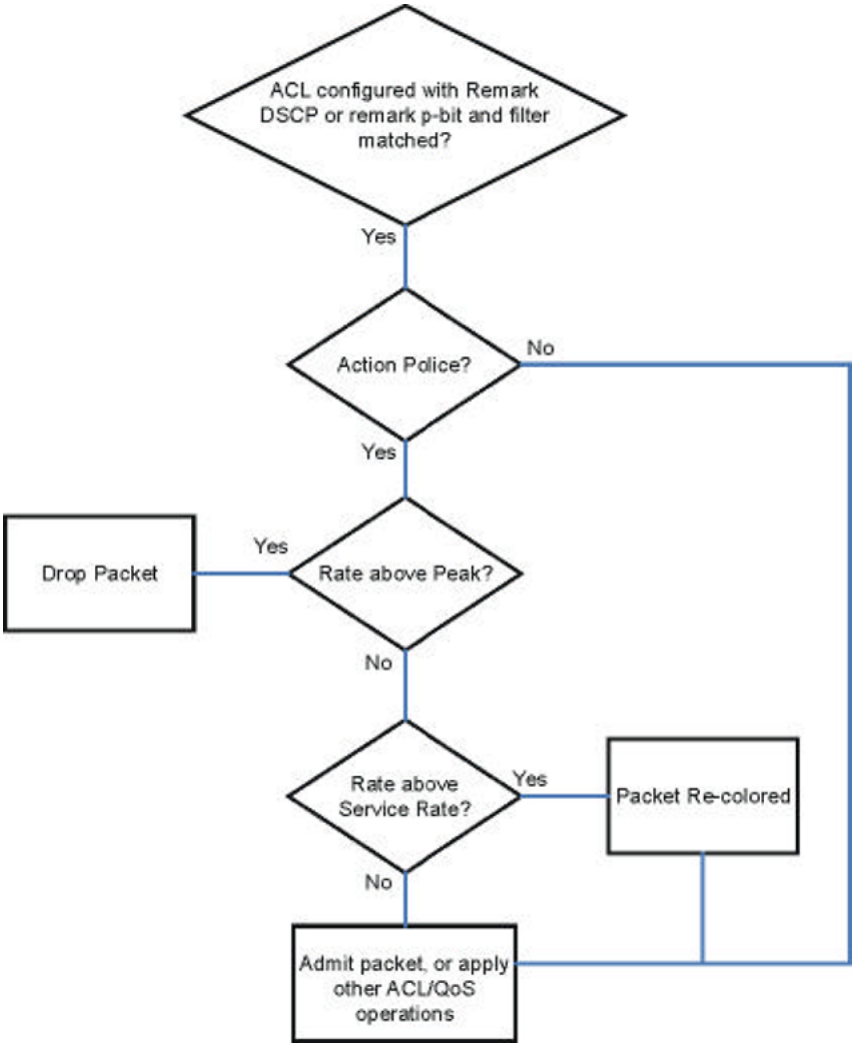


Figure 25: Policing actions

Port-based traffic policing

To provide QoS functionality at the MAC layer, RS modules and 8800 modules support a port-based policer. Port-based policing applies before the traffic reaches the network processor. You can use both policy-based policers and port-based policers at the same time.

Port-based policing rate limits aggregate port traffic. For example, if the system includes a 10 Gb/s link, but the rest of the system cannot handle 10 Gb/s traffic, you can use a port-based policer to rate limit to 5 Gb/s. The policer drops all traffic above 5 Gb/s.

Queue-based traffic shaping

Queue-based shapers are sets of egress queues. Each port can have only one queue-based shaper. A queue-based shaper shapes all outgoing traffic to the configured rate for that queue.

Shapers delay some or all packets in a traffic stream to bring the stream into compliance with a traffic profile. Shaping limits the output bandwidth to meet the downstream requirement, which eliminates bottlenecks in topologies with data rate mismatches.

Shapers apply at egress after the packet traverses ingress filters or policers.

For egress queue sets, you can configure a minimum and a maximum rate.

Egress queue set minimum rate

You can configure a minimum rate for balanced or low-priority queues. The minimum rate is a promise to allocate that minimum bandwidth percentage to the queue. If the output port is not congested and no more packets to service exist in priority queues, each balanced or low-priority queue can use the available bandwidth up to line rate or the configured maximum rate. The minimum rate does not apply to high- and low-priority queues.

Egress queue set maximum rate

You can configure a maximum rate for queues in balanced, low-priority and high-priority groups. The maximum rate limits the transmission of data higher than the configured rate. Traffic that exceeds the max-rate limit either buffers for the next time interval or is dropped if the buffer is full.

Traffic shaping statistics

Every elementary egress queue uses two hardware counters. The counters are total pages and dropped pages.

Statistical precision makes it difficult to compare actual queue output because statistics count pages. The first page is 144 bytes, all subsequent pages are 512 bytes. Packets of less than 144 (or 148, counting the packet header extension) bytes appear as one page. Packets of sizes greater than 144 bytes display a number of pages greater than the number of frames.

A packet header extension (PHE) is used when a packet originates from another R or RS module.

For more information about the relationship between packet size and memory pages used for egress queuing, see Egress queues and pages on page 349.

Comments? infodev@avaya.com

Port-based shaping

The port-based shaper rate limits the output traffic to the configured value for each port. By default, port-based shaping is disabled. The Ethernet Routing Switch 8800/8600 supports a minimum shaper rate of 1 Mb/s and a maximum of 10 Gb/s. The switch drops offending traffic.

For configuration instructions, see Configuring port-based shaping on page 91 (Enterprise Device Manager), Configuring the port-based shaper on page 164 (CLI), and Configuring the port-based shaper on page 239 (ACLI).

Broadcast and multicast traffic bandwidth limiters

The Ethernet Routing Switch 8800/8600 supports bandwidth limiters for ingress broadcast and multicast traffic. The modules drop traffic that violates the bandwidth limit.

For configuration instructions, see Configuring broadcast and multicast bandwidth limiting on page 163 (CLI) and Configuring broadcast and multicast bandwidth limiting on page 237 (ACLI).

QoS and MPLS

MPLS does not define new QoS architectures; MPLS QoS uses the DiffServ architecture defined for IP QoS.

IP DiffServ and MPLS DiffServ are similar in the following respects:

- both use classification, marking, policing, and shaping at the network edge
- both use buffer management and packet scheduling mechanisms to implement EF, AF, and Best-effort (BE) PHB

MPLS QoS differs from IP DiffServ because the DSCP parameter is not directly visible to MPLS Label Switch Routers (LSR), which forward based on the EXP parameter. Make QoS information visible to LSRs by using the EXP parameter. The Avaya Ethernet Routing Switch 8800/8600 uses ingress EXP bit to internal QoS and egress QoS to EXP bit mappings. The EXP bits map directly to the internal QoS level. Mappings take effect only on MPLS-enabled interfaces, and the switch trusts all MPLS interfaces.

The MPLS EXP bits in the label stack carry the packet QoS level between routers. On ingress, the classification stage derives the PHB from the EXP parameter in the top label stack entry. On egress, the PHB maps to an EXP value. The router marks the EXP in the top label stack entry of the packet before the packet enters a queue for transmission.

On the Avaya Ethernet Routing Switch 8800/8600, you globally define EXP to PHB profiles and PHB to EXP profiles (mappings) for the router.

The Ethernet Routing Switch supports setting EXP bits for both tunnel and service labels based on either 802.1p or DSCP markings.

Only MPLS-enabled interfaces trust MPLS EXP bits . If a port on which you disable MPLS receives an MPLS frame to bridge, it does not trust the EXP markings. If an MPLS edge switch receives a standard IP packetto go out on an MPLS interface, the switch can mark the EXP bits. In this case, the internal QoS-to-EXP egress mappings configure the EXP bits of the packet.

For more information about MPLS, see *Avaya Ethernet Routing Switch 8800/8600 Configuration — MPLS Services,* (NN46205-519). You can view or configure EXP mappings using the CLI, ACLI, or Enterprise Device Manager.

QoS and VoIP

Voice over Internet Protocol (VoIP) traffic requires low latency and jitter. To ensure the switch handles VoIP traffic appropriately, configure proper QoS.

When you use the Ethernet Routing Switch 8800/8600 as a core router, to treat VoIP traffic appropriately, configure ports as core ports (this is the default port setting). In this case, the switch trusts QoS markings applied to VoIP traffic and does not re-mark QoS settings. However, if this configuration is not sufficient, you can also apply filters, route policies, or re-mark traffic.

When you use the Ethernet Routing Switch 8800/8600 as an edge router (access port, or untrusted), you must pay attention to how the switch marks VoIP traffic. Because the Ethernet Routing Switch 8800/8600 does not support Power over Ethernet (PoE), and the switch generally operates in the network core, VoIP traffic is not a concern. If you use the Ethernet Routing Switch 8800/8600 as an edge device and you want to apply QoS to VoIP traffic, you can configure a specific VLAN (for example, a Voice VLAN) to apply a QoS level to VoIP traffic. In this case, Avaya recommends that you assign the VLAN default QoS level to 6 (Premium).

For Release 5.0, the Ethernet Routing Switch 8800/8600 supports a security mechanism called NSNA. NSNA supports the use of special VoIP VLANs; for more information, see *Avaya Ethernet Routing Switch 8800/8600 Security,* (NN46205-601).

Automatic QoS

The Avaya Automatic QoS feature allows Avaya data products to better support Avaya Converged Voice deployments (VoIP) by automatically recognizing the DSCP values that

Avaya Voice applications use, and associating these DSCP values with the proper egress queues. Without Avaya Automatic QoS support, you need to manually configure the DSCP values on the Ethernet Routing Switch and map them to the appropriate queues. With Avaya Automatic QoS enabled, manual DSCP-to-queue mapping is not required.

The following table shows various traffic types mapped to the standard DSCP values, the Avaya Automatic QoS DSCP values, and their associated queues.

Table 12: Avaya Automatic QoS DSCP Values

Traffic type	Standard DSCP value	Old queue value	Avaya Automatic QoS DSCP value (hex/decimal)	New queue value
VoIP Data (Premium)	0x2E (46) EF	6	0x2F (47)	6
VoIP Signaling (Platinum)	0x28 (40) CS5	5	0x29 (41)	5
Video (Platinum)	0x22 (34) AF41	5	0x23 (35)	5
Streaming (Gold)	0x1A (26) SF31	4	0x1B (27)	4

For proper functioning of the feature, you must enable Avaya Automatic QoS on the Ethernet Routing Switch and on the associated Avaya Voice application.

Avaya Auto QoS is supported on the following Avaya voice and data products:

- Ethernet Routing Switch 4500
    - Release 5.2
    - Edge with Avaya Automatic QoS mixed or pure mode
- Ethernet Routing Switch 5000
    - Release 6.0
    - Edge with Avaya Automatic QoS mixed or pure mode
- Ethernet Routing Switch 8300
    - Release 4.2
    - Avaya Automatic QoS core only
- Ethernet Routing Switch 8800/8600
    - Release 5.1
    - Avaya Automatic QoS core only
- CS 1000
    - Avaya Automatic QoS supported in Element Manager
    - Release 5.5

    - Patch MPLR26485 is required
- CS 2100
    - SE10
    - Edge with Avaya Automatic QoS supported in Element Manager
- BCM 50, SRG 50, and BCM450
    - BCM50/SRG50 requires a minimum of Release 3.0 software with Smart Update BCM050.R300.SU.System-115 or later
    - BCM450 requires a minimum of Release 1.0 software with Smart Update BCM450.R100.SU.System-003 or later

For more information on configuration of these products, see *Avaya Automatic QoS Technical Configuration Guide for the ERS 4500, 5000, BCM 50, 450, CS1000, CS2100 and SRG 50, NN48500-576*.

You can configure the Ethernet Routing Switch 8800/8600 as a core switch only. Avaya Automatic QoS on the Ethernet Routing Switch 8800/8600 has no edge configuration. Presently, when used as a core switch for Avaya Automatic QoS with either the Ethernet Routing Switch 4500 or Ethernet Routing Switch 5000 as an edge switch, only Avaya Automatic QoS mixed mode is supported on the edge switch.

To configure Avaya Automatic QoS operation, configure the Avaya Voice Application with the proper Avaya Automatic QoS setting, enable DiffServ on the connected ingress port on the Ethernet Routing Switch, and then configure the port as a trusted core port. (The default operational value for Avaya Ethernet Routing Switch 8800/8600 ports is core.)

802.1Q tagged packets

The Ethernet Routing Switch 8800/8600 I/O modules. Modules support an 802.1p-bit-override feature for tagged packets that allows the modules to ignore the 802.1p-bit and classify traffic based on the DSCP values instead.

Chapter 4: Traffic filtering fundamentals

Traffic filtering on the Avaya Ethernet Routing Switch 8800/8600 is a mechanism to manage traffic by defining filtering conditions and associating these conditions with specific actions. Filtering blocks unwanted traffic and prioritizes other traffic, which efficiently manages bandwidth and protects your network.

Overview

Using traffic filters, you can reduce network congestion and control access to network resources by blocking, forwarding, or prioritizing specified traffic on an interface.

The Avaya Ethernet Routing Switch 8800/8600 can use traffic filtering for many purposes. Filtering can provide security and can help ensure that all traffic is treated according the Class of Service (COS) required by the application. The Ethernet Routing Switch can drop low-priority traffic under congestion, police incoming traffic, and mark or drop nonconforming traffic. The traffic class (internal to the switch), drop precedence, DSCP, EXP, and 802.1p bit markings define the COS. The switch supports DiffServ marking and re-marking using filters.

You need not use filters to provide QoS. Filters can override QoS packet operations.

On I/O modules, each port supports 8 or 64 hardware egress queues, with control traffic (for example, spanning tree) assigned to the highest priority queue. You can implement filters by using access control templates (ACT), access control entries (ACE), and access control lists (ACL).

Traffic filters for R, RS, and 8800 series modules

The Avaya Ethernet Routing Switch 8800/8600 utilizes filtering implementation that uses R, RS and 8800 modules and ACLs to support ingress and egress Layer 2 through Layer 7 filtering.

The Ethernet Routing Switch 8800/8600 software provides some configuration guidelines. For example, when you add virtual local area networks (VLAN) to an ACL, a message indicates the filters apply only to the R, RS, or 8800 module port members of that VLAN. When you add ports to an ACL, the switch ensures that the port belongs to an R, RS, or 8800 module.

In R, RS, or 8800 module traffic filtering, a filtering rule (an ACE) defines a pattern found in a packet and the desired behavior for that packet. An ACL is a group of ACE filtering rules associated with a logical interface at ingress or egress.

As each packet enters an interface with an ACL, the interface scans matching ACEs for that packet and applies the actions of those ACEs according to precedence.

Filters operate in the same manner for R modules and RS and 8800 modules. The only difference between R module and RS and 8800 module filter operations is port mirroring. See RS and 8800 modules and port mirroring on page 81 and R modules and port mirroring on page 81.

Deep packet pattern match filters

The Avaya Ethernet Routing Switch 8800/8600 offers deep packet inspection to detect and block attacks that directly target applications and data that use the packet payload. Using deep packet filters, the switch can identify the traffic content and completely block, rate limit, or shape it, and can apply any filter rule to the packet. Deep packet pattern match filters rely on ACL-based filters that operate based on matches of up to 80 bytes deep in the packet. You can configure these filters at the bit level.

R, RS, and 8800 series module filters and packet layer traversal

The Ethernet Routing Switch 8800/8600 offers powerful and easy-to-use filters. R, RS, and 8800 module-based filters apply to packets regardless of the OSI layer they traverse. Generally, the ACLs of other companies apply at routing boundaries only; if a packet does traverse a Layer 3 boundary, the ACL does not apply. As a result, to provide filtering for each layer, other companies must either apply Layer 2 ACLs with Layer 3 ACLs, or use private VLANs. Either option makes filter configurations crowded and difficult to debug. Avaya R, RS, and 8800 module filters apply to the packet regardless of the Layer N operation that applies to the packet (switched or routed).

Access control templates

An ACT defines the selection of match fields for each ACL. Filters require an ACT. Before you add an ACE to an ACL, you must first associate the ACL with an existing ACT.

Access control templates navigation

- ACT attributes on page 67
- ACT patterns for offset filtering on page 67
- Predefined ACTs on page 70
- ACT configuration guidelines on page 72

ACT attributes

An ACT defines a set of match fields, or attributes, for an ACL. The Avaya Ethernet Routing Switch 8800/8600 supports the following attributes:

- ARP operation—If the packet is an ARP packet, this attribute matches the ARP operation (ARP request or ARP response). The supported operators for this attribute are none or operation.
- Ethernet—Specifies one of the following Ethernet attributes: none, source MAC, destination MAC, etherType, port, VLAN, or VLAN Tag Priority.
- IP—Specifies one or more of the following IP attributes: none, source IP, destination IP, IP fragmentation flag, IP options, IP protocol type, or DSCP.
- IPv6—Specifies one or more of the following IPv6 attributes: none, source IPv6, destination IPv6, or nextHdr.
- Protocol—Specifies one or more of the following protocol attributes: none, TCP source port, UDP source port, TCP destination port, UDP destination port, TCP flags, or ICMP message type.

ACT patterns for offset filtering

An ACT can contain pattern parameters used for offset filtering. To use an ACT pattern, select the base; this specifies where to start the offset filter. Then select, in bits, the offset bit position and the offset length.

You can configure up to three ACT pattern attributes for each ACL. If you require more than three ACT pattern attributes, combine a port and a VLAN ACL type to support up to six ACT pattern attributes.

Although the pattern length for one ACT pattern can be up to 56 bits, combine two or three ACT patterns to filter a pattern length of greater than 56 bits. For example, you can combine two ACT patterns to filter a pattern of up to 112 bits in length.

The following table shows the available pattern options.

Table 13: ACT pattern options

Field	Description	
Base	A user-defined header for the ACEs of the ACL.	
	Item	Description
	etherBegin	Beginning of the Ethernet packet.
	macDstBegin	Beginning of the MAC destination field in the Ethernet packet header.
	macSrcBegin	Beginning of the source MAC field in the Ethernet packet header.
	ethTypeLenBegin	Beginning of the type and length field in the Ethernet packet header.
	arpBegin	Beginning of the hardware address type field in the ARP packet.
	ipHdrBegin	Beginning of the IP packet header (version field).
	ipOptionsBegin	Beginning of the IP options field in the IP header. This item is normally after the IP destination address. If the packet does not include IP options (the header length is equal to 5), the filter does not apply. The filter applies only if the header length is greater than 5.
	ipPayloadBegin	Located after the IP destination address. If the packet includes IP options, it is after the IP options field, plus padding.
	ipTosBegin	Beginning of the TOS byte in the IP header.
	ipProtoBegin	Beginning of the IP type in the IP header (starting with the ninth byte).
	ipSrcBegin	Beginning of the source IP field in the IP header.
	ipDstBegin	Beginning of the destination IP field in the IP header.
	tcpBegin	Beginning of the TCP packet.
	tcpSrcportBegin	Beginning of the source port field in the TCP header.
	tcpDstportBegin	Beginning of the destination port field in the TCP header.
	tcpFlagsEnd	End of the TCP flags field in the TCP header (beginning of the window field).
	udpBegin	Beginning of the UDP packet.

Comments? infodev@avaya.com

Field	Description	
	udpSrcportBegin	Beginning of the source port field in the UDP header.
	udpDstportBegin	Beginning of the destination port field in the UDP header.
	etherEnd	End of Ethernet header.
	ipHdrEnd	End of IP header (after IP options and padding).
	icmpMsgBegin	Beginning of the ICMP header (type field in the ICMP message header).
	tcpEnd	End of TCP header.
	udpEnd	End of UDP header.
	ipv6HdrBegin	Beginning of the IPv6 packet header (version field).
Offset	Configures the offset (in bits) to the beginning offset of the user-defined field with the selected header option as a base. Valid values are 0–76800.	
Length	Configures the number of bits to extract from the beginning of the offset. Valid values are 1–56.	

ACT pattern examples

The following table provides examples that use ACT patterns. To view the entire configuration example for these patterns, see *Filters and QoS for ERS 8800/8600 R-Series Modules Technical Configuration Guide, NN48500-541*.

Table 14: ACT pattern examples

Function	Configuration

Use a pattern to prevent SQLslam. Activity of this worm is readily identifiable on a network by the presence of 376-byte UDP packets.	Start at the beginning of the IP TOS field
	The pattern begins 216 bits (27 bytes, data field) from the beginning of the IP TOS field
	The pattern length is 48 bits (6 bytes)
	Use the ACT pattern in an ACE, add the offset pattern of 040101010101
	`config filter act 1 pattern SQLslam add ip-tos-begin 216 48`
	`config filter acl 4 ace 1 advanced custom-filter1 SQLslam eq 040101010101`

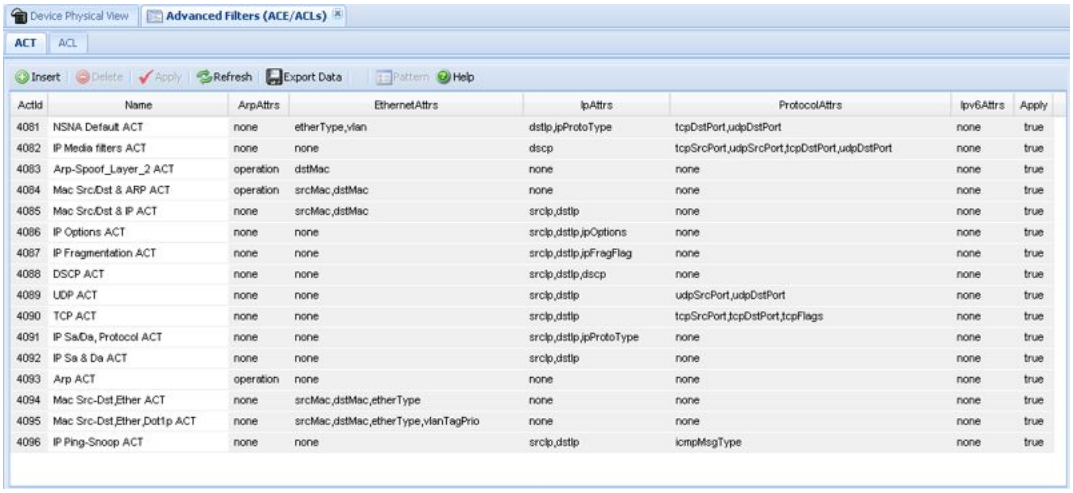
Use a pattern to prevent Nachia attacks.	Start at the beginning of the IP TOS field The pattern begins 224 bits (28 bytes) from the beginning of the IP TOS field The pattern length is 24 bits (3 bytes)

Function	Configuration

	Use the ACT pattern in an ACE, add the offset pattern of aaaaaa
	`config filter act 1 pattern Nachia add ip-tos-begin 224 24`
	`config filter acl 4 ace 2 advanced custom-filter2 Nachia eq aaaaaa`

Predefined ACTs

You can configure custom ACTs or you can choose from a list of predefined ACTs. The following figure shows the Ethernet Routing Switch 8800/8600 predefined ACTs viewed with Enterprise Device Manager. The information shown includes the ARP, Ethernet, Protocol, IPv6, and IP attributes associated with each ACT.



ActId	Name	ArpAttrs	EthernetAttrs	IpAttrs	ProtocolAttrs	Ipv6Attrs	Apply
4081	NSNA Default ACT	none	etherType,vlan	dstIp,ipProtoType	tcpDstPort,udpDstPort	none	true
4082	IP Media filters ACT	none	none	dscp	tcpSrcPort,udpSrcPort,tcpDstPort,udpDstPort	none	true
4083	Arp-Spoof_Layer_2 ACT	operation	dstMac	none	none	none	true
4084	Mac Src/Dst & ARP ACT	operation	srcMac,dstMac	none	none	none	true
4085	Mac Src/Dst & IP ACT	none	srcMac,dstMac	srcIp,dstIp	none	none	true
4086	IP Options ACT	none	none	srcIp,dstIp,ipOptions	none	none	true
4087	IP Fragmentation ACT	none	none	srcIp,dstIp,ipFragFlag	none	none	true
4088	DSCP ACT	none	none	srcIp,dstIp,dscp	none	none	true
4089	UDP ACT	none	none	srcIp,dstIp	udpSrcPort,udpDstPort	none	true
4090	TCP ACT	none	none	srcIp,dstIp	tcpSrcPort,tcpDstPort,tcpFlags	none	true
4091	IP Sa/Da, Protocol ACT	none	none	srcIp,dstIp,ipProtoType	none	none	true
4092	IP Sa & Da ACT	none	none	srcIp,dstIp	none	none	true
4093	Arp ACT	operation	none	none	none	none	true
4094	Mac Src-Dst,Ether ACT	none	srcMac,dstMac,etherType	none	none	none	true
4095	Mac Src-Dst,Ether,Dot1p ACT	none	srcMac,dstMac,etherType,vlanTagPrio	none	none	none	true
4096	IP Ping-Snoop ACT	none	none	srcIp,dstIp	icmpMsgType	none	true

Figure 26: Predefined ACT list

Use a predefined ACT whenever possible. You can create your own ACTs; however, ensure that you include the minimum required parameters on which to filter. The more attributes on which you choose to filter, the longer it takes the Ethernet Routing Switch 8800/8600 to process incoming data.

The following table describes the action of each predefined ACT.

Table 15: Predefined ACT actions

ACT ID	ACT name	Description
4080	VPS Default ACT	Filters on packets used specifically by the VPS application.

Comments? infodev@avaya.com

ACT ID	ACT name	Description
4081	SNA Default ACT	etherType, vlan, DestIp, IpProtoType, tcpDstPort, and udpDestPort. Used with Avaya Secure Network Access.
4082	IP Media filters ACT	Filters on Protocol attributes tcpSrcPort, udpSrcPort, tcpDstPort, and udpDstPort.
4083	Arp-Spoof_Layer_2 ACT	Filters on packets with ARP information, and on the Ethernet attribute dstMac. PreventsARP spoofing.
4084	Mac Src/Dst & ARP ACT	Filters on packets with ARP information, and on the Ethernet attributes dstMac and srcMac.
4085	Mac Src/Dst & IP ACT	Filters on the Ethernet attributes dstMac and srcMac, and on the IP attributes dstIp and ScrIp.
4086	IP Options ACT	Filters on the IP attributes srcIp, dstIp, and ipOptions.
4087	IP Fragmentation ACT	Filters on the IP attributes srcIp, dstIp, and ipFragFlag.
4088	DSCP ACT	Filters on the IP attributes srcIp, dstIp, and dscp.
4089	UDP ACT	Filters on the IP attributes srcIp, dstIp; and on the Protocol attributes udpSrcPort, udpDstPort.
4090	TCP ACT	Filters on the IP attributes srcIp, dstIp; and on the Protocol attributes tcpSrcPort, tcpDstPort, tcpFlags.
4091	IP Sa/Da, Protocol ACT	Filters on the IP attributes srcIp, dstIp, and ipProtoType.
4092	IP Sa and Da ACT	Filters on the IP attributes srcIp, and dstIp.
4093	Arp ACT	Filters on packets with ARP information.
4094	Mac Src-Dst,Ether ACT	Filters on packets with Ethernet attributes srcMac, dstMac, and etherType.
4095	Mac Src-Dst,Ether,Dot1p ACT	Filters on packets with Ethernet attributes srcMac, dstMac, etherType, and vlanTagPrio.
4096	IP Ping-Snoop ACT	Filters on the IP attributes: srcIp, dstIp and the protocol attribute icmpMsgType. Used with the Ping Snoop feature. For more information about Ping Snoop, see *Avaya Ethernet Routing Switch 8800/8600 Troubleshooting,* (NN46205-703).

ACT configuration guidelines

ACTs define the attributes and pattern information used in the ACEs of an ACL. One or more ACLs can use an ACT. After you create the ACL using an ACT, you cannot modify the ACT.

When you configure a new ACT, choose only the attributes you plan to use when you configure the ACEs. For each additional attribute you include in an ACT, the switch must perform an additional lookup. To enhance performance, keep the number of ACT attributes as small as possible. For example, if you plan to filter on source and destination IP addresses and DSCP, select only these IP attributes. The number of ACEs within an ACL does not affect performance.

Important:
Be careful when you configure an ACT, because the CLI allows you to configure mutually-exclusive ACT attributes.

The following list describes ACT guidelines:

- For pattern matching filters, the switch supports three patterns for each ACT.
- After you configure the ACT, you must activate it (Apply = true). After you activate the ACT, you cannot modify it; you can only delete it.
- You can delete an ACT only when no ACLs use that ACT.
- The switch supports 4000 ACTs and 4000 ACLs.
- The switch reserves ACT and ACL IDs 4001 to 4096 for system-defined ACTs and ACLs. You can use these ACTs and ACLs, but you cannot modify them.

An ACT with an IPv6 attribute has a single ACL of type IPv6.

An ACT with only Ethernet attributes can include up to two ACLs. You can have only one IPv4 and one IPv6 ACL.

Access control lists

The Avaya Ethernet Routing Switch 8800/8600 I/O modules use ACLs for filtering. An ACL comprises an ordered list of ACEs (filter rules). The ACEs provide specific actions, such as dropping packets within a specified IP range, or a specific UDP port or port range. For more details, see Access control entries on page 75. When an ingress or egress packet meets the match criteria specified in one or more ACEs within an ACL, the corresponding action occurs.

An ACL can contain multiple ACEs, which the ACL uses to control multiple flows. A packet can match attributes in more than one ACE. The actions that apply to the packet are the nonconflicting actions of the matching ACEs. The ACE priority resolves which action, among conflicting actions, applies.

The default action applies when no ACEs match a packet, while global actions apply to all ACEs that match a packet. The default action is permit, and the default global action is none (no action). You can modify the default and global actions at any time.

ACL global actions include

- none
- mirror
- count
- mirror-count
- ipfix
- mirror-ipfix
- count-ipfix
- mirror-count-ipfix

In addition to the system-defined attributes, you can choose up to three patterns to match against. You can match anywhere in the packet on the ingress side, and anywhere within the first 144 bytes on the egress side. You can combine the three patterns, up to 7 bytes each, to form a 21-byte pattern match.

Four types of ACLs exist:

- Ingress port (inPort)
- Ingress VLAN (inVLAN)

  When you use type inVlan, ports that you define under the ACL apply the filter to ingress packets on those ports.
- Egress port (outPort)
- Egress VLAN (outVLAN)

  When you use type outVlan, ports that you define under the ACL apply the filter to egress packets on those ports.

The ingress and egress VLAN ACLs apply to all the active port members of that VLAN. By default, you create an ACL in the enabled state.

The Avaya Ethernet Routing Switch 8800/8600 supports both port-based and VLAN-based ACLs. Depending on the configuration, you can apply the actions of both ACLs to a packet. In such cases, the port-based ACL actions have priority and apply first.

The Ethernet Routing Switch 8800/8600 supports two default (or predefined) ACLs: the IP Media Filters ACL and the IP Ping-Snoop ACL. These operate with ACTs of the same name.

The following figure shows the relationships between ACTs, ACEs, and ACLs.

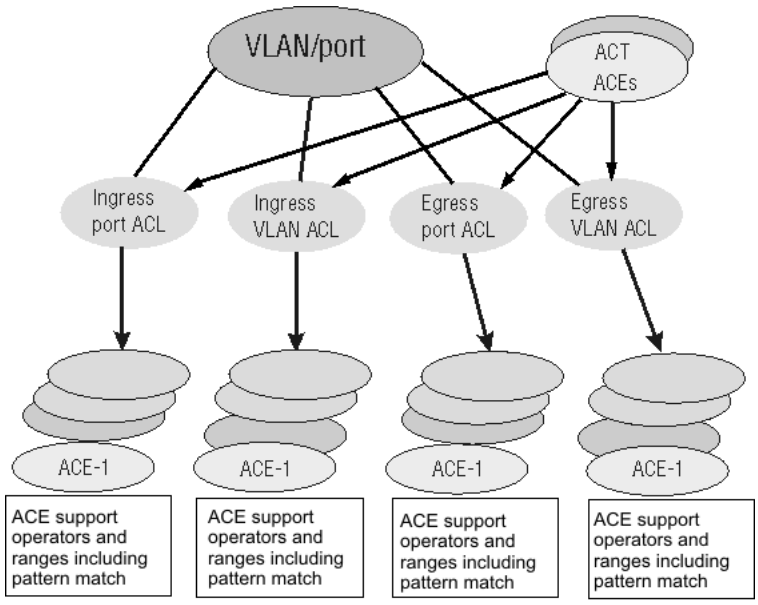


Figure 27: ACT, ACE, and ACL relationships

ACL priority

You can configure both port-based ACLs and VLAN-based ACLs. Avaya recommends that you apply only one type of ACL to a packet; however, sometimes the actions of both port-based and VLAN-based ACLs must apply to a packet. In this case, apply the port-based ACL actions first. Apply VLAN-based ACL actions only if the mode (permit or deny) is the same as for the port-based ACL and if the VLAN-based ACL ACE actions do not overlap with the port-based ACL actions.

ACL priority examples

The following examples demonstrate the resulting action based on the configured mode and actions:

Example 1

Port and VLAN-based ACL configuration:

- Port-based ACL—mode permit, any action
- VLAN-based ACL—mode deny, any action

The actions of the port-based ACL apply.

Example 2

Comments? infodev@avaya.com

Port and VLAN-based ACL configuration:

- Port-based ACL

  ACE 1: mode permit, action police
- VLAN-based ACL

  ACE 1: mode permit, action police

  ACE 2: mode permit, action remark-dscp

The actions of the port-based ACL and the actions of ACE 2 of the VLAN-based ACL apply.

Example 3

Port and VLAN-based ACL configuration:

- Port-based ACL

  ACE 1: mode permit, action police
- VLAN-based ACL

  ACE 1: mode permit, actions police, remark-dscp

The actions of the port-based ACL apply.

Access control entries

Access control entries (ACE) provide the match criteria and rules for ACL-based filters.

Access control entries navigation

- ACE overview on page 75
- ACE actions on page 76
- ACE priority on page 77
- Common ACE uses and configurations on page 78
- Example: ACE TCP Established flag filter on page 79

ACE overview

An ACE is one filter rule that makes up an ACL. A filter rule is a statement that defines a pattern (found in a packet) and the desired behavior for packets that carry the pattern. When the packets match an ACE rule, the specified action occurs.

An ACE affects matching packets on all interfaces associated with the contained ACL. As each packet enters an interface with an associated ACL, the interface scans the list for a pattern

that matches the incoming packet. A behavior rule associated with the pattern determines packet treatment.

If multiple ACEs in an ACL match a packet, you can choose a preferred ACE by assigning precedence to the rule. The switch determines precedence by the ACE ID: the lower the ID number, the higher the precedence. Behavior for a packet that meets the criteria specified by more than one rule is derived from the highest precedence rule to ensure deterministic behavior.

If you do not specify a value for an ACT attribute in the ACE, that attribute value is treated as a wildcard. You can configure a maximum of 1000 ACEs for each port for ingress and egress. The system supports a maximum of 10 000 ACEs.

When you disable the ACL, the ACL state affects the administrative state of all ACEs within it.

Avaya Ethernet Routing Switch 8800/8600 I/O modules limit the memory for statistics counters. The system supports up to 1000 counters for ingress (depending on the overlapping attribute values) and an equal number for egress.

ACE actions

You must specify actions for ACEs. The following table shows a sample of ACL and ACE parameters and valid ingress and egress actions.

Table 16: Ingress and egress ACL and ACE parameters

Ingress (port or VLAN-based)		
Match criteria MAC, p-bits, VLAN tag, ARP, IP, DSCP, TCP, and UDP	Match pattern base, offset, and length	Action Permit, deny, redirect to next hop, redirect to next hop IPv6, redirect to MLT index, remark 802.1p, remark DSCP, police, send to eqress queue
Egress (port or VLAN-based)		
Match criteria MAC, p-bits, VLAN tag, ARP, IP, DSCP, TCP, and UDP	Match pattern base, offset, and length	Action permit and deny
Priority		
Based on ID (port-based ACL before VLAN-based ACL)		

If a packet matches multiple ACEs, the Avaya Ethernet Routing Switch 8800/8600 applies the noncontradicting actions of all ACEs according to precedence (ACE ID). If you specify a stop-on-match flag, the switch stops at that ACE.

Comments? infodev@avaya.com

If the switch redirects a packet, it does not perform regular packet processing for the packet. The mirroring configuration, policer configuration, and egress queue ID configuration must occur outside the context of filtering.

ACE priority

If a packet matches multiple ACEs in an ACL, the actions of the highest priority ACE apply. The actions of the remaining ACEs apply only if the mode is the same as the highest priority ACE, and if the actions do not overlap with the highest priority ACE.

ACE priority examples

The following examples demonstrate the action taken based on the configured mode and actions:

Example 1

ACE 1 and 2 configuration:

- ACE 1—mode permit, actions police
- ACE 2—mode deny, actions mirror

The actions of only ACE 1 apply.

Example 2

ACE 1 and 2 configuration:

- ACE 1—mode deny, action mirror
- ACE 2—mode permit, action police

The actions of only ACE 1 apply.

Example 3

ACE 1, 2, 3, and 4 configuration:

- ACE 1—mode permit, action police
- ACE 2—mode deny, action mirror
- ACE 3—mode permit, actions police, mirror
- ACE 4—mode permit, action remark-dscp

The actions of ACE 1 and ACE 4 apply.

Example 4

ACE 1, 2, 3, and 4 configuration:

- ACE 1—mode permit, action police
- ACE 2—mode deny, action mirror

- ACE 3—mode permit, actions mirror, stop-on-match
- ACE 4—mode permit, actions remark-dscp

The actions of ACE 1 and ACE 3 apply.

Common ACE uses and configurations

The following table describes configurations you can use to perform common actions.

Table 17: Common ACE uses and configurations

Function	ACE configuration

Permit a specific host network access	Use action permit
	Configure the source IP address as the host IP address
	`filter acl 1 ace 5 create name "Permit_access_to_1.2.3.4"`
	`filter acl 1 ace 5 action permit stop-on-match true`
	`filter acl 1 ace 5 ip src-ip eq 1.2.3.4`
	`filter acl 1 ace 5 enable`

Deny a specific host network access	Use action deny
	Configure the source IP address as the host IP address
	`filter acl 1 ace 5 create name "Deny_access_to_1.2.3.4"`
	`filter acl 1 ace 5 action deny stop-on-match true`
	`filter acl 1 ace 5 ip src-ip eq 1.2.3.4`
	`filter acl 1 ace 5 enable`

Permit a specific range of hosts network access	• use action permit
	• configure the source IP address as the range of host IP addresses
	`filter acl 1 ace 5 create name "Permit_access_to_1.2.3.4-5.6.7.8"`
	`filter acl 1 ace 5 action permit stop-on-match true`
	`filter acl 1 ace 5 ip src-ip eq 1.2.3.4-5.6.7.8`
	`filter acl 1 ace 5 enable`

Deny Telnet traffic	Use action deny Configure the protocol as TCP and the TCP destination port as 23 `filter acl 1 ace 5 create name "Deny_telnet"`

Comments? infodev@avaya.com

Function	ACE configuration

	`filter acl 1 ace 5 action deny stop-on-match true`
	`filter acl 1 ace 5 ip ip-protocol-type eq tcp`
	`filter acl 1 ace 5 protocol tcp-dst-port eq 23`
	`filter acl 1 ace 5 enable`

Allow only internal networks to initiate a TCP session	Use the Established filter. See Example: ACE TCP Established flag filter on page 79.

Deny FTP traffic	Use action deny
	Configure the protocol as TCP and the TCP destination port as 21
	`filter acl 1 ace 5 create name "Deny_ftp"`
	`filter acl 1 ace 5 action deny stop-on-match true`
	`filter acl 1 ace 5 ip ip-protocol-type eq tcp`
	`filter acl 1 ace 5 protocol tcp-dst-port eq 21`
	`filter acl 1 ace 5 enable`

Example: ACE TCP Established flag filter

The following ACE filter matches for the Established flag of TCP packets. This filter matches traffic after a TCP three-way handshake is complete. This usually occurs in the context of traffic between the Internet and servers.

The following Established flag filter matches and permits any packet with a protocol type of TCP and looks for the TCP flags Reset (RST) or Acknowledgement (ACK).

Example 1:

```
filter acl 1 ace 5 create name "ESTABLISHED"
filter acl 1 ace 5 action permit stop-on-match true
filter acl 1 ace 5 ip src-ip eq 1.6.172.0-1.6.172.255
filter acl 1 ace 5 ip ip-protocol-type eq tcp
filter acl 1 ace 5 protocol tcp-dst-port ge 1023
filter acl 1 ace 5 protocol tcp-flags match-any rst,ack
filter acl 1 ace 5 enable
```

Because most IP traffic uses port numbers less than 1023, any packet with a destination port less than 1023, or with an unset ACK or RST bit, is denied. Therefore, when a host attempts to initiate a TCP connection by sending the first TCP packet (without SYN or RST bit set) for a port number less than 1023, it is denied; the TCP session fails. The switch permits any internally initiated TCP sessions because they have ACK or RST bits set for returning packets, and they use port numbers greater than 1023.

Example 2:

```
filter acl 100 ace 10 create name "10_50_all_established"
filter acl 100 ace 10 action permit stop-on-match true
filter acl 100 ace 10 debug count enable
filter acl 100 ace 10 ip dst-ip eq 10.50.0.0-10.50.255.255
filter acl 100 ace 10 ip ip-protocol-type eq tcp,icmp
filter acl 100 ace 10 protocol tcp-src-port eq 21-22,80,443,3389
filter acl 100 ace 10 protocol tcp-flags match-any rst,ack
filter acl 100 ace 10 enable
```

Port mirroring, ACLs, and ACEs

Use port mirroring to monitor and analyze network traffic. Port mirroring supports both ingress (incoming traffic) and egress (outgoing traffic) port mirroring. When you enable mirroring, the switch forwards the mirrored (source) port ingress or egress packets normally, and sends a copy of the packets from the mirrored port to the mirroring (destination) port. You can observe and analyze packet traffic at the mirroring port by using a network analyzer.

You can configure two mirroring functions: ACL and ACE-based mirroring, and individual port diagnostic mirroring, for which you need not configure filters.

Configure an ACL or an ACE to perform the mirroring operation. To do so, you can configure the ACL global action to mirror, or you can configure the ACE debug action to mirror. If you use the global action, mirroring applies to all ACEs that match in an ACL.

You can use filters to reduce the amount of mirrored traffic. Apply an ACL to the mirrored port in the egress, ingress, or both directions. Filters forward traffic patterns that match the ACL or ACE with an action of permit to the destination and to the mirroring port. Filters do not forward traffic patterns that match an ACE with an action of drop (deny) to the destination, but traffic still reaches the mirroring port. If you enable a port or VLAN filter, that filter is the mirroring filter.

You can specify more than one mirroring destination by using multiple ACEs. Use each ACE to specify a different destination. The following table identifies the procedures to use to configure port mirroring.

Table 18: Port mirroring procedures

For information about	See
Configuring port mirroring using Enterprise Device Manager	Configuring an access control list on page 107 and Configuring ACEs on page 111
Configuring port mirroring using the CLI	Configuring global and default actions for an ACL on page 190 and Configuring ACE debug actions on page 202
Configuring port mirroring using the ACLI	Configuring global and default actions for an ACL on page 260 and Configuring ACE debug actions on page 273
Configuration examples	Mirroring using ACLs on page 223

For information about	See
Port mirroring and diagnostics	*Avaya Ethernet Routing Switch 8800/8600 Troubleshooting,* (NN46205-704)

R modules and port mirroring

R modules support two port mirroring modes: receive (Rx) (ingress, that is, inPort and inVLAN) and transmit (Tx) (egress, that is, outPort and outVLAN).

In Rx mode, when you configure the ACE Debug or ACL Global options to mirror, use the ACE to configure the mirroring destination port.

In Tx mode, when you configure the ACE Debug or ACL Global options to mirror, use the Diagnostics parameter to configure the mirroring destination. For example, in Enterprise Device Manager, choose Edit, Diagnostics, Port Mirrors tab to select the destination ports.

RS and 8800 modules and port mirroring

RS and 8800 modules offer enhanced port mirroring. Using RS and 8800 modules, you can specify a destination multilink trunking (MLT) group, a destination port or set of ports, or a destination VLAN.

RS and 8800 modules support rxFilter and txFilter modes, but operate different from R modules. As you do for R modules, you select the mode by configuring the inPort, outPort, inVLAN, and outVLAN ACL parameters. You can globally configure the mirroring action in an ACL, or for a specific ACE by using the ACE Debug actions. However, regardless of the ingress or egress mode, you configure the mirroring destination by using an ACE.

For more information about port mirroring, see *Avaya Ethernet Routing Switch 8800/8600 Troubleshooting,* (NN46205-703).

Traffic filter configuration

Traffic filtering is a mechanism that manages traffic by defining filtering conditions and associating these conditions with specific actions. Within a DiffServ network, use IP filtering to reassign QoS levels based on a range of filtering conditions.

The following steps summarize the filter configuration process:

1. Determine your desired match fields.
2. Use a predefined ACT that includes your desired match fields; otherwise, configure an ACT with your desired match fields.

3. Configure an ACL and associate it with the ACT.
4. Configure an ACE within the ACL.
5. Configure the desired precedence, traffic type, and action.

   You determine the traffic type when you create either an ingress or egress ACL.
6. Modify the fields for the ACE.

ACL, ACT, and ACE configuration guidelines

ACEs of type inVlan with an ACT that includes srcIp and with an ACL default action of deny require additional configuration to function properly. See Workaround for inVlan, srcIp ACL on page 351.

Alternatively, Avaya recommends that you create ACLs with a default action of permit and with an ACE mode of deny. For deny and permit ACLs or ACEs, the default action and the mode must be opposite for the ACE (filter) to have meaning.

When you configure filters, keep the following scaling limits in mind.

Table 19: ACT, ACE, ACL scaling

Parameter	Maximum number
ACLs for each switch	4000
ACEs for each switch	4000
ACEs for each ACL	500
ACEs for each port	2000 • 500 inPort • 500 inVLAN • 500 outPort • 500 outVLAN

Secure Network Access

Secure Network Access (SNA) is an Avaya network access control solution where the edge devices (for example, the Ethernet Routing Switch 8800/8600) work in coordination with access controllers and policy servers to enforce security policy compliance on all endpoints (for example, PCs, laptops, IP phones) that access network computing resources. SNA

provides network access only to compliant and trusted endpoint devices and can restrict the access of noncompliant devices.

SNA uses filters to restrict access. Avaya defines a preconfigured ACT, called SNA Default ACT, for this purpose. For more information about filters and SNA, see *Avaya Ethernet Routing Switch 8800/8600 Security,* (NN46205-601).

Comments? infodev@avaya.com

Chapter 5: QoS and IP filter configuration

Configure Quality of Service (QoS) and IP filters to set up your network to prioritize specific types of traffic to ensure traffic receives the appropriate QoS level and to manage traffic by defining filtering conditions and associating these conditions with specific actions.

QoS and IP filter configuration tasks

This work flow shows you the sequence of tasks you perform to configure QoS and IP filters on the Avaya Ethernet Routing Switch 8800/8600.

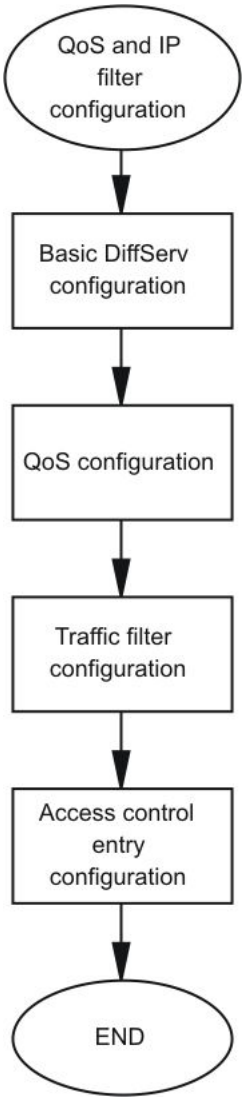


Figure 28: QoS and IP filter configuration tasks

Chapter 6: Basic DiffServ configuration using Enterprise Device Manager

Use DiffServ to implement classification and mapping functions at the network boundary or access points to regulate packet behavior. For information about configuring the QoS level for a MAC address, see *Avaya Ethernet Routing Switch 8600/8800 Configuration — VLANS and Spanning Tree*, (NN46205–517).

Enabling DiffServ on a port

Enable DiffServ so that the switch provides DiffServ-based QoS on that port.

Procedure steps

1. On the Device physical view, select a port.
2. In the navigation tree, open the following folders: **Edit > Port**.
3. Click **General**.
4. Click the **Interface** tab.
5. Select the **DiffServ** checkbox.
6. Click **Apply**.

Configuring Layer 3 trusted or untrusted ports

Configure a port as trusted or untrusted to determine the Layer 3 QoS actions the switch performs. A trusted port honors incoming DSCP markings. An untrusted port overrides DSCP markings.

Procedure steps

1. On the Device physical view, select a port.
2. In the navigation tree, open the following folders: **Edit > Port**.
3. Click **General**.
4. Click the **Interface** tab.
5. Select **core** (trusted) or **access** (untrusted) for the **Layer3Trust** port setting.
6. Click **Apply**.

Configuring Layer 2 trusted or untrusted ports

Configure a port as trusted or untrusted to determine the Layer 2 QoS actions the switch performs. A trusted port (override false) honors incoming 802.1p bit markings. An untrusted port (override true) overrides 802.1p bit markings.

Procedure steps

1. On the Device physical view, select a port.
2. In the navigation tree, open the following folders: **Edit > Port**.
3. Click **General**.
4. Click the **Interface** tab.
5. To configure the port as a Layer 2 untrusted port, select the **Layer2Override8021p** checkbox.

   By default, all ports are Layer 2 trusted (the **Layer2Override8021p** checkbox is cleared)..
6. Click **Apply**.

Configuring the port QoS level

Use the default port QoS level to assign a default QoS level for all traffic (providing the packet does not match an ACL to re-mark the packet).

Procedure steps

1. On the Device physical view, select a port.
2. In the navigation tree, open the following folders: **Edit > Port**.
3. Click **General**.
4. Click the **Interface** tab.
5. Configure **QosLevel** as required by selecting a radio button.
6. Click **Apply**.

Configuring the VLAN QoS level

Use the default VLAN QoS level to assign a default QoS level for all traffic (providing the packet does not match an ACL to re-mark the packet).

Prerequisites

- A configured VLAN exists. If you configure a new VLAN, you configure the QoS level as part of that configuration.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **VLAN**.
2. Click **VLANs**.
3. Click the **Advanced** tab.
4. Double-click a row in the **QosLevel** column, and then select the level.
5. Click **Apply**.

Comments? infodev@avaya.com

Chapter 7: QoS configuration using Enterprise Device Manager

Configure Quality of Service (QoS) to allocate network resources where you need them most.

For information about statistics, see *Avaya Ethernet Routing Switch 8800/8600 Performance Management,* (NN46205-704).

Broadcast and multicast bandwidth limiting

Use broadcast and multicast bandwidth limiting to restrict the amount of ingress broadcast and multicast traffic on a port. The port drops traffic that violates the bandwidth limit.

You can configure broadcast and multicast bandwidth limiting only by using the CLI or the ACLI.

See Configuring broadcast and multicast bandwidth limiting on page 163.

Configuring port-based shaping

Use egress port-based shaping to bind the maximum rate at which traffic leaves the port.

For information about how to configure queue-based shaping, see Configuring egress queue set queues on page 94.

Procedure steps

1. On the Device Physical View, select a port.
2. In the navigation tree, open the following folders: **Configuration** > **Edit** > **Port**.
3. Click **General**.
4. From **Interface** tab, under**EgressRateLimitState**, select **enable**.
5. From **EgressRateLimit**, enter an egress rate limit in kilobits per second.
6. Click **Apply**.

Configuring a policy-based policer

Use a QoS policy to configure peak and service policing rates for specific lane members. Use an Access Control Entry (ACE) to apply the policy to traffic.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **QOS**.
2. Click **Policy**.
3. Click **Insert**.
4. Configure the name and ID as required.
5. Configure the peak and service rates and lane members.

   The peak rate must be greater than or equal to the service rate. You can use the following variable definitions table to help you configure QoS policies.
6. Click **Insert**.

   Configure a filter to use a policy by using the Police parameter as you configure an ACE.
7. To modify a value in the **Policy** tab, double-click the parameter to change. Change the value, and then click **Apply**.
8. To delete a policy, select a policy and click **Delete**.

Variable definitions

Use the data in the following table to configure a policy-based policer.

Variable	Value
GpId	Identifies a global policer (GP) ID value that corresponds to the local policer. Valid values range from 1–16383.
PeakRate	Identifies a local policer peak rate in kilobits per second equal to the corresponding GP ID.
SvcRate	Identifies a local policer service rate in kilobits per second equal to the corresponding GP ID.
Name	Specifies an administratively assigned name for this global policer.

Comments? infodev@avaya.com

Variable	Value
LaneMembers	Specifies a port number for a set of lanes.

Configuring an egress queue set

Configure an egress queue set to apply the same egress queue configuration (a template) to a group (set) of ports.

Important:
If you add or modify an egress queue set, you must restart the switch.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **QOS**.
2. Click **Egress Queue Set**.
3. Click **Insert**.
4. Configure the ID or accept the default value.
5. Choose either an 8- or 64-queue template.

   10/100/1000 Mb/s ports must use the eight-queue template.
6. Configure the number of balanced queues, high-priority queues, and low-priority queues.
7. Configure the name and port members.
8. Click **Apply**.
9. Click **Insert**.

   A message indicates that you must restart the switch to apply the changes. Restart the switch after you make all configuration changes.
10. To delete an egress queue set, select the queue set to delete and click **Delete**.

Variable definitions

Use the data in the following table to configure an egress queue set.

Variable	Value
Id	Specifies a value that uniquely identifies the egress queue template.
MaxQueues	Specifies the maximum number of queues in this template, either 8 or 64. The default is 8.
BalancedQueues	Specifies the total number of balanced queues in this template. The range is 0–48.
BalancedQList	Specifies the list of balanced queues in this template.
HiPriQueues	Specifies the total number of high-priority queues in this template. The range is 0–64.
HiPriQList	Specifies the list of high-priority queues in this template.
LoPriQueues	Specifies the total number of low-priority queues in this template. The range is 0–8.
LoPriQList	Specifies the list of low-priority queues in this template.
Name	Specifies an administratively assigned name for this egress queue template.
PortMembers	Specifies the port members to add to the egress queue template.
Apply	Applies the egress queue template.

Configuring egress queue set queues

Establish queue-based shapers on egress queue set queues. Egress queue sets define the QoS treatment that traffic receives. Configure the queue parameters to suit customer QoS requirements.

When you create a new custom queue, you MUST re-configure the default values provided for the new queue to suit customer QoS requirements.

You can modify some egress queue set queue attributes (Name, MinRate, MaxRate, and MaxLength) for custom queues. You cannot modify queueing style. To modify queueing style, create a new egress queue set with the desired queueing styles.

As you change the queue set queue parameters, do not use the Refresh button, or you erase your changes. Instead, after you make changes, click Apply, and then click Close.

Prerequisites

- An egress queue set exists.

Important:

If you modify an applied egress queue set queue, you must restart the switch.

Comments? infodev@avaya.com

Important:

For each Balanced queue, you must specify a desired minimum rate (min-rate) guarantee and a maximum-rate (max-rate) limit.

For Priority queues (either high or low priority), a minimum rate guarantee does not apply. Configure only a rate limit (max-rate).

The sum of minimum rate guarantees must be less than the port line rate minus the sum of high-priority queue rate limits. If this condition is not met, minimum rates are not guaranteed.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **QOS**.
2. Click **Egress Queue Set**.
3. Select the queue set for which you want to configure queues, and then click **Queue**.
4. On the **Queue** tab, double-click a desired attribute and change the attribute.
5. Click **Apply** to apply the desired attributes. Do not click **Refresh**.
6. If you modify an applied queue set, reapply the queue set, save the configuration, and then restart the switch. You can click **Refresh** on the **Egress Queue Set** tab to see that **Apply** is false after you change the queue parameters.

Variable definitions

Use the data in the following table to configure queues.

Variable	Value
Queue Set Id	Specifies the ID of the queue set.
Qid	Specifies the queue offset from the base queue for this port. Valid values range from 0–63.
Name	Specifies the Networks Service Class (NSC) for this egress queue.
Style	Specifies the egress queue style. Valid values are • hipri (high priority) • balanced • lopri (low priority)

Variable	Value
MinRate	Specifies the egress queue minimum rate guarantee in Kb/s. Applies to balanced and low priority queues only.
MaxRate	Specifies the egress queue maximum rate in Kb/s.
MaxLength (in pages)	Specifies the maximum queue length.

Modifying an egress queue set or queue

You can modify some of the egress queue set parameters for custom queues.

Important:
If you modify an egress queue set, you must restart the switch.

Prerequisites

- An egress queue set exists.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **QOS**.
2. Click **Egress Queue Set**.
3. Change the **Name** or **PortMember** attributes as required.

   To change an attribute, double-click the desired parameter, and then choose the new parameter from the list.

   You cannot change any other Egress Queue Set parameter on this tab. If you must change other parameters, delete the queue set, and then create a new one.
4. Click **Apply**.
5. To change the queue parameters, select a queue set, and then click **Queue**.
6. You can modify any parameter that does not appear dimmed. After you make the changes, click **Apply**.
7. Reapply the queue set corresponding to this queue.

   You can use the Refresh button on the Egress Queue Set tab to see that Apply is indeed false after you change the queue parameters.

8. To save the configuration, select the chassis and open the following folders: **Configuration** > **Edit**.
9. Click **Chassis**.
10. In the **System** tab, select **SaveRuntimeConfig** or **SaveBootConfig** under the **ActionGroup1** options.
11. To restart the switch, click **Configuration** > **Edit** > **Chassis**. On the **System** tab, in the **ActionGroup4** section, select **hardReset**, and then click **Apply**.

Modifying ingress 802.1p to QoS mappings

You can modify the ingress 802.1p to QoS mappings to change traffic priorities. However, Avaya recommends that you use the default mappings.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **QOS**.
2. Click **IngressMap**.
3. Click the **Ingress 8021p to QoS** tab.
4. Modify the QoS mappings as required.
5. Click **Apply**.

Variable definitions

Use the data in the following table to modify 802.1p mappings.

Variable	Value
InIeee8021p	Specifies the ingress IEEE 802.1p priority. The range is 0–7.
QoSLevel	Specifies the internal QoS level. The range is 0–7.

Modifying ingress DSCP to QoS mappings

You can modify the ingress DSCP to QoS mappings to change traffic priorities. However, Avaya recommends that you use the default mappings.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **QOS**.
2. Click **IngressMap**.
3. Click the **Ingress DSCP to QoS** tab.
4. Modify the QoS mappings as required.
5. Click **Apply**.

Variable definitions

Use the data in the following table to modify DSCP mappings.

Variable	Value
InDscp	Specifies the ingress DSCP value, in decimal. The range is 0-63.
InDscpBinaryFormat	Specifies the ingress DSCP value, in binary.
QoSLevel	Specifies the internal QoS level. The range is 0–7.

Modifying ingress MPLS to QoS mappings

You can modify the ingress Multiprotocol Label Switching (MPLS) to QoS mappings to change traffic priorities. However, Avaya recommends that you use the default mappings.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **QOS**.
2. Click **IngressMap**.
3. Click the **Ingress MPLS Exp Bit to QoS** tab.
4. Modify the QoS mappings as required.
5. Click **Apply**.

Comments? infodev@avaya.com

Variable definitions

Use the data in the following table to modify MPLS mappings.

Variable	Value
MplsExp	Specifies the MPLS Exp level. The range is 0–7.
Level	Specifies the internal QoS level. The range is 0–7.

Modifying egress QoS to 802.1p mappings

You can modify the egress QoS to 802.1p mappings to change traffic priorities. However, Avaya recommends that you use the default mappings.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **QOS**.
2. Click **EgressMap**.
3. In the **Egress QoS to 8021p** tab, modify the QoS mappings as required.
4. Click **Apply**.

Variable definitions

Use the data in the following table to modify 802.1p mappings.

Variable	Value
QosLevel	Specifies the internal QoS level. The range is 0–7.
OutIeee8021p	Specifies the egress IEEE 802.1p priority. The range is 0–7.

Modifying egress QoS to DSCP mappings

You can modify the egress QoS to DSCP mappings to change traffic priorities. However, Avaya recommends that you use the default mappings.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **QOS**.
2. Click **EgressMap**.
3. Click the **Egress QoS to DSCP** tab.
4. Modify the QoS mappings as required.
5. Click **Apply**.

Variable definitions

Use the data in the following table to modify DSCP mappings.

Variable	Value
QosLevel	Specifies the internal QoS level. The range is 0–7.
OutDscp	Specifies the egress DSCP value, in decimal. The range is 0-63.
OutDscpBinaryFormat	Specifies the egress DSCP value, in binary.

Modifying egress QoS to MPLS mappings

You can modify the egress QoS to MPLS mappings to change traffic priorities. However, Avaya recommends that you use the default mappings.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **QOS**.
2. Click **EgressMap**.

Comments? infodev@avaya.com

3. Click the **Egress QoS to MPLS Exp Bit** tab.
4. Modify the QoS mappings as required.
5. Click **Apply**.

Variable definitions

Use the data in the following table to modify MPLS mappings.

Variable	Value
QosLevel	Specifies the internal QoS level. The range is 0–7.
MplsExp	Specifies the MPLS Exp level. The range is 0–7.

Comments? infodev@avaya.com

Chapter 8: Traffic filter configuration using Enterprise Device Manager

Use traffic filtering to provide security by blocking unwanted traffic and prioritizing other traffic.

For information about statistics, see *Avaya Ethernet Routing Switch 8800/8600 Performance Management,* (NN46205-704).

Traffic filter configuration procedures

This task flow shows you the sequence of procedures you perform to configure traffic filters.

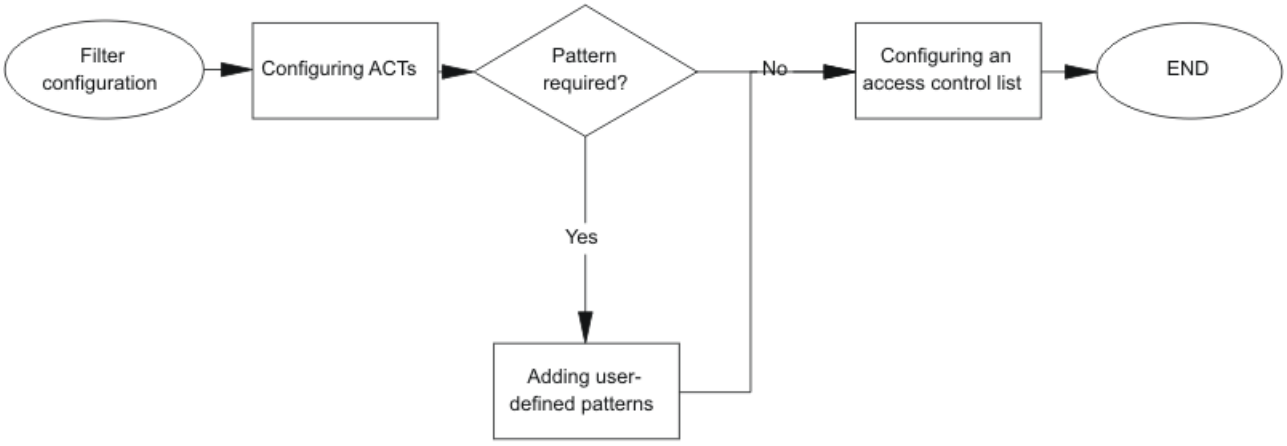


Figure 29: Traffic filter configuration procedures

Configuring ACTs

Use an access control template (ACT) to specify all possible match fields for an access control list (ACL).

Prerequisites

- Add patterns before you activate the ACT (Apply = true).

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. To add a new ACT, click **Insert**.
4. Type an **ActId** or accept the default ACT ID.
5. Name the ACT.
6. Select the Address Resolution Protocol (ARP), Ethernet, IP, protocol, and IPv6 attributes you require.
7. Click **Insert**.
8. If you need to add a pattern, you must do so before you activate the ACT.
9. On the **ACT** dialog box, select **true** to activate the ACT you just configured.

   After you configure Apply to true, you can no longer modify the ACT. If you require different attributes or patterns, you must delete the ACT and create a new one.
10. To delete an ACT, select the ACT, and then click **Delete**.

    You cannot delete an ACT if an ACL references it. You must first delete the ACL.

Variable definitions

Use the data in the following table to configure ACTs.

Variable	Value
ActId	Specifies a unique identifier for the ACT. The range is 1–4096.
Name	Specifies a descriptive user-defined name for the ACT entry.
ArpAttrs	Specifies one of the following ARP attributes: • none • operation (the only valid option for ARP attributes)

Comments? infodev@avaya.com

Variable	Value
	The default is none.
EthernetAttrs	Specifies one or more of the following Ethernet attributes: • none • srcMac • dstMac • etherType • port • vlan • vlanTagPrio The default is none.
IpAttrs	Specifies one or more of the following IP attributes: • none • scrip • dstip • ipFragFlag • ipOptions • ipProtoType • dscp The default is none.
ProtocolAttrs	Specifies one or more of the following protocol attributes: • none • tcpSrcPort • udpSrcPort • tcpDstPort • udpDstport • tcpFlags • icmpMsgFlags The default is none.
Ipv6Attrs	Specifies one or more of the following protocol attributes: • none • srcIpv6 • dstIpv6 • nextHdr

Variable	Value
	The default is none.
Apply	Indicates whether the ACT applies.

Adding a user-defined pattern

Add a user-defined pattern to which the filter can match. You can configure up to three patterns for each ACT.

You can insert a pattern only into an inactive ACT.

Prerequisites

- An ACT exists.
- You did not apply the ACT.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. On the **ACT** tab, select the ACT in which to insert a pattern.
4. Click **Pattern** icon shown on the task bar above.
5. Click **Insert**.
6. Configure the pattern, and then click **Insert**.

   Important:

   After you insert the pattern, you cannot modify the base pattern on which this user-defined pattern is based. To change the base pattern, you must first delete the associated ACEs and then reconfigure and reenable them after modifying the ACT pattern.
7. To activate the ACT, on the **ACT** tab, set **Apply** to **true** for the ACT.

Comments? infodev@avaya.com

Variable definitions

Use the data in the following table to configure ACT patterns.

Variable	Value		
Name	Specifies a descriptive user-defined name for the ACL pattern entry.		
Base	Specifies one of the following as the user-defined header for the ACEs of the ACL: (The default is none.)		
	• none • macSrcBegin • ipHdrBegin • ipTosBegin • ipDstBegin • tcpDstportBegin • udpSrcportBegin • ipHdrEnd • updEnd	• etherBegin • ethTypeLenBegin • ipOptionsBegin • ipProtoBegin • tcpBegin • tcpFlagsEnd • udpDstportBegin • icmpMsgBegin • ipv6HdrBegin	• macDstBegin • arpBegin • ipPayloadBegin • ipSrcBegin • tcpSrcportBegin • udpBegin • etherEnd • tcpEnd
Offset	Configures the offset in bits to the beginning offset with the selected header option as a base. Valid values are 0–76800. The default is 0.		
Length	Configures the number of bits to extract from the beginning of the offset. Valid values are 1–56. The default is 1.		

Configuring an access control list

Use an ACL to specify an ordered list of ACEs, or filter rules. The ACEs provide specific actions for the filter to perform.

When you create an ACL with the type inVlan that uses an ACT based on the source IP address, the ACL no longer works after the ARP aging time elapses. This does not create a security breach. For a solution to this issue, see Workaround for inVlan, srcIp ACL on page 351.

When you create an ACL with the type inVlan that uses an ACT based on the source IP address, the ACL no longer works after the ARP aging time elapses. This does not create a security breach. See Appendix A of *Avaya Ethernet Routing Switch Configuration — QoS and Traffic Filters,* (NN46205-507) for a workaround for this issue.

To modify an ACL parameter, double-click the parameter you wish to change. Change the value, and then click Apply. You cannot change a parameter that appears dimmed; in this case, delete the ACL and configure a new one.

Prerequisites

- The ACT exists.
- You applied the ACT.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Click **Insert**.
5. Type an ACL ID from 1 to 4096 or accept the default value.
6. Click **[...]** besides the **ActId** field to select an ACT ID.
7. Select an Act ID and then click **Ok**.
8. Specify whether the ACL is VLAN or port-based, and whether it is ingress (in) or egress (out).
9. Specify a name for the ACL.
10. If the ACL is VLAN-based, click the **VlanList** ellipsis (**...**) and then choose a VLAN list.
11. If the ACL is port-based, select the **PortList** by clicking the ellipsis (**...**).
12. Select the desired ports, and then click **Ok**.
13. Configure the **DefaultAction** and the **GlobalAction**.
14. Enable or disable the **State**, as required.
15. Click **Insert**.
16. To delete an ACL, select the ACL and click **Delete**.

Variable definitions

Use the data in the following table to configure an ACL.

Comments? infodev@avaya.com

Variable	Value
AclId	Specifies a unique identifier for the ACL from 1–4096.
ActId	Specifies a unique identifier for the ACT entry from 1–4096.
Type	Specifies whether the ACL is VLAN- or port-based. Valid options are • inVlan • outVlan • inPort • outPort **Important:** The inVlan and outVlan ACLs drop packets if you add a VLAN after ACE creation.
Name	Specifies a descriptive user-defined name for the ACL.
VlanList	For inVlan and outVlan ACL types, specifies all VLANs associated with the ACL.
PortList	For inPort and outPort ACL types, specifies the ports associated with the ACL.
DefaultAction	Specifies the action taken when no ACEs in the ACL match. Valid options are deny and permit, with permit as the default. Deny means the system drops the packets; permit means the system forwards packets.
GlobalAction	Indicates the action applied to all ACEs that match in an ACL: • none • mirror • count • mirror-count • count-ipfix • ipfix • mirror-count-ipfix • mirror-ipfix The default is none. If you enable mirroring, ensure that you specify the source or destination mirroring ports:

Variable	Value
	• For R modules in Tx mode: specify ports in the Edit, Diagnostics, Port Mirrors tab • For RS and 8800 modules, or R modules in Rx mode: specify ports in the ACE Debug tab
State	Enables or disables all of the ACEs in the ACL. The default value is enable.
PktType	Specifies IPv4 or IPv6. The default is IPv4.
AceListSize	Indicates the number of ACEs in an ACL.

Chapter 9: Access control entry configuration using Enterprise Device Manager

Use an access control entry (ACE) to define a pattern (found in a packet) and the desired behavior for packets that carry the pattern.

ACEs of type inVlan with an ACT that includes srcIp and with an access control list (ACL) default action of deny, require additional configuration to function properly. See Workaround for inVlan, srcIp ACL on page 351.

ACEs of type inVlan with an access control template (ACT) that includes srcIp, and with an access control list (ACL) default action of deny, require additional configuration to function properly.

Alternatively, Avaya recommends that you create ACLs with a default action of permit, and with an ACE mode of deny. For deny or permit ACLs or ACEs, the default action and the mode must be opposite for the ACE (filter) to have meaning.

Configuring ACEs

Use an ACE to define filter actions, for example, re-marking the DSCP, or mirroring.

Prerequisites

- The ACL exists.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the ACL to which to add an ACE.

5. Click **ACE** icon in the task bar above.
6. Click **Insert**.
7. Configure the ACE ID, or accept the default.
8. Name the ACE.
9. Choose the mode: **deny** (drop packets) or **permit** (forward packets).

   Caution:
   Risk of packet loss

   Avaya recommends that you do not select copyToPrimaryCp or copyToSecondaryCp. If you select the copyToPrimaryCp parameter, the switch sends packets to the CP, which can overload it. You can use the Packet Capture Tool (PCAP), rather than select the parameter copyToPrimaryCp.
10. Configure the ACE actions and flags as required.
11. Click **Insert**.
12. To enable the ACE, in the **ACE Common** tab, set **AdminState** to **enable**, and then click **Apply**.
13. To delete an ACE Common entry, select the entry and click **Delete**.

Variable definitions

Use the data in the following table to configure ACE actions and flags.

Variable	Value
AceId	Specifies a unique identifier and priority for the ACE.
AclId	Specifies the ACL ID.
Name	Specifies a descriptive user-defined name for the ACE. The system automatically assigns a name if you do not type one.
AdminState	Indicates the status of the ACE as enabled or disabled. You can modify an ACE only if you disable it.
OperState	Indicates the current operational state of the ACE.
Mode	Indicates the operating mode for this ACE. Valid options are deny and permit, with deny as the default.
MltIndex	Specifies whether to override the MLT-index picked by the MLT algorithm when the system sends a packet from MLT ports. Valid values range from 0–8, with 0 as the default. Multicast traffic does not support the MLT index.

Comments? infodev@avaya.com

Variable	Value
RemarkDscp	Specifies whether the DSCP parameter marks nonstandard traffic classes and local-use Per-Hop Behavior. The default is disable.
RemarkDot1Priority	Specifies whether Dot1 Priority, as described by Layer 2 standards (802.1Q and 802.1p) is enabled. The default is disable.
Police	Specifies the policer. Valid values range from 0–16383, with 0 (zero) as the default. When you do not want to use policing, configure the value to 0. Configure a policer using the QoS, Policy tab.
RedirectNextHop	Redirects matching IP traffic to the next hop.
RedirectUnreach	Configures the desired behavior for redirected traffic when the specified next hop is not reachable. The default value is deny.
EgressQueue	Specifies a 10/100/1000 Mb/s module egress queue to which to send matching packets. If you specify a value greater than 8, it does not apply to the 10/100/1000 Mb/s module because this module supports only 8 queues. However, the value applies to the 1 Gb/s and 10 Gb/s module types. The default value is 64.
EgressQueue1g	Specifies a 1 Gb/s module egress queue to which to send matching packets. The default value is 64.
EgressQueue10g	Specifies a 10 Gb/s module egress queue to which to send matching packets. The default value is 64.
EgressQueueADSSC	Identifies the configured ACE ADSSC. The default is disable.
StopOnMatch	Enables or disables the stop-on-match option. This option specifies whether to stop or continue after an ACE matches the packet. When this ACE matches, the switch does not attempt a match on other ACEs with lower priority. The default is disable.
Flags	Specifies one of the following flag values: • none—No action (default value) • count—Enables or disables counting if a packet matches the ACE • copyToPrimaryCp—Enables or disables the copying of matching packets to the primary CP • copyToSecondaryCp—Enables or disables the copying of matching packets to the secondary CP • mirror—Enables or disables the mirroring of matching packets to an interface

Variable	Value
	If you enable mirroring, ensure that you also configure the appropriate parameters: • For R, RS, and 8800 modules in Rx mode, and for RS and 8800 modules: DstPortList, DstVlanId, or DstMltId. • For R modules in Tx mode: configure the Edit, Diagnostics, Port Mirrors tab.
DstPortList	Specifies the ports to which to mirror traffic.
DstVlanId	Specifies the VLAN to which to mirror traffic.
DstMltId	Specifies the Multilink Trunking (MLT) group to which to mirror traffic.
IpfixState	Specifies whether IPFIX is enabled or disabled. The default is disable.
RedirectNextHopIpv6	Redirects matching IPv6 traffic to the next hop.

Configuring ACE actions

Use the Action/Debug tab to configure the actions of an ACE or to modify the ACE. Actions determine the process that occurs when a packet matches (or does not match) an ACE. Use debug actions (flags) to use filters for troubleshooting and monitoring procedures.

Prerequisites

- The ACE exists.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL on the **ACL** tab.
5. Click **ACE** icon in the task bar above.
6. Select an **AceId**.

Comments? infodev@avaya.com

7. Click **Action/Debug** icon in the task bar above.
8. Configure the actions as required, and then click **Apply**.

Modifying ACE parameters

Modify ACE parameters so that the filter uses different parameters.

Prerequisites

- The ACE exists.

Procedure steps

1. Navigate to the **ACE Common** tab.
2. Except for the debug actions (flags), disable the **AdminState** of the ACE before you perform modifications.
3. Double-click the ACE parameter to change. Change the parameter as required.
4. Re-enable the **AdminState** if required, and then click **Apply**.

Configuring ACE ARP entries

Use ACE ARP entries so that the filter looks for ARP request or response packets.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has ARP attributes.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select a parameter for the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. Select a parameter for the appropriate ACE.
7. Click **Arp** icon in the task bar above.
8. Click **Insert**.
9. Select ARP request or response.
10. Click **Insert**.

Variable definitions

Use the data in the following table to configure ARP ACEs.

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the ACE index.
Type	Specifies the ACE ARP operation. The only option is operation.
Oper	Specifies the operator for the ACE ARP operation. The only valid option is eq (equal).
Value	Specifies the ARP packet type. Valid options are arpRequest and arpResponse.

Viewing all ACE ARP entries for an ACL

View all of the ACE ARP entries associated with an ACL.

Comments? infodev@avaya.com

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **Arp** icon in the task bar above.

   The ACE ARP, ACL (x) dialog box appears showing all ARP entries.
6. To modify a parameter, double-click the parameter, select the option, and then click **Apply**.

Configuring an ACE Ethernet source address

Use ACE Ethernet source address entries so that the filter looks for specific Ethernet source addresses.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has Ethernet srcMac attributes.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. Select the appropriate ACE.

7. Click **Eth**.
8. Click **Insert**.
9. Specify the ACE Ethernet operation.
10. In the **List** dialog box, specify the Ethernet source address.
11. Click **Insert**.

Variable definitions

Use the data in the following table to configure Ethernet ACEs.

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the associated ACE index.
Oper	Specifies the operators for the source MAC address: • eq—exact match • ne—not equal • le—less than or equal to • ge—greater than or equal to
List	Specifies the MAC address to match in the following format: • a single MAC address • a range of MAC addresses • a list of MAC addresses

Configuring an ACE Ethernet destination address

Use ACE Ethernet destination address entries so that the filter looks for specific Ethernet destination addresses.

Comments? infodev@avaya.com

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has Ethernet dstMac attributes.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. On the **ACE Common** tab, select the appropriate ACE.
7. Click **Eth** icon in the task bar above.
8. Click the **Destination Address** tab.
9. Click **Insert**.
10. Specify the ACE Ethernet operation.
11. In the **List** box, specify the Ethernet destination address.
12. Click **Insert**.

Configuring an ACE LAN traffic type

Use ACE Ethernet type entries so that the filter looks for specific LAN traffic packets: IP, ARP, IPX-802.3, IPX-802.2, IPX-SNAP, IPX-Ethernet2, AppleTalk, Dec-Lat, Dec-Other, SNA-802.2, SNA-Ethernet2, NetBios, XNS, VINES, IPv6, rRAPR, and PPPoE.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has Ethernet etherType attributes.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. On the **ACE Common** tab, select the appropriate ACE.
7. Click **Eth** icon in the task bar above.
8. Click the **Ethernet Type** tab.
9. Click **Insert**.
10. Specify the operation type.
11. In the **TypeList** box, enter the Ethernet types. Specify values in the following order, for example, ip, arp, rarp or 1, 2, 3–5.
12. Click **Insert**.

Variable definitions

Use the data in the following table to help you configure Ethernet ACEs.

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the associated ACE index.
TypeOper	Identifies Ethernet type operators. Valid values are • eq—exact match • ne—not equal

Comments? infodev@avaya.com

Variable	Value
TypeList	Specifies the Ethernet type. Entries include: 0 to 0xffff or ip, arp, ipx802.3, ipx802.2, ipxSnap, ipxEthernet2, appleTalk, decLat, decOther, sna802.2, snaEthernet2, netBios, xns, vines, ipv6, rarp, and PPPoE.

Configuring an ACE Ethernet VLAN tag priority

Use ACE Ethernet VLAN tag priority entries so that the filter looks for specific VLAN tag priorities.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has Ethernet vlanTagPrio attributes.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. On the **ACE Common** tab, select the appropriate ACE.
7. Click **Eth** icon in the task bar above.
8. Click the **Vlan Tag Priority** tab.
9. Click **Insert**.
10. Specify the operation type.
11. In the **VlanTagPrio** box, select the priority bits.
12. Click **Insert**.

Variable definitions

Use the data in the following table to configure tag priorities.

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the associated ACE index.
Oper	Specifies the operators for the ACE Ethernet VLAN tag priority: • eq—exact match • ne—not equal
VlanTagPrio	Specifies the priority bits (3-bit field) from the 802.1Q/p tag: • zero • one • two • three • four • five • six • seven • undefined

Configuring an ACE Ethernet port

Use ACE Ethernet port entries so that the filter looks for traffic on specific ports. You can only insert an ACE Common Ethernet port for VLAN ACL types.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has Ethernet port attributes.

Comments? infodev@avaya.com

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. On the **ACE Common** tab, select the appropriate ACE.
7. Click **Eth** icon in the task bar above.
8. Click the **Port** tab.
9. Click **Insert**.
10. Specify the operation type.
11. Click the **Port** ellipses (**...**).
12. Choose the ports.
13. Click **OK**.
14. Click **Insert**.

Variable definitions

Use the data in the following table to configure ACE Ethernet ports.

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the associated ACE index.
Oper	Specifies the operators for the ACE Ethernet port: • eq—exact match • ne—not equal
Port	Specifies the port or port list on which to perform a match.

Configuring an ACE Ethernet VLAN ID

Use ACE Ethernet VLAN ID entries so that the filter looks for traffic on specific VLANs. You can insert an ACE Ethernet VLAN ID only for ACL VLAN types.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has Ethernet vlan attributes.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. On the **ACE Common tab**, select the appropriate ACE.
7. Click **Eth** icon in the task bar above.
8. Click the **Vlan Id** tab.
9. Click **Insert**.
10. Specify the operation type.
11. Enter the **VlanIdList**.
12. Click **Insert**.

Variable definitions

Use the data in the following table to configure VLAN IDs.

Comments? infodev@avaya.com

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the associated ACE index.
Oper	Specifies the operators for the ACE Ethernet VLAN ID: • eq—exact match • ne—not equal
VlanIdList	Specifies the VLAN ID on which to perform a match.

Viewing all ACE Ethernet entries for an ACL

View all of the ACE Ethernet entries associated with an ACL.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **Eth** icon in the task bar above to view all of the ACE Ethernet entries.

Variable definitions

Use the data in the following table to youconfigure ACEs.

Variable	Value
AclId	Specifies the ACL Ethernet index.
AceId	Specifies the ACE Ethernet index.
SrcAddrList	Specifies the list of Ethernet source addresses to match.
ScrAddrOper	Specifies the operators for the ACE Ethernet source MAC address.

Variable	Value
DstAddrList	Specifies the list of Ethernet destination addresses to match.
DstAddrOper	Specifies the operators for the ACE Ethernet destination MAC address.
EtherTypeList	Specifies the EtherType value from the Ethernet header. For example, ARP uses 0x0806 and IP uses 0x0800. Platform support determines the behavior for 802.1Q/p tagged packets. The EtherType for 802.1Q tagged frames is 0x8100. The range is 0–65535 and supports lists and ranges of values. An invalid Ether-type of 65536 indicates that you do not want the parameter in the match criteria.
EtherTypeOper	Specifies the Ethernet type operators.
VlanTagPrio	Specifies the priority bits (3-bit field) from the 802.1Q/p tag.
VlanTagPrioOper	Specifies the operators for the ACE Ethernet VLAN tag priority.
Port	Specifies the port number or port list to match.
PortOper	Specifies the operator for the ACE Ethernet port.
VlanIdList	Specifies the VLAN ID to match.
VlanIdOper	Specifies the operator for the ACE Ethernet VLAN ID.

Configuring an ACE IP source address

Use ACE IP source address entries to have the filter look for specific source IP addresses.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has IP srcIp attributes.

Comments? infodev@avaya.com

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. On the **ACE Common** tab, select the appropriate ACE.
7. Click **IP** icon in the task bar above.
8. Click **Insert**.
9. Specify the operation type.
10. In the **List** box, enter the source IP address.
11. Click **Insert**.

Variable definitions

Use the data in the following table to configure IP source address ACEs.

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the associated ACE index.
Oper	Specifies the operators for the ACE IP source address: • eq—exact match • ne—not equal • le—less than or equal to • ge—greater than or equal to
List	Specifies the source IP address in the following format: • a single IP address • a range of IP addresses • a list of IP addresses

Configuring an ACE IP destination address

Use ACE IP destination address entries to have the filter look for specific destination IP addresses.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has IP dstIp attributes.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. On the **ACL** tab, select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. On the **ACE Common** tab, select the appropriate ACE.
7. Click **IP** icon in the task bar above.
8. Click the **Destination Address** tab.
9. Click **Insert**.
10. Specify the operation type.
11. In the **List** box, enter the destination IP address. This value can be a single address, a range, or a list.
12. Click **Insert**.

Variable definitions

Use the data in the following table to configure IP destination address ACEs.

Comments? infodev@avaya.com

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the associated ACE index.
Oper	Specifies the operators for the ACE IP destination address: • eq—exact match • ne—not equal • le—less than or equal to • ge—greater than or equal to
List	Specifies the destination IP address in the following format: • a single IP address • a range of IP addresses • a list of IP addresses

Configuring an ACE IP DSCP

Use ACE IP DSCP entries to have the filter look for packets with specific DSCP markings.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has IP dscp attributes.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. On the **ACL** tab, select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. On the **ACE Common** tab, select the appropriate ACE.

7. Click **IP** icon in the task bar above.
8. Click the **DSCP** tab.
9. Click **Insert**.
10. Specify the operation type.
11. In the **List** box, enter the count for the DSCP values.
12. Click **Insert**.

Variable definitions

Use the data in the following table to configure IP DSCP ACEs.

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the associated ACE index.
Oper	Specifies the operators for the ACE IP DSCP: • eq—exact match • ne—not equal
List	Specifies a count for the number of discrete ranges entered for the DSCP values. Entries include 0–256, disable, phbcs0, phbcs1, phbaf11, phbaf12, phbaf13, phbcs2, phbaf21, phbaf22, phbaf23, phbcs3, phbaf31, phbaf32, phbaf33, phbcs4, phbaf41, phbaf42, phbaf43, phbcs5, phbcs6, phbef, and phbcs7.

Configuring an ACE IP protocol

Use ACE IP protocol entries to have the filter look for packets of specific protocols; for example, ICMP, TCP, UDP, IPSec-ESP, IPSec-AH, OSPF, VRRP, and SNMP.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has IP ipProtoType attributes.

Comments? infodev@avaya.com

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. On the **ACL** tab, select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. On the **ACE Common** tab, select the appropriate ACE.
7. Click **IP** icon in the task bar above.
8. Click the **Protocol** tab.
9. Click **Insert**.
10. Specify the operation type.
11. In the **List** box, enter the IP protocol type.
12. Click **Insert**.

Variable definitions

Use the data in the following table to configure protocol ACEs.

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the associated ACE index.
Oper	Specifies the operators for the ACE IP protocol: • eq—exact match • ne—not equal
List	Specifies the IP protocol type. Entries include 0–256, undefined, icmp, tcp, udp, ipsecesp, ipsecah, ospf, vrrp, and snmp.

Configuring ACE IP options

Use ACE IP option entries to have the filter look for packets with an IP option specified.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has IP ipOptions attributes.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. On the **ACL** tab, select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. On the **ACE Common** tab, select the appropriate ACE.
7. Click **IP** icon in the task bar above.
8. Click the **Options** tab.
9. Click **Insert**.
10. Specify the logical operator.

    Any is the only valid choice.
11. Click **Insert**.

Variable definitions

Use the data in the following table to configure IP option ACEs.

Variable	Value
AclId	Specifies the ACL index.

Comments? infodev@avaya.com

Variable	Value
AceId	Specifies the associated ACE index.
Oper	Specifies the logical operator for the ACE IP options. Any is the only valid option.

Configuring ACE IP fragmentation

Use ACE IP fragmentation entries to have the filter look for packets with the fragmentation flag set.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has IP ipFragFlag attributes.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. Select the appropriate ACE.
7. Click **IP** icon in the task bar above.
8. Click the **Fragmentation** tab.
9. Click **Insert**.
10. Specify the operator for IP fragmentation.

    Eq is the only valid choice.
11. Specify the fragmentation bits to match from the IP header.
12. Click **Insert**.

Variable definitions

Use the data in the following table to configure fragmentation ACEs.

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the associated ACE index.
Oper	Specifies the operators for ACE IP fragmentation. The only valid value is eq (equals).
Fragmentation	Specifies the IP fragmentation bits to match from the IP header: • noFragment • anyFragment • moreFragment • lastFragment The default is noFragment.

Viewing all ACE IP entries for an ACL

View all of the ACE IP entries associated with an ACL.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **IP** icon in the task bar above to view all ACE IP entries.

Variable definitions

Use the data in the following table to understand ACE parameters.

Variable	Value
AclId	Specifies the ACL IP index.
AceId	Specifies the ACE IP index.
SrcAddrList	Specifies the list of IP source addresses from the IP header to match.
ScrAddrOper	Specifies the operators for the ACE IP source address.
DstAddrList	Specifies the list of IP destination addresses from the IP header to match.
DstAddrOper	Specifies the operators for the ACE IP destination address.
DscpList	Specifies how the 6-bit DSCP parameter from the TOS byte in the IPv4 header encodes PHB information following RFC 2474.
DscpOper	Specifies the operators for the ACE IP DSCP.
ProtoList	Specifies the IP protocol type from the IP header to match. The range is 0–255.
ProtoOper	Specifies the operators for the ACE IP protocols.
Options	Specifies the IP options to match from the IP header.
OptionsOper	Specifies the logical operator. Any is the only option.
Fragmentation	Specifies the IP fragmentation bits to match from the IP header.
FragOper	Specifies the operator for IP fragmentation.

Configuring an ACE TCP source port

Use ACE TCP source port entries to have the filter look for packets with a specific TCP source port.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has protocol tcpSrcPort attributes.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. Select the appropriate ACE.
7. Click **Proto** icon in the task bar above.
8. Click **Insert**.
9. Specify the operator for the TCP source port.
10. Specify the port number or port list to match.
11. Click **Insert**.

Variable definitions

Use the data in the following table to configure TCP source port ACEs.

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the associated ACE index.
Oper	Specifies the operators for the ACE protocol TCP source port: • eq—exact match • ne—not equal • le—less than or equal to • ge—greater than or equal to
Port	Specifies the port number in the following format: • a single port number • a range of port numbers • a list of port numbers

Configuring an ACE UDP source port

Use ACE UDP source port entries to have the filter look for packets with a specific UDP source port.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has protocol udpSrcPort attributes.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **ACE** icon in the task bar above after it becomes active.
6. Select the appropriate ACE.
7. Click **Proto** icon in the task bar above.
8. Double-click the **UDP Source Port** tab.
9. Click **Insert**.
10. Specify the operator for the UDP source port.
11. Specify the port number or port list to match.
12. Click **Insert**.

Variable definitions

Use the data in the following table to configure UDP source port ACEs.

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the associated ACE index.
Oper	Specifies the operators for the ACE protocol UDP source port: • eq—exact match • ne—not equal • le—less than or equal to • ge—greater than or equal to
Port	Specifies the port number in the following format: • a single port number • a range of port numbers • a list of port numbers

Configuring an ACE TCP destination port

Use ACE TCP destination port entries to have the filter look for packets with a specific TCP destination port.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has protocol tcpDstPort attributes.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.

Comments? infodev@avaya.com

5. Click **ACE** icon in the task bar above.
6. Select the appropriate ACE.
7. Click **Proto** icon in the task bar above.
8. Click the **TCP Destination Port** tab.
9. Click **Insert**.
10. Specify the operator for the TCP destination port.
11. Specify the port number or port list to match.
12. Click **Insert**.

Variable definitions

Use the data in the following table to configure TCP destination port ACEs.

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the associated ACE index.
Oper	Specifies the operators for the ACE protocol TCP destination port: • eq—exact match • ne—not equal • le—less than or equal to • ge—greater than or equal to
Port	Specifies the port number. As noted at the bottom of the tab, potential entries include 0–65535, echo, ftpdata, ftpcontrol, ssh, telnet, dns, http, bgp, h.323, and undefined.

Configuring an ACE UDP destination port

Use ACE UDP destination port entries to have the filter look for packets with a specific TCP destination port.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has protocol udpDstPort attributes.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. Select the appropriate ACE.
7. Click **Proto** icon in the task bar above.
8. Click the **UDP Destination Port** tab.
9. Click **Insert**.
10. Specify the operator for the UDP destination port.
11. Specify the port number or port list to match.
12. Click **Insert**.

Variable definitions

Use the data in the following table to configure UDP destination port ACEs.

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the associated ACE index.
Oper	Specifies the operators for the ACE protocol UDP destination port: • eq—exact match • ne—not equal

Comments? infodev@avaya.com

Variable	Value
	• le—less than or equal to • ge—greater than or equal to
Port	Specifies the port number. Entries include 0–65535, echo, dns, bootpServer, bootpClient, tftp, rip, rtp, rtcp, and undefined.

Configuring an ACE ICMP message type

Use ACE ICMP message type entries to have the filter look for packets of a specific ICMP message type.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has protocol icmpMsgType attributes.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. Select the appropriate ACE.
7. Click **Proto** icon in the task bar above.
8. Click the **Icmp Msg Type** tab.
9. Click **Insert**.
10. Specify the operator for the ICMP message type.

11. In the **List** box, specify the ICMP messages to match.
12. Click **Insert**.

Variable definitions

Use the data in the following table to help you configure ICMP ACEs.

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the associated ACE index.
Oper	Specifies the operators for the ACE protocol ICMP message type: • eq—exact match • ne—not equal
Port	Specifies the port number. Entries include 0–255, echoreply, destunreach, sourcequench, redirect, echo-request, routeradv, routerselect, time-exceeded, param-problem, timestamp-request, timestamp-reply, addressmask-request, addressmask-reply, and traceroute.

Configuring an ACE TCP flag

Use ACE TCP flag entries to have the filter look for packets with a specific TCP flag.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has protocol tcpFlags attributes.

Comments? infodev@avaya.com

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. Select the appropriate ACE.
7. Click **Proto** icon in the task bar above.
8. Click the **TCP Flags** tab.
9. Click **Insert**.
10. Specify the operator for the TCP flags entry.
11. In the **List** box, specify the TCP flags to match.
12. Click **Insert**.

Variable definitions

Use the data in the following table to configure TCP flag ACEs.

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the associated ACE index.
Oper	Specifies the operators for the ACE protocol TCP flags entry: • matchAny • matchAll
List	Specifies the TCP flags—none, fin (finish connection), syn (synchronize), rst (reset connection), push, ack (acknowledge), urg (urgent), and undefined.

Viewing all ACE Protocol entries for an ACL

View all of the ACE Protocol entries associated with an ACL.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **Proto** icon in the task bar above.

   The ACE Protocol, ACL (x) dialog box appears.

Variable definitions

Use the data in the following table to understand the protocol parameters.

Variable	Value
AclId	Specifies the ACL protocol index.
AceId	Specifies the ACE protocol index.
TcpSrcPort	Specifies the port number or port list to match.
TcpSrcPortOper	Specifies the operator for the ACE protocol TCP source port.
UdpSrcPort	Specifies the port number or port list to match.
UdpSrcPortOper	Specifies the operator for the ACE protocol UDP source port.
TcpDstPort	Specifies port number or port list to match.
TcpDstPortOper	Specifies the operator for the ACE protocol TCP destination port.
UdpDstPort	Specifies the port number or port list to match.
UdpDstPortOper	Specifies the operator for the ACE protocol UDP destination port.

Comments? infodev@avaya.com

Variable	Value
IcmpMsgTypeList	Specifies one or a list of ICMP messages to match. The valid range is 0–255 (reserved).
IcmpMsgTypeOper	Specifies the operator for the ACE protocol ICMP message types.
TcpFlagsList	Specifies one or a list of TCP flags to match. The valid range is 0–63.
TcpFlagsOper	Specifies the operator for the ACE protocol TCP flags.

Configuring an ACE Pattern 1 entry

Configure an ACE pattern entry to have the filter look for a specific pattern in a packet.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has a pattern.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. On the **ACL** tab, select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. Select the appropriate ACE.
7. Click **Adv** icon in the task bar above.
8. Click **Insert**.
9. Specify a name for the ACE pattern entry.
10. Specify the operators for the ACE pattern.

11. Assign the pattern value.
12. Click **Insert**.

Variable definitions

Use the data in the following table to configure ACE patterns.

Variable	Value
AclId	Specifies the ACL index.
AceId	Specifies the associated ACE index.
Name	Specifies a descriptive user-defined name for the ACE pattern entry.
Oper	Specifies the operators for the ACE pattern: • eq—exact match • le—less than or equal to • ge—greater than or equal to
Value	Configures the pattern value as a numeric string. The numeric value of each byte is encoded in one octet of the string. Unused bytes remain at the trailing end of string. The Pattern Length field configures the number of bytes to extract from this string.

Configuring an ACE Pattern 2 entry

Configure an ACE pattern entry to have the filter look for a specific pattern in a packet.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has two patterns.

Comments? infodev@avaya.com

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. Select the appropriate ACE.
7. Click **Adv** icon in the task bar above.
8. Click **Pattern 2** tab.
9. Click **Insert**.
10. Specify a name for the ACE pattern entry.
11. Specify the operators for the ACE pattern.
12. Assign the pattern value.
13. Click **Insert**.

Configuring an ACE Pattern 3 entry

Configure an ACE pattern entry to have the filter look for a specific pattern in a packet.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has three patterns.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **ACE** icon in the task bar above.
6. Select the appropriate ACE.
7. Click **Adv** icon in the task bar above.
8. Click **Pattern 3** tab.
9. Click **Insert**.
10. Specify a name for the ACE pattern entry.
11. Specify the operators for the ACE pattern.
12. Assign the pattern value.
13. Click **Insert**.

Viewing all ACE Advanced pattern entries for an ACL

View all of the ACE Advanced entries associated with an ACL.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select the appropriate ACL.
5. Click **Adv** icon in the task bar above.

   The ACE Advanced, ACL (x) dialog box appears.

Variable definitions

Use the data in the following table to configure ACEs.

Variable	Value
AclId	Specifies the ACL pattern index.
AceId	Specifies the ACE pattern index.
Pattern1Name	Specifies the name chosen by the administrator for the ACE pattern 1 entry.
Pattern1Value	Specifies the pattern 1 value as numeric string. The numeric value of each byte is encoded in one octet of the string. Unused bytes are left at the trailing end of string.
Pattern1Oper	Specifies the operators for ACE pattern 1.
Pattern2Name	Specifies the name chosen by the administrator for the ACE pattern 2 entry.
Pattern2Value	Specifies the pattern 2 value as a numeric string.
Pattern2Oper	Specifies the operators for ACE pattern 2.
Pattern3Name	Specifies the name chosen by the administrator for the ACE pattern 3 entry.
Pattern3Value	Specifies the pattern 3 value as a numeric string.
Pattern3Oper	Specifies the operators for ACE pattern 3.

Configuring an ACE IPv6 source address

Configure an ACE IPv6 source address to have the filter look for a specific IPv6 source addresses.

Prerequisites

- The ACE exists.
- The ACL exists.
- The associated ACL packet type must be IPv6.
- The ACT has IPv6 attributes of srcIpv6.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select an IPv6 ACL.
5. Click **ACE** icon in the task bar above.
6. Select an ACE.
7. Click **IPv6** icon in the task bar above.
8. Click the **Source Address** tab.
9. Click **Insert**.
10. Specify the operation and the IPv6 address.
11. Click **Insert**.

Variable definitions

Use the data in the following table to configure IPv6 source or destination address ACEs.

Variable	Value
AclId	Specifies the ACL ID.
AceId	Specifies the ACE ID.
Oper	Specifies the ACE operation. The only option is eq (equals).
List	Specifies the IPv6 address—a binary string of 16 octets in network byte-order. Enter a single IPv6 address, a range of IPv6 addresses, or multiple IPv6 addresses.

Configuring an ACE IPv6 destination address

Configure an ACE IPv6 destination address to have the filter look for a specific IPv6 destination addresses.

The IPv6 parameters that you can configure depend on the ACT configuration.

Comments? infodev@avaya.com

Prerequisites

- The ACE exists.
- The ACL exists.
- The associated ACL packet type must be IPv6.
- The ACT has IPv6 attributes of dstIpv6.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select an IPv6 ACL.
5. Click **ACE** icon in the task bar above.
6. Select an ACE.
7. Click **IPv6** icon in the task bar above.
8. Click the **Destination Address** tab.
9. Click **Insert**.
10. Specify the operation and the Destination Address.
11. Click **Insert**.

Configuring an ACE IPv6 next header

Configure an ACE IPv6 next header to have the filter look for a packets with the next header parameter assigned.

The IPv6 parameters that you can configure depend on the ACT configuration.

Prerequisites

- The ACE exists.
- The ACL exists.
- The associated ACL packet type must be IPv6.
- The ACT has IPv6 attributes of nxtHdr.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select an IPv6 ACL.
5. Click **ACE** icon in the task bar above.
6. Select an ACE.
7. Click **IPv6** icon in the task bar above.
8. Click the **Next Hdr** tab.
9. Click **Insert**.
10. Specify the operation and the Next header parameters.
11. Click **Insert**.

Variable definitions

Use the data in the following table to configure IPv6 next header ACEs.

Variable	Value
AclId	Specifies the ACL ID.
AceId	Specifies the ACE ID.
Oper	Specifies the ACE operation. The options are eq (equal) or ne (not equal).

Comments? infodev@avaya.com

Variable	Value
NxtHdr	Specifies the next header: hop-by-hop, tcp, udp, routing, frag, ipsecESP, ipsecAh, icmpv6, noNxtHdr, undefined.

Viewing IPv6 attributes for an ACL

View all of the ACE IPv6 entries associated with an ACL.

Procedure steps

1. In the navigation tree, open the following folders: **Configuration** > **Security** > **Data Path**.
2. Click **ACL Filters**.
3. Click the **ACL** tab.
4. Select a parameter of an IPv6 ACL.
5. Click **IPv6** icon in the task bar above.

Variable definitions

Use the data in the following table to understand IPv6 ACE parameters.

Variable	Value
AclId	Specifies the unique identifier for the ACL.
AceId	Specifies the unique identifier for the ACE.
SrcAddrList	Lists the source IPv6 addresses.
SrcAddrOper	Specifies equal (eq) or not equal (ne) or any in relation to the listed source addresses.
DstAddrList	Lists the IPv6 destination addresses.
DstAddrOper	Specifies equal (eq) or not equal (ne) or any in relation to the listed source addresses.
NxtHdrNxtHdr	Displays the next header value.
NxtHdrOper	Specifies equal (eq) or not equal (ne) or any in relation to the listed source addresses.

Comments? infodev@avaya.com

Chapter 10: Basic DiffServ configuration using the CLI

Use DiffServ to provide appropriate Quality of Service (QoS) to specific traffic types.

Job aid

The following roadmap lists some of the QoS commands and the parameters that you can use to perform the procedures in this section.

Table 20: Roadmap of QoS CLI commands

Command	Parameter
`config ethernet <port>`	`802.1p-override <enable|disable>`
	`access-diffserv <true|false>`
	`enable-diffserv true`
	`qos-level <0-6>`
`config vlan <vlan id>`	`fdb-static add <mac> port <value> qos <0-6>`
	`fdb-entry qos-level <mac> status <value> <0-6>`
	`qos-level <0-6>`

Enabling DiffServ on a port

Enable DiffServ so that the switch provides DiffServ-based QoS on a port.

Procedure steps

1. Enable DiffServ:

```
config ethernet <port> enable-diffserv
```

Variable definitions

Use the data in the following table to use the **config ethernet <ports> enable-diffserv <true|false>** command.

Variable	Value
enable-diffserv <true|false>	True enables DiffServ for the port or ports selected. If true all other QoS parameter values and functions now take affect and apply. If false, these parameters and settings do not apply. By default, enable-diffserv is false.

Configuring Layer 3 trusted or untrusted ports

Configure a port as trusted or untrusted to determine the Layer 3 QoS actions the switch performs. A trusted port honors incoming DSCP markings. An untrusted port overrides DSCP markings.

Prerequisites

- DiffServ is enabled.

Procedure steps

1. Configure the port as Layer 3 trusted or untrusted:

   ```
   config ethernet <port> access-diffserv <true|false>
   ```

Variable definitions

Use the data in the following table to use the **config ethernet <port>** command.

Comments? infodev@avaya.com

Variable	Value
`access-diffserv <true|false>`	*true* specifies an access port and overrides incoming DSCP bits; *false* specifies a core port and honors and handles incoming DSCP bits. The default is false.

The Enterprise Device Manager field for this parameter is Layer3Trust. A CLI value of true equals a value of access for Device Manger and CLI value of false equals a value of core for Enterprise Device Manager.

Configuring Layer 2 trusted or untrusted ports

Configure a port as trusted or untrusted to determine the Layer 2 QoS actions the switch performs. A trusted port (override disabled) honors incoming 802.1p bit markings. An untrusted port (override enabled) overrides 802.1p bit markings.

Prerequisites

- DiffServ is enabled.

Procedure steps

1. Configure the port as Layer 2 trusted or untrusted:

   ```
   config ethernet <port> 802.1p-override <enable|disable>
   ```

Variable definitions

Use the data in the following table to use the **`config ethernet <port>`** command.

Variable	Value
`802.1p-override <enable|disable>`	*enable* overrides incoming 802.1p bits; *disable* honors and handles incoming 802.1p bits. The default is disable.

Configuring the port QoS level

Use the default port QoS level to assign a default QoS level for all traffic (providing the packet does not match an ACL to re-mark the packet).

Procedure steps

1. Configure the port QoS level:

   ```
   config ethernet <port> qos-level <0-6>
   ```

Variable definitions

Use the data in the following table to use the **`config ethernet <port>`** command.

Variable	Value
`qos-level <0-6>`	Specifies the default QoS level for the port traffic. QoS level 7 is reserved for network control traffic. The default is 1.

Configuring the VLAN QoS level

Change the default port or VLAN QoS levels to assign a default QoS level for all traffic, if the packet does not match an ACL to re-mark the packet.

Procedure steps

1. Configure the VLAN QoS level:

   ```
   config vlan <vlan-id> qos-level <0-6>
   ```


Comments? infodev@avaya.com

<vlan-id> specifies the VLAN ID (1 to 4094) for which to specify the QoS level.

Variable definitions

Use the data in the following table to use the **config vlan <vlan-id>** command.

Variable	Value
qos-level <0-6>	Specifies the default QoS level for the VLAN traffic. QoS level 7 is reserved for network control traffic. The default is 1.

Configuring the QoS level for a MAC address

Apply a QoS level to traffic from specific VLAN MAC addresses to provide special QoS treatment to the packets or to modify the QoS level providing the packet does not match an ACL to re-mark the packet.

Procedure steps

1. Configure the source MAC QoS level for a dynamically learned address:

   ```
   config vlan <vlan id> fdb-entry qos-level <mac> status <value> <0-6>
   ```

2. Configure the source MAC QoS level for a static address:

   ```
   config vlan <vlan id> fdb-static add <mac> port <value> qos <0-6>
   ```

Variable definitions

Use the data in the following table to use the fdb-entry command.

Variable	Value
<mac>	Specifies the MAC address in the format 0x00:0x00:0x00:0x00:0x00:0x00:0x00
status <value>	Specifies the forwarding database (FDB) status (other|invalid|learned|self|mgmt)

Variable	Value
<0-6>	Specifies the QoS level. The default is 1.

Use the data in the following table to use the fdb-static command.

Variable	Value
add <mac>	Adds or configures the source MAC QoS level to a VLAN bridge. <*mac*> specifies the MAC address in the format 0x00:0x00:0x00:0x00:0x00:0x00.
port <value>	<*value*> specifies the port number
qos <0-6>	<*0-6*> specifies the QoS level. The default is 1.

Example of configuring a QoS level for a MAC address

Procedure steps

1. To change the source MAC QoS level to 2 for the MAC address 00:00:00:00:01:0a on VLAN 2 through port 7/26, enter the following command:

   ```
   ERS-8610:5# config vlan 2 fdb-static add 00:00:00:00:01:0a port 7/26 qos 2
   ```


Comments? infodev@avaya.com

Chapter 11: QoS configuration using the CLI

Use the procedures in this section to configure Quality of Service (QoS) on your Avaya Ethernet Routing Switch 8800/8600.

For information about statistics, see *Avaya Ethernet Routing Switch 8800/8600 Performance Management,* (NN46205-704).

Job aid

The following roadmap lists some of the QoS commands and the parameters that you can use to perform the procedures in this section.

Table 21: Roadmap of QoS CLI commands

Command	Parameter
`config ethernet <port>`	`broadcast-bandwidth-limit <value> [<enable|disable>]`
	`broadcast-rate-limit`
	`multicast-bandwidth-limit <value> [<enable|disable>]`
	`multicast-rate-limit`
	`police <kbps> [<enable|disable>]`
	`shape <kbps> [<enable|disable>]`
`config ethernet <slot/port>`	`enable-diffserv <true|false>`
	`access-diffserv <true|false>`
	`qos 802.1p-override <enable|disable>`
`config qos egress-queue-set <id>`	`apply`
	`create qmax <value> [balanced-queues <value>] [hipri-queues <value>] [lopri-queues <value>] [name <value>]`

Command	Parameter
	delete
	info
	name <value>
config qos egress-queue-set <id> port	add <ports>
	info
	remove <ports>
config qos egress-queue-set <id> queue <qid>	info
	name
	set [min-rate <value>] [max-rate <value>] [max-length <value>]
config qos egressmap	1p <level> <ieee1p>
	ds <level> <dscp>
	exp <level> <exp>
	info
config qos ingressmap	1p <ieee1p> <level>
	ds <dscp> <level>
	exp <exp> <level>
	info
config qos policy <policy-id>	create peak-rate <value> svc-rate <value> [lanes <value>] [name <value>]
	delete
	info
	modify peak-rate <value> svc-rate <value>
	name <value>
config qos policy <policy-id> lanes	add <lane-list>
	remove <lane-list>
show port stats egress-queues	[<ports>]
	[queues <value>]
	[verbose]

Comments? infodev@avaya.com

Command	Parameter
show qos config egress-queue-set	all
	egress-queue-set <id> [queues]
	port <ports>
show qos config eqmap <slot-number>	
show qos config policy	lane <lane-no>
	all
	port <ports>
	policy <policy-id>
show qos egressmap	1p [<level>]
	ds [<level>]
	exp
show qos ingressmap	1p [<ieee1p>]
	ds [<dscp>]
	exp
show qos stats egress-queue-set	all [verbose]
	egress-queue-set <id> [verbose]
	port <ports> [verbose]
show qos stats policy	all
	port <ports> [policy <value>]
	lane <lane-no> [policy <value>]

Configuring broadcast and multicast bandwidth limiting

Use broadcast and multicast bandwidth limiting to limit the amount of ingress broadcast and multicast traffic on a port. The switch drops traffic that violates the bandwidth limit.

Procedure steps

1. Configure broadcast bandwidth limiting:

```
config ethernet <port> broadcast-bandwidth-limit <value>
[<enable|disable>]
```

2. Configure multicast bandwidth limiting:

```
config ethernet <port> multicast-bandwidth-limit <value>
[<enable|disable>]
```

Variable definitions

Use the data in the following table to use the **`config eth <port>`** commands.

Variable	Value
`broadcast-bandwidth-limit <value> [<enable|disable>]`	Specifies the bandwidth limit for broadcast traffic from 250–2147483647 Kb/s. **`<enable|disable>`** enables or disables bandwidth limiting. The default is disabled.
`multicast-bandwidth-limit <value> [<enable|disable>]`	Specifies the bandwidth limit for multicast traffic from 250–2147483647 Kb/s. **`<enable|disable>`** enables or disables bandwidth limiting. The default is disabled.

Configuring the port-based shaper

Use port-based shaping to rate-limit all egress (outgoing) traffic to a specific rate.

For information about configuring queue-based shaping, see Configuring an egress queue set queue on page 173.

Procedure steps

1. Configure port-based shaping:

```
config ethernet <port> shape <kbps> [<enable|disable>]
```

Variable definitions

Use the information in the following table to use the command in this procedure.

Comments? infodev@avaya.com

Variable	Value
`<enable|disable>`	Enables or disables port-based shaping on the port. The default is disable.
`<kbps>`	Configures the shaping rate from 1000–10000000 Kb/s.

Configuring a port-based policer for RS and 8800 modules

Use a port-based policer to bandwidth-limit incoming traffic. The system drops or re-marks violating traffic. Only RS and 8800 modules support this policer.

Procedure steps

1. Configure the policing limit and enable or disable policing:

   ```
   config ethernet <port> police <kbps> <enable|disable>
   ```

Variable definitions

Use the following variable definitions table to the commands in this procedure.

Variable	Value
`police <kbps>`	Specifies the ingress rate limit (policing limit) in kilobits per second. The range is 1000–10000000.
`<enable|disable>`	Enables or disables policing (ingress-rate-limiting). The default is enable.

Configuring a policy-based policer

Use a QoS policy to configure peak and service policing rates for specific lane members. Use an ACE to apply the policy to traffic.

Procedure steps

1. Configure a policer (traffic policy):

```
config qos policy <policy-id> create peak-rate <value> svc-
rate <value> [lanes <value>] [name <value>]
```

2. Ensure the configuration is correct:

```
show qos config policy policy <policy-id>
```

Variable definitions

Use the information in the following table to use the **`config qos policy <policy-id>`** command.

Variable	Value
`create peak-rate <value> svc-rate <value> [lanes <value>] [name <value>]`	Configures the following options: • **`create peak-rate`** *<value>* specifies a peak rate value in kilobits per second for the policy. • **`svc-rate`** *<value>* specifies a service rate value in kilobits per second for the policy. • **`lanes`** *<value>* identifies a specific lane or all lanes to which the policy applies. • **`name`** *<value>* specifies a service rate value in kilobits per second for the policy.
`delete`	Deletes an existing policy. You cannot delete a policy if an access control entry references the policy.
`info`	Displays current setting information for the policy.
`modify peak-rate <value> svc-rate <value>`	Configures the following options: • **`modify peak-rate`** *<value>* modifies a peak rate value in kilobits per second for the policy. • **`svc-rate`** *<value>* modifies a service rate value in kilobits per second for the policy.
`name <value>`	Modifies the name of the policer template.

Use the information in the following table to use the **`show qos config policy`** command.

Variable	Value
`all`	Displays all configured policing data.
`lane <lane-no>`	Displays policing data by lane.
`policy <policy-id>`	Displays policing data by policy ID.

Comments? infodev@avaya.com

Variable	Value
`port <ports>`	Displays policing data by port.

Job aid

The following table describes the headings in the show command output.

Table 22: show qos config policy output

Field	Description
PolicerID	Specifies the policer ID number.
Name	Specifies the name of the policer.
peak-rate	Specifies a policer peak rate in Kb/s.
svc-rate	Specifies a local policer service rate in Kb/s.
lanes	Specifies the lane numbers associated with the policy.

Adding lanes to a policy-based policer

Add or remove lanes from a policer so that the policer operates only on specific lane members.

Prerequisites

- The policy exists.

Procedure steps

1. Add lanes from an existing policer:

```
config qos policy <policy-id> lanes add <lane-list>
```

Variable definitions

Use the information in the following table to use the **`config qos policy <policy-id> lanes`** command.

Variable	Value
`add <lane-list>`	Adds lanes to an existing policer template.
`remove <lane-list>`	Removes lanes from an existing policer template.

Configuring an egress queue set

Configure an egress queue set to apply the same egress queue configuration (a template) to a group (set) of ports.

Important:
If you add or modify an egress queue set, you must restart the switch.

Procedure steps

1. Configure the egress queue set template:

   ```
   config qos egress-queue-set <id> create qmax <value>
   [balanced-queues <value>] [hipri-queues <value>] [lopri-
   queues <value>] [name <value>]
   ```

2. Associate ports with the egress queue set:

   ```
   config qos egress-queue-set <id> port add <port>
   ```

   The system verifies that the requested port types support the number of queues in the egress queue set. If you add new ports to the template that you already applied, the system sends additional messages to the relevant module control processors and configures the hardware accordingly.

3. Ensure the configuration is correct:

   ```
   show qos config egress-queue-set egress-queue-set <id>
   ```

   ```
   config qos egress-queue-set <id> info
   ```

4. To configure the egress queue set queues, configure the egress queue set queues now, before you apply the egress queue set.
5. Apply the queue set:

   ```
   config qos egress-queue-set <id> apply
   ```

6. After all configurations are complete, restart the switch.

   ```
   boot
   ```

Variable definitions

Use the information in the following table to use the **`config qos egress-queue-set <id>`** command.

Variable	Value
`apply`	Applies the egress queue set when you issue the command. Otherwise, the operation is lost after you leave the current context. When you create an egress queue set, apply occurs when you issue the command. When you modify a queue set, apply occurs after you save the configuration and boot the switch.
`create qmax <value> [balanced-queues <value>] [hipri-queues <value>] [lopri-queues <value>] [name <value>]`	Specifies the maximum number of queues, either 8 or 64, as well as the number of balanced, high-priority, and low-priority queues in the egress queue set. The sum of the number of queues for balanced, high-priority (hipri), and low-priority (lopri) queues must be less than or equal to the qmax.
`delete`	Deletes the egress queue set.
`info`	Shows current queue set information.
`name <value>`	Modifies the name of the egress queue set template.

Use the information in the following table to use the **`config qos egress-queue-set <id> port`** command.

Variable	Value
`add <ports>`	Specifies the list of ports to add to the existing egress queue set template. Use this command to move a port from the default ADSSC setup to a different egress queue set. If you add ports to an applied template, the system sends additional messages to the relevant module control processors and configures the hardware accordingly.

Variable	Value
`info`	Shows information about a queue port configuration.
`remove <ports>`	Specifies the list of ports to remove from the existing egress queue set template. Removing ports from a specific egress queue set configures the ADSSC default appropriate for the port type. If you attempt to remove a port from the ADSSC default template, a warning message appears and the port stays with the default ADSSC.

Use the following table to use the **`show qos config egress-queue-set`** command.

Variable	Value
`all`	Displays all configured egress queue set data.
`egress-queue-set <id> [queues]`	Displays egress queue set data identified by name or specific ID.
`port <ports>`	Displays egress queue set data by port.

Example of configuring an egress queue set

Procedure steps

1. Configure the queue set:

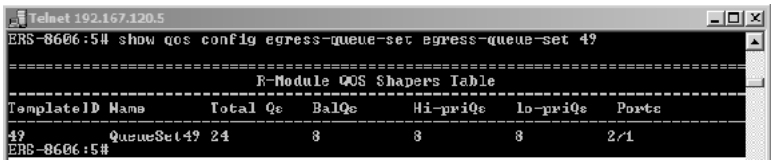
   ERS-8606:5# **config qos egress-queue-set 49 create qmax 64 balanced-queues 8 hipri-queues 8 lopri-queues 8 name QueueSet49**

2. Add ports:

   ERS-8606:5# **config qos egress-queue-set 49 port add 2/1**

3. Ensure the configuration is correct:

   ERS-8606:5# **show qos config egress-queue-set egress-queue-set 49**



4. Apply the queue set:

   ERS-8606:5# **config qos egress-queue-set 49 apply**

Comments? infodev@avaya.com

Job aid

The following table describes the headings in the show command output.

Table 23: egress queue set show command output

Field	Description
TemplateID	Template ID.
Name	Name of the queue set queue template.
Total Qs	Total number of all queues.
BalQs	Number of balanced queues.
Hi-priQs	Number of high-priority queues.
lo-priQs	Number of low-priority queues.
Ports	Specifies the ports associated with the queue.

Modifying an egress queue set

Configure an egress queue set to apply the same egress queue configuration (a template) to a group (set) of ports.

Important:
If you add or modify an egress queue set, you must restart the switch.

Procedure steps

1. Modify the egress queue set template:

   ```
   config qos egress-queue-set <id> create qmax <value>
   [balanced-queues <value>] [hipri-queues <value>] [lopri-
   queues <value>] [name <value>]
   ```

2. Modify associated ports with the egress queue set:

   ```
   config qos egress-queue-set <id> port add <port>
   ```

3. Ensure the configuration is correct:

   ```
   show qos config egress-queue-set egress-queue-set <id>
   ```

```
config qos egress-queue-set <id> info
```

4. To configure the egress queue set queues, do so now, before you apply the egress queue set.
5. Apply the queue set:

   ```
   config qos egress-queue-set <id> apply
   ```

   The following message appears:

   ```
   WARNING: The egress-queue-set QoS change made will take effect only after
   the configuration is saved and the chassis is rebooted.
   ```

6. Save the configuration as required:

   ```
   save config
   ```

   ```
   save config standby config.cfg
   ```

   ```
   save bootconfig
   ```

   ```
   save bootconfig standby boot.cfg
   ```

7. Restart the switch:

   ```
   boot -y
   ```

8. After the switch comes back online, ensure that the changes were made:

   ```
   config qos egress-queue-set <id> info
   ```

Variable definitions

Use the information in the following table to use the **`config qos egress-queue-set <id>`** command.

Variable	Value
`apply`	Applies the egress queue set. Apply occurs when you issue the command. Otherwise, the operation is lost after you leave the current context. When you create an egress queue set, apply occurs when you issue the command. When you modify a queue set, apply occurs after you save the configuration and boot the switch.
`create qmax <value> [balanced-queues <value>] [hipri-queues <value>] [lopri-queues`	Specifies the maximum number of queues, either 8 or 64, as well as the number of balanced, high-priority, and low-priority queues in the egress queue set. The sum of the number of queues for balanced, high-priority (hipri), and low-priority (lopri) queues must be less than or equal to the qmax.

Comments? infodev@avaya.com

Variable	Value
`<value>] [name <value>]`	
`delete`	Deletes the egress queue set.
`info`	Shows current queue set information.
`name <value>`	Modifies the name of the egress queue set template.

Use the information in the following table to use the **`config qos egress-queue-set <id> port`** command.

Variable	Value
`add <ports>`	Specifies the list of ports to add to the existing egress queue set template. Use this command to move a port from the default ADSSC setup to a different egress queue set. If you add ports to an applied template, the system sends additional messages to the relevant module control processors and configures the hardware accordingly.
`info`	Shows information about a queue port configuration.
`remove <ports>`	Specifies the list of ports to remove from the existing egress queue set template. Removing ports from a specific egress queue set configures the ADSSC default appropriate for the port type. If you attempt to remove a port from the ADSSC default template, a warning message appears and the port stays with the default ADSSC.

Configuring an egress queue set queue

Configure an egress queue to customize shaping behavior. Base queue-based shapers on egress queue set queues.

When you create a new custom queue, you MUST re-configure the default values provided for the new queue to suit customer QoS requirements.

Important:

For each Balanced queue, you must specify a desired minimum rate (min-rate) guarantee and a maximum-rate (max-rate) limit.

For Priority queues (either high or low priority), a minimum rate guarantee does not apply. Configure only a rate limit (max-rate).

The sum of minimum rate guarantees must be less than the port line rate minus the sum of high-priority queue rate limits. If this condition is not met, minimum rates are not guaranteed.

Important:
If you add or modify an egress queue set, you must restart the switch.

Prerequisites

- The egress queue set exists.

Procedure steps

1. Configure an egress queue set queue:

   ```
   config qos egress-queue-set <id> queue <qid> set [min-rate <value>] [max-rate <value>] [max-length <value>]
   ```

   This action removes the associated egress queue set. **<qid>** identifies the queue ID, from 1 to 386.

2. Ensure the configuration is correct:

   ```
   config qos egress-queue-set <id> queue <qid> info
   ```

   ```
   show qos config egress-queue-set egress-queue-set 49 queues
   ```

3. Apply the changes to the queue set:

   ```
   config qos egress-queue-set <id> apply
   ```

   If you modified an existing queue set, save the configuration, and then restart the switch.

Variable definitions

Use the information in the following table to use the **config qos egress-queue-set <id> queue <qid>** command.

Variable	Value
info	Shows information about a queue configuration.
name	Modifies the name of the egress queue.

Comments? infodev@avaya.com

Variable	Value
`set [min-rate <value>] [max-rate <value>] [max-length <value>]`	Configures the following options: • **`min-rate`** and **`max-rate`**—specify the line rate in percent to accommodate various port speeds in the same template. For example, if a 20 percent rate applies to a 10 and a 1 Gb/s port; the result is a 2 Gb/s bandwidth allocation for 10 Gb/s ports, and 200 Mb/s for 1 Gb/s ports. The min-rate minimum is 1 percent and the max-rate maximum is 100 percent. • **`max-length`**—you can specify the limit to which a queue can grow. The queue length does not imply that a queue has a fixed number of buffers. For example, a queue can grow to full memory size of 32 K buffers.

Example of configuring an egress queue set queue

Procedure steps

1. Configure the egress queue set queue:

   ```
   ERS-8606:5# config qos egress-queue-set 49 queue 3 set max-rate 70
   ```

2. Ensure the configuration is correct:

   ```
   ERS-8606:5# show qos config egress-queue-set egress-queue-set 49 queues
   ```

3. Apply the queue set:

   ```
   ERS-8606:5# config qos egress-queue-set 49 apply
   ```

4. Save the configuration:

   ```
   ERS-8606:5# save config
   ```

   ```
   ERS-8606:5# save bootconfig
   ```

5. Restart the switch:

   ```
   ERS-8606:5# reboot -y
   ```

6. After the switch comes back online, verify that the egress queue set applies and is correct:

   ```
   ERS-8606:5# config qos egress-queue-set 49 info
   ```

   ```
   ERS-8606:5# config qos egress-queue-set 49 queue 3 info
   ```

Job aid

The following table describes the headings in the show command output.

Table 24: egress queue set queue show command output

Field	Description
Qid	Queue offset from the base queue.
Q-name	Name of the queue.
Q-style	Queuing style: low priority, high priority, or balanced.
min-rate	Minimum guaranteed rate.
max-rate	Maximum data rate.
max-q-length	Maximum queue length.

Configuring ingress mappings

You can modify the ingress mappings to change traffic priorities. However, Avaya recommends that you use the default mappings.

Procedure steps

1. Configure MPLS to QoS ingress mappings:

   ```
   config qos ingressmap exp <exp> <level>
   ```

2. Configure DSCP to QoS ingress mappings:

   ```
   config qos ingressmap ds <dscp> <level>
   ```

3. Configure 802.1p bit to QoS ingress mappings:

   ```
   config qos ingressmap 1p <ieee1p> <level>
   ```

4. Ensure the configuration is correct:

   ```
   show qos ingressmap <1p|ds|exp> [<value>]
   ```


Comments? infodev@avaya.com

Variable definitions

Use the information in the following table to use the **`config qos ingressmap`** command.

Variable	Value
`1p <ieee1p> <level>`	Maps the IEEE 802.1p bit to QoS level. • <*level*> configures the QoS Level from 0–7. • <*ieee1p*> configures the IEEE 1P as an index from 0–7. Each QoS level has a default IEEE 1P value: • level 0—1 • level 1—0 • level 2—2 • level 3—3 • level 4—4 • level 5—5 • level 6—6 • level 7—7
`ds <dscp> <level>`	Maps the DS byte to QoS level. • <*level*> configures the QoS level from 0–7. • <*dscp*> configures the DiffServ Code Point (DSCP) as an index from 0–63.
`exp <exp> <level>`	Maps the MPLS EXP bit to a QoS level with a range from 0–7.
`info`	Displays information about the QoS ingress mappings.

Use the information in the following table to use the **`show qos ingressmap`** command.

Variable	Value
`1p [<ieee1p>]`	Shows the 802.1p bit to QoS ingress mappings.
`ds [<dscp>]`	Shows the DSCP to QoS ingress mappings.
`exp`	Shows the MPLS to QoS ingress mappings.

Configuring egress mappings

You can modify the egress mappings to change traffic priorities. However, Avaya recommends that you use the default mappings.

Procedure steps

1. Configure QoS to MPLS egress mappings:

   ```
   config qos egressmap exp <level> <exp>
   ```

2. Configure QoS to DSCP egress mappings:

   ```
   config qos egressmap ds <level> <dscp>
   ```

3. Configure QoS to 802.1p bit egress mappings:

   ```
   config qos egressmap 1p <level> <ieee1p>
   ```

4. Ensure the configuration is correct:

   ```
   show qos egressmap <1p|ds|exp> [<level>]
   show qos config eqmap <slot-number>
   ```

Variable definitions

Use the information in the following table to use the **`config qos egressmap`** command.

Variable	Value
`1p <level> <ieee1p>`	Maps the Qos level to IEEE 802.1p priority. • *<level>* configures the QoS level from 0–6. • *<ieee1p>* configures the IEEE 802.1p priority from 0–7. Each QoS level has a default IEEE 1P value: • level 0—1 • level 1—0 • level 2—2 • level 3—3 • level 4—4 • level 5—5

Comments? infodev@avaya.com

Variable	Value
	• level 6—6 • level 7—7
`ds <level> <dscp>`	Maps the QoS level to DS byte. • *<level>* configures the QoS level from 0–6. • *<dscp>* configures the DiffServ Code Point (DSCP) as an index from 0–63.
`exp <level> <exp>`	Maps the QoS level to MPLS EXP level. The range for each is 0–7.
`info`	Displays information about the QoS egress mappings.

Use the information in the following table to use the **`show qos egressmap`** command.

Variable	Value
`1p [<level>]`	Shows the QoS to 802.1p bit egress mappings.
`ds [<level>]`	Shows the QoS to DSCP egress mappings.
`exp`	Shows the QoS to MPLS egress mappings.

Configuring Avaya Automatic QoS

Configure the Avaya Automatic QoS to automatically recognize the DSCP values that Avaya voice applications use and to associate them with the proper egress queues.

Procedure steps

1. Enable diffserv on a port by using the following command:

   ```
   config ethernet <slot/port> enable-diffserv true
   ```
2. Enable a port as a trusted core port by using the following CLI command:

   ```
   config ethernet <slot/port> access-diffserv false
   ```
3. For tagged ports, enable 802.1p override by using the following command:

   ```
   config ethernet <slot/port> 802.1p-override enable
   ```

Comments? infodev@avaya.com

Chapter 12: Traffic filter configuration using the CLI

Use traffic filtering to block unwanted traffic or to prioritize desired traffic.

For information about statistics, see *Avaya Ethernet Routing Switch 8800/8600 Performance Management,* (NN46205-704).

Traffic filter configuration using the CLI procedures

This task flow shows you the sequence of procedures you perform to configure traffic filters.

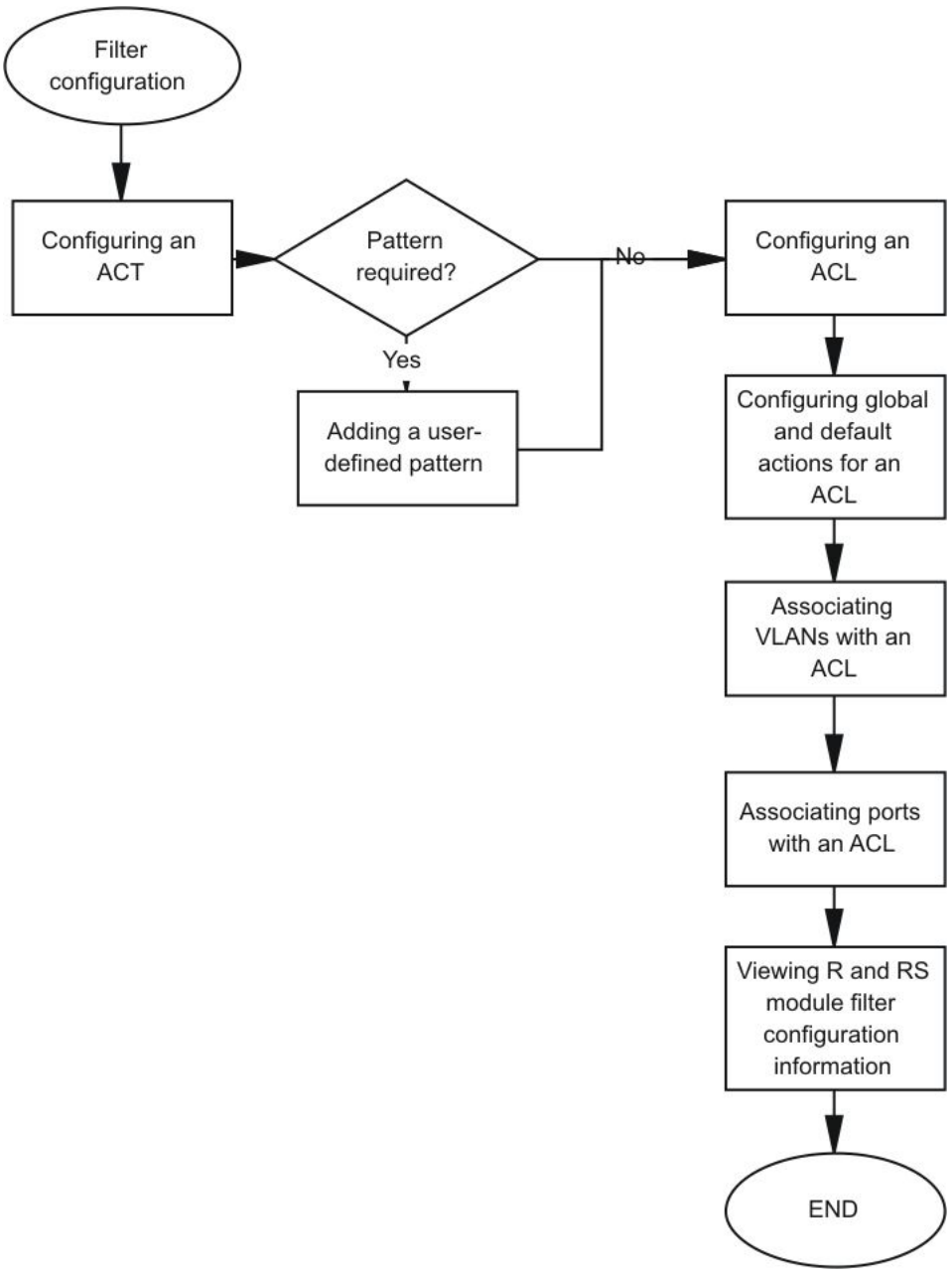


Figure 30: Traffic filter configuration using the CLI procedures

Job aid

The following roadmap lists traffic filter commands that you can use to perform the procedures in this section.

Table 25: Roadmap of traffic filter CLI commands

Command	Parameters
clear filter acl statistics default [<acl-id>]	—
clear filter acl statistics port [<acl-id>] [<acl-id> <ace-id>] [<acl-id> <ace-id> <port-num>]	—
config filter acl <acl-id>	create <type> act <value> [pktType <value>] [name <value>]
	delete
	disable
	enable
	info
	name <value>
config filter acl <acl-id> port	<ports>
	info
	remove <ports>
config filter acl <acl-id> set	default-action <value>
	global-action <value>
	info
config filter acl <acl-id> vlan	add <vid> [<vid2-vid3>]
	info
	remove <vid> [<vid2-vid3>]
config filter act <act-id>	apply
	arp <arp-attributes>
	create [name <value>]
	delete
	ethernet <ethernet-attributes>
	info
	ip <ip-attributes>
	ipv6 <ipv6-attributes>

Command	Parameters
	name <value>
	protocol <protocol-attributes>
config filter act <act-id> pattern <pattern-name>	add <base> <offset> <length>
	delete
	info
	modify <base> <offset> <length>
	name <pattern-name>
show filter acl ace [<acl-id>] [<ace-id>]	—
show filter acl action [<acl-id>] [<ace-id>]	—
show filter acl advanced [<acl-id>] [<ace-id>]	—
show filter acl arp [<acl-id>] [<ace-id>]	—
show filter acl config <acl-id>] [<ace-id>]	—
show filter acl debug [<acl-id>] [<ace-id>]	—
show filter acl ethernet [<acl-id>] [<ace-id>]	—
show filter acl info [<acl-id>]	—
show filter acl ip [<acl-id>] [<ace-id>]	—
show filter acl ipv6 [<acl-id>] [<ace-id>]	—
show filter acl protocol [<acl-id>] [<ace-id>]	—
show filter acl statistics default [<acl-id>]	—
show filter acl statistics port [<acl-id>] [<acl-id> <ace-id>] [<acl-id> <ace-id> <port-num>]	—

Comments? infodev@avaya.com

Command	Parameters
`show filter act [<act-id>]`	—
`show config module filter [verbose] [module <value>] [mode <value>]`	—
`show filter act-pattern [<act-id>]`	—

Configuring an ACT

Use an access control template (ACT) to specify all possible match fields for an access control list (ACL).

Prerequisites

- Add patterns before you activate the ACT (Apply = true).

Procedure steps

1. Create the ACT:

   ```
   config filter act <act-id> create [name <value>]
   ```

   `<act-id>` specifies an ACT ID from 1 to 4096.

2. Configure the required ACT attributes: ARP, IP, IPv6, protocol, and Ethernet. You can specify Access Control Entry (ACE) attributes only for the attributes that you specify in the ACT.
3. To add a pattern, you must do so before you activate the ACT.
4. Ensure the configuration is correct:

   ```
   show filter act [<act-id>]
   ```

5. Apply (commit) your changes:

   ```
   config filter act <act-id> apply
   ```

   After you issue the apply command, you can no longer modify the ACT. If you require different attributes or patterns, you must delete the ACT and create a new one.

Variable definitions

Use the information in the following table to use the **config filter act <act-id>** command.

Variable	Value
apply	Applies or commits the ACT. After you issue the **apply** command, you can change the ACT only by deleting it and creating a new one if no ACLs are associated with the ACT.
arp <arp-attributes>	Specifies the permitted ARP attributes for the ACT. Separate the list of allowed attributes by commas: • none • operation If you select none, this action deletes the node and prevents you from selecting other attributes.
create [name <value>]	Creates an ACT. The **name** <*value*> parameter is optional and specifies a descriptive name for the ACT using 0–32 characters. If you do not enter a name, the switch generates a default name. The ACT ID acts as an index to the ACT table. You can change the name at any time, even after you issue the **apply** command.
delete	Deletes an ACT if no associated ACLs exist.
ip <ip-attributes>	Specifies the permitted IP attributes for the ACT. You must separate the list of attributes commas. The list can include • none • srcIp, dstIp, ipFragFlag, ipOptions, ipProtoType, and dscp If you select none, this action deletes the node and prevents you from selecting other attributes. The default is none.
ethernet <ethernet-attributes>	Specifies the permitted Ethernet attributes for the ACT. You must separate the list of attributes commas. The list can include • none • srcMac, dstMac, etherType, <port|vlan>, and vlanTagPrio

Comments? infodev@avaya.com

Variable	Value
	If you select none, this action deletes the node and prevents you from selecting other attributes. The default is none.
`info`	Shows information about the ACTs.
`ipv6 <ipv6-attributes>`	Specifies the permitted IPv6 attributes. You must separate the list of attributes commas. The list can include • none • srcIpv6, dstIpv6, and nextHdr If you select none, this action deletes the node and prevents you from selecting other attributes. The default is none.
`name <value>`	Specifies a name for the ACT using 0–32 characters.
`protocol <protocol-attributes>`	Specifies the permitted protocol attributes for the ACT. You must separate the list of attributes commas. The list can include • none • tcpSrcPort, udpSrcPort, tcpDstPort, udpDstPort, tcpFlags, and icmpMsgFlags If you select none, this action deletes the node and prevents you from selecting other attributes. The default is none.

Adding a user-defined pattern

Add a user-defined pattern to which the ACT can match.

You can insert a pattern into an ACT only if it is inactive (not applied). An ACT can have a maximum of three associated patterns.

Prerequisites

- An ACT exists.
- You did not apply the ACT.

Procedure steps

1. Create a template for patterns within an ACT:

   ```
   config filter act <act-id> pattern <pattern-name> add <base>
   <offset> <length>
   ```

2. Ensure the configuration is correct:

   ```
   show filter act-pattern [<act-id>]
   ```

Variable definitions

Use the information in the following table to use the **`config filter act <act-id> pattern <pattern-name>`**command.

Variable	Value
`add <base> <offset> <length>`	Adds a template for patterns you create. <*base*>—the base and the offset together determine the beginning of the pattern. Permitted values for the base include • none • ether-begin, mac-dst-begin, mac-srcbegin, ethTypeLen-begin, arp-begin, ip-hdr-begin, ip-options-begin, ip-payload-begin, ip-tos-begin, ip-proto-begin, ip-src-begin, ip-dst-begin, ipv6-hdr-begin, tcp-begin, tcp-srcport-begin, tcp-dstport-begin, tcp-flags-end, udp-begin, udp-srcport-begin, udp-dstport-begin, ether-end, ip-hdr-end, icmp-msg-begin, tcp-end, and udp-end <*offset*> is the number of bits from the base where the pattern starts. <*length*> is the length in bits, from 1–56, of the user-defined field.
`delete`	Deletes access control template.
`info`	Displays information about the template patterns you created under an ACT.
`modify <base> <offset> <length>`	Modifies a template for user-defined patterns for this ACT ID. Options are the same as for the **`add`** command.
`name <pattern-name>`	Renames the pattern with a new name that you define. Each of the three patterns must have a unique name. <*pattern-name*> specifies a pattern name of up to 32 characters.

Comments? infodev@avaya.com

Configuring an ACL

Use an ACL to specify an ordered list of ACEs, or filter rules. The ACEs provide specific actions for the filter to perform.

When you create an ACL with the type inVlan that uses an ACT based on the source IP address, the ACL no longer works after the ARP aging time elapses. This does not cause a security breach. For a solution to this issue, see Workaround for inVlan, srcIp ACL on page 351 .

You cannot use an ACL to reference an ACT until you activate the ACT.

Prerequisites

- An ACT exists.
- You cannot use an ACL to reference an ACT until you apply the ACT.

Procedure steps

1. Configure an ACL :

   ```
   config filter acl <acl-id> create <type> act <value> [pktType <value>] [name <value>]
   ```

   <acl-id> specifies the unique identifier (from 1 to 4096) for the ACL.
2. Associate ports or VLANs to the ACL as required.
3. Configure the ACL actions as required.
4. Enable the ACL:

   ```
   config filter acl <acl-id> enable
   ```
5. Ensure the configuration is correct:

   ```
   show filter acl info [<acl-id>]
   ```

Variable definitions

Use the information in the following table to use the **config filter acl <acl-id>** command.

Variable	Value
`create <type> act <value> [pktType <value>] [name <value>]`	Creates an ACL only when you associate an ACT with that ACL. Options include • <*type*>—type of ACL: inVlan, outVlan, inPort, or outPort. • **`act`** <*value*>—an ACT ID from 1–4096. • **`pktType`** <*value*>—Layer 3 packet type (ipv4 or ipv6) • **`name`** <*value*>—an optional parameter that specifies a descriptive name for the ACL using 0–32 characters.
`delete`	Deletes an ACL. Removes all VLANs or brouter ports under this ACL and deletes all ACEs. It does not delete the ACTs.
`disable`	Disables the ACL state, and all associated ACEs.
`enable`	Enables the ACL state, and all associated ACEs. Enable is the default.
`info`	Displays information related to the ACL.
`name <value>`	Renames an ACL.

Configuring global and default actions for an ACL

Configure the default action to specify packet treatment when a packet does not match an ACE.

Configure the global action to specify packet treatment when a packet does match an ACE.

Prerequisites

- The ACL exists.

Procedure steps

1. Configure the global action for an ACL:

   ```
   config filter acl <acl-id> set global-action <value>
   ```

2. Configure the default action for an ACL:

   ```
   config filter acl <acl-id> set default-action <value>
   ```


Comments? infodev@avaya.com

Variable definitions

Use the information in the following table to use the **config filter acl <acl-id> set** command.

Variable	Value
default-action <value>	Specifies the default action to take when no ACEs match. Options include **<deny|permit>**. The default is permit.
global-action <value>	The <*value*> parameter specifies the global action for matching ACEs: • none • mirror, count, mirror-count, ipfix, mirror-ipfix, count-ipfix, and mirror-count-ipfix If you enable mirroring, ensure you specify the source or destination mirroring ports: • For R modules in Tx mode: use **config diag mirror-by-port** commands to specify mirroring ports. • For RS and 8800 modules, or R modules in Rx mode, use the **config filter acl <acl-id> ace <ace-id> debug** commands to specify mirroring ports.
info	Displays the status of the global and default actions.

Associating VLANs with an ACL

Associate VLANs with, or remove VLANs from, an ACL so that filters apply or do not apply to VLAN traffic, respectively.

Prerequisites

- The ACL exists.
- The VLANs exist.

Procedure steps

1. Associate VLANs with an ACL:

   ```
   config filter acl <acl-id> vlan add <vid> [<vid2-vid3>]
   ```

2. Remove VLANs from an ACL:

   ```
   config filter acl <acl-id> vlan remove <vid> [<vid2-vid3>]
   ```

Variable definitions

Use the information in the following table to use the **`config filter acl <acl-id> vlan`** command.

Variable	Value
`add <vid> [<vid2-vid3>]`	Associates a VLAN or a VLAN list with an ACL. The *<vid>* parameter is a list of VLANs separated by a comma, or a range of VLANs specified from low to high [vlan-id - vlan-id].
`info`	Displays the ACL VLAN status.
`remove <vid> [<vid2-vid3>]`	Removes a VLAN or VLAN list from an ACL. The *<vid>* parameter is a list of VLANs separated by a comma, or a range of VLANs specified from low to high [vlan-id to vlan-id].

Associating ports with an ACL

Associate ports with, or remove ports from, an ACL so that filters do or do not apply to port traffic, respectively.

Comments? infodev@avaya.com

Prerequisites

- The ACL exists.

Procedure steps

1. Associate ports with an ACL:

   ```
   config filter acl <acl-id> port add <ports>
   ```

2. Remove ports from an ACL:

   ```
   config filter acl <acl-id> port remove <ports>
   ```

Variable definitions

Use the information in the following table to use the **`config filter acl <acl-id> port`** command.

Variable	Value
`add <ports>`	Associates a port or a port list with an ACL. The *<ports>* parameter is a list of ports in the following format: [<slot/port>] or [<slot/port-slot/port>].
`remove <ports>`	Removes a port or a port list from an ACL. The *<ports>* parameter is a list of ports in the following format: [<slot/port>] or [<slot/port-slot/port>].
`info`	Displays the ACL port status.

Viewing filter configuration information

You can view configuration information for ACL-based filters.

Procedure steps

1. View configuration information about filters:

```
show config module filter [verbose] [mode <value>]
```

Variable definitions

Use the information in the following table to use the **show** command.

Variable	Value
`mode <value>`	Shows filter configuration output in either CLI or ACLI mode. **`<value>`** is cli or acli.
`verbose`	Shows detailed output.

Job aid

This section shows the **`show config module filter`** command output.

```
ERS-8606:5# show config module filter
Preparing to Display Configuration... #
# MON APR 14 11:05:31 2008 UTC
# box type : ERS-8006
# software version : REL4.2.0.0_B157
# monitor version : 4.2.0.0/157
# cli mode : 8600 CLI
#
#
# Asic Info :
# SlotNum|Name |CardType |MdaType |Parts Description
#
# Slot 1 -- 0x00000001 0x00000000
# Slot 2 -- 0x00000001 0x00000000
# Slot 3 8630GBR 0x2432511e 0x00000000 RSP=25 CLUE=2 F2I=1 F2E=1
FTMUX=17 CC= 3 FOQ=266 DPC=184 BMC=776 PIM=257 MAC=4
# Slot 4 8648GTR 0x24220130 0x00000000 RSP=25 CLUE=2 F2I=1 F2E=1
FTMUX=0 CC=3 FOQ=266 DPC=6 BMC=776 PIM=257 MAC=4
# Slot 5 8692SF 0x200e0100 0x00000000 CPU: CPLD=19 MEZZ=4 SFM:
OP=3 TMUX=2 SWIP=23 FAD=16 CF=28
# Slot 6 -- 0x00000001 0x00000000 config
#
# R-MODULE FILTER CONFIGURATION
#
filter act 1 create name "ACT-1ADV"
filter act 1 ethernet srcMac
filter act 1 ip srcIp
filter act 1 protocol tcpSrcPort
filter act 1 apply filter act 2 create name "ACT-2AD VS"
filter act 2 pattern kelie add ip-hdr-begin 0 1
filter act 2 apply
filter acl 1 create inPort act 1
filter acl 1 set global-action mirror-count
filter acl 1 ace 1 create name "Adv"
filter acl 1 ace 1 action permit filter acl 1 ace 1 debug
copytoprimarycp enable
filter acl 2 create inPort act 2
filter acl 2 ace 1 create name "KB"
filter acl 2 ace 1 action permit remark-dot1p five
back
ERS-8606:5#
```

Chapter 13: Access control entry configuration using the CLI

An access control entry (ACE) comprises an ordered list of traffic filtering rules.

Job aid

The following roadmap lists traffic filter commands that you can use to perform the procedures in this section.

Table 26: Roadmap of traffic filter CLI commands

Command	Parameters
`clear filter acl statistics port [<acl-id>] [<acl-id> <ace-id>] [<acl-id> <ace-id> <port-num>]`	–
`config filter acl <acl-id> ace <ace-id>`	`action <mode> [mlt-index <value>] [remark-dscp <value>] [remark-dot1p <value>] [police <value>] [redirect-next-hop <value>] [unreachable <value>] [egress-queue <value>] [stop-on-match <value>] [egress-queue-adssc <value>] [ipfix <value>]`
	`create [name <value>]`
	`debug [count <value>] [copytoprimarycp <value>] [copytosecondarycp <value>] [mirror <value>] [mirroring-dst-ports <value>] [mirroring-dst-vlan <value>] [mirroring-dst-mlt <value>]`
	`delete`
	`disable`

Command	Parameters
	enable
	info
	name <value>
config filter acl <acl-id> ace <ace-id> advanced	custom-filter1 <pattern1-name> <ace-op> <value>
	custom-filter2 <pattern2-name> <ace-op> <value>
	custom-filter3 <pattern3-name> <ace-op> <value>
	delete <pattern-attributes>
	info
config filter acl <acl-id> ace <ace-id> arp	delete <arp-attributes>
	info
	operation <ace-op> <arp-oper-type>
config filter acl <acl-id> ace <ace-id> ethernet	delete <ethernet-attributes>
	dst-mac <ace-op> <dst-mac-list>
	ether-type <ace-op> <ether-type>
	info
	port <ace-op> <ports>
	src-mac <ace-op> <src-mac-list>
	vlan-id <ace-op> <vid>
	vlan-tag-prio <ace-op> <vlan-tag-prio>
config filter acl <acl-id> ace <ace-id> ip	delete <ip-attributes>
	dscp <ace-op> <dscp-list>
	dst-ip <ace-op> <dst-ip-list>
	info
	ip-frag-flag <ace-op> <ip-frag-flag>
	ip-options <ace-op>

Comments? infodev@avaya.com

Command	Parameters
	ip-protocol-type <ace-op> <ip-protocol-type>
	src-ip <ace-op> <src-ip-list>
config filter acl <acl-id> ace <ace-id> ipv6	delete <ipv6-attributes>
	dst-ipv6 <ace-op> <dst-ipv6-list>
	info
	src-ipv6 <ace-op> <src-ipv6-list>
	nxt-hdr <ace-op> <nxt-hdr>
config filter acl <acl-id> ace <ace-id> protocol	delete <protocol-attributes>
	icmp-msg-type <ace-op> <icmp-msg-type>
	info
	tcp-dst-port <ace-op> <tcp-portlist>
	tcp-flags <ace-op> <tcp-flags>
	tcp-src-port <ace-op> <tcp-portlist>
	udp-dst-port <ace-op> <udp-portlist>
	udp-src-port <ace-op> <udp-portlist>
config filter acl <acl-id> ace <ace-id> remove-mirror-dst	mirroring-dst-ports <port>
	mirroring-dst-vlan <vid>
	mirroring-dst-mlt <mid>
show filter acl ace [<acl-id>] [<ace-id>]	–
show filter acl action [<acl-id>] [<ace-id>]	–
show filter acl advanced [<acl-id>] [<ace-id>]	–
show filter acl arp [<acl-id>] [<ace-id>]	–

Command	Parameters
`show filter acl config <acl-id>] [<ace-id>]`	–
`show filter acl debug [<acl-id>] [<ace-id>]`	–
`show filter acl ethernet [<acl-id>] [<ace-id>]`	–
`show filter acl ip [<acl-id>] [<ace-id>]`	–
`show filter acl ipv6 [<acl-id>] [<ace-id>]`	–
`show filter acl protocol [<acl-id>] [<ace-id>]`	–
`show filter acl statistics port [<acl-id>] [<acl-id> <ace-id>] [<acl-id> <ace-id> <port-num>]`	–

Configuring ACEs

Use an access control entry (ACE) to define a packet pattern and the desired behavior for packets that carry the pattern.

ACEs of type inVlan with an ACT that includes srcIp, and with an ACL default action of deny, require additional configuration to function properly. See Workaround for inVlan, srcIp ACL on page 351 for the CLI commands for this special configuration.

Alternatively, Avaya recommends that you create ACLs with a default action of permit, and with an ACE mode of deny. For deny and permit ACLs and ACEs, the default action and the mode must be opposite for the ACE (filter) to have meaning.

Prerequisites

- The ACL exists.

Procedure steps

1. Create an ACE:

   ```
   config filter acl <acl-id> ace <ace-id> create [name <value>]
   ```

2. Configure the action mode as deny or permit:

   ```
   config filter acl <acl-id> ace <ace-id> action <deny|permit>
   ```

3. Configure actions as required.
4. Ensure the configuration is correct:

   ```
   show filter acl ace [<acl-id>] [<ace-id>]
   ```

5. Enable the ACE:

   ```
   config filter acl <acl-id> ace <ace-id> enable
   ```

Variable definitions

Use the information in the following table to use the **`config filter acl <acl-id> ace <ace-id>`** commands.

Variable	Value
`action <deny|permit>`	Updates desired action parameters for the ACE.
`create [name <value>]`	Creates an Access Control Entry (ACE). The ACE ID determines precedence (that is, the lower the ID, the higher the precedence). The **`name`** <*value*> parameter is optional and specifies a descriptive name for the ACE using 0–32 characters. You can modify ACE attributes only after you disable the ACE. If you issue the same command several times, the new values overwrite the previous command. For example, if you enter the following commands the values you enter with the third command overwrite the first command: `config filter acl acl-2 ace ace-3 ip src-ip eq 1.1.1.1`

Variable	Value
	`config filter ac acl-2 ace-3 ip dst-ip eq 5.5.5.5` `config filter acl acl-2 ace ace-3 ip src-ip eq 7.7.7.7`
`debug`	Updates desired debug parameters for access control entry.
`delete`	Deletes an ACE.
`disable`	Disables an ACE within an ACL. The default is disable.
`enable`	Enables an ACE within an ACL. After you enable an ACE, if you need to make changes, you must first disable it.
`info`	Displays information related to the ACE.
`name <value>`	Renames an ACE using a descriptive name from 0–32 characters.

Configuring ACE actions

Actions determine the process that occurs when a packet matches an ACE.

Prerequisites

- The ACL exists.
- The ACE exists.

Procedure steps

1. Configure ACE actions:

   ```
   config filter acl <acl-id> ace <ace-id> action <deny|permit>
   [mlt-index <value>] [remark-dscp <value>] [remark-dot1p
   <value>] [police <value>] [redirect-next-hop <value>]
   [unreachable <value>] [egress-queue <value>] [stop-on-match
   <value>] [egress-queue-adssc <value>] [ipfix <value>]
   ```

2. Ensure the configuration is correct:

   ```
   show filter acl action [<acl-id>] [<ace-id>]
   ```


Comments? infodev@avaya.com

Variable definitions

Use the information in the following table to use the **`config filter acl <acl-id> ace <ace-id> action <deny|permit>`** command.

Variable	Value
`egress-queue <value>`	Specifies the offset from the base queue number (0–63). **`<value>`** can be one, two, or three values. The first value specifies the Egress Queue ID for the 8648GTR, 8648GTRS, 8848GT, 8648GBRS, 8848GB, and gigabit ports of the 8634XGRS and 8834XG modules. The second value specifies the Egress Queue ID for the 8630GBR, 8612XRS, 8812XL, and 10 Gb ports of the 8634XGRS and the 8834XG modules. The third specifies the Egress Queue ID for 8683XLR and 8683XZR modules. If you specify only one value, the same value applies to all module types. If you specify two values, the first value applies to 8648GTR, 8848GT, 8648GTRS, 8648GBRS, 8848GB, and gigabit ports of 8634XGRS, and 8834XG, and the second value applies to 8630GBR, 8612XLRS, 8812XL, and 10 Gb ports of 8634XGRS and 8834XG modules. If you specify all three values, the three values apply to the respective module types as explained in the preceding paragraph.
`egress-queue-adssc <value>`	Specifies the ACE ADSSC egress queue value as one of the following: • disable • critical, network, premium, platinum, gold, silver, bronze, or standard The default is disable.
`ipfix <enable|disable>`	Enables or disables IPFIX. The default is disable.
`mlt-index <index>`	Overrides the mlt-index chosen by the MLT algorithm for packets sent on MLT ports. The MLT index varies from 0–8. If three ports exist in an MLT (for example, A, B, and C) and you specify an index of 6, the Avaya Ethernet Routing Switch 8800/8600 applies the MOD function and chooses port C. If port C becomes nonoperational, the filtered packets exit from port B. Multicast traffic does not support the MLT index.
`police <value>`	Specifies the policy ID of a policer (0–16383). A policy must already exist.

Variable	Value
`redirect-next-hop <value>`	Specifies the next-hop IP address for redirect mode (a.b.c.d). If you specify a next-hop IPv6 address for redirect mode, enter 0.0.0.0 <IPv6 address>.
`remark-dot1p <value>`	Specifies the new 802.1 priority bit for matching packets: • disable • zero, one, two, three, four, five, six, or seven The default is disable.
`remark-dscp <value>`	Specifies the new Per-Hop Behavior for matching packets: • disable • phbcs0, phbcs1, phbaf11, phbaf12, phbaf13, phbcs2, phbaf21, phbaf22, phbaf23, phbcs3, phbaf31, phbaf32, phbaf33, phbcs4, phbaf41, phbaf42, phbaf43, phbcs5, phbef, phbcs6, and phbcs7 The default is disable.
`stop-on-match <true|false>`	Enables or disables the stop-on-match option. This option specifies whether to stop or continue after an ACE matches the packet. After this ACE matches, the switch does not attempt a match on other ACEs with lower priority. The default is false.
`unreachable <deny|permit>`	Denies or permits packet dropping when the next hop is unreachable. The default is deny.

Configuring ACE debug actions

Use debug actions to use filters for troubleshooting or traffic monitoring.

Caution:

Risk of packet loss

Avaya recommends that you do not select copyToPrimaryCp or copyToSecondaryCp. If you select the copyToPrimaryCp parameter, the switch sends packets to the CP, which can overload it. You can use the Packet Capture Tool (PCAP), rather than using copyToPrimaryCp.

Prerequisites

- The ACL exists.
- The ACE exists.

Procedure steps

1. Configure debug actions for an ACE:

   ```
   config filter acl <acl-id> ace <ace-id> debug [count <value>]
   [copytoprimarycp <value>] [copytosecondarycp <value>]
   [mirror <value>] [mirroring-dst-ports <value>] [mirroring-
   dst-vlan <value>] [mirroring-dst-mlt <value>]
   ```

2. Ensure the configuration is correct:

   ```
   show filter acl debug [<acl-id>] [<ace-id>]
   ```

Variable definitions

Use the information in the following table to use the **`config filter acl <acl-id> ace <ace-id> debug`** command.

Variable	Value
`count <enable|disable>`	Enables or disables counting after a packet matching the ACE is found. The default is disable.
`copytoprimarycp <enable|disable>`	Enables or disables the ability to copy matching packets to the primary (Master) CPU. The default is disable.
`copytosecondarycp <enable|disable>`	Enables or disables the ability to copy matching packets to the secondary (Standby) CPU. The default is disable.
`mirror <enable|disable>`	Enables or disables mirroring for the ACE. If you enable mirroring, ensure that you configure the appropriate parameters: • For R, RS and 8800 modules in Rx mode, and for RS and 8800 modules, use **`mirroring-dst-ports`**,

Variable	Value
	mirroring-dst-vlan, or **mirroring-dst-mlt**. • For R modules in Tx mode, use the **config diag mirror-by-port** commands to specify the mirroring source or destination. The default is disable.
mirroring-dst-ports <value>	Specifies the destination port or ports for mirroring.
mirroring-dst-vlan <value>	Specifies the destination VLAN for mirroring.
mirroring-dst-mlt <value>	Specifies the destination MLT group for mirroring.

Example of configuring R module TxFilter mode mirroring

This configuration sends mirrored ICMP packets from port 2/1 to port 4/1.

1. Configure ACT 3:

   ```
   ERS8610:5# config filter act 3 create
   ERS8610:5# config filter act 3 ipProtoType
   ERS8610:5# config filter act 3 apply
   ```

2. Configure an outVLAN ACL that uses ACT 3 and VLAN 2:

   ```
   ERS8610:5# config filter acl 21 create outVlan act 3
   ERS8610:5# config filter acl 21 vlan add 2
   ```

3. Add ACE 21 with action of permit to mirror ICMP traffic:

   ```
   ERS8610:5# config filter acl 21 ace 1 create name icmp
   ERS8610:5# config filter acl 21 ace 1 action permit
   ERS8610:5# config filter acl 21 ace 1 ip ip-protocol-type eq icmp
   ERS8610:5# config filter acl 21 ace 1 debug mirror enable
   ERS8610:5# config filter acl 21 ace 1 enable ERS8610:5#
   ```

4. Because this is an R module in txFilter mode, configure the mirroring source and destination ports:

Comments? infodev@avaya.com

```
ERS8610:5# config diag mirror-by-port 1 create in-port 1/1
out-port 3/1 mode txFilter enable true
```

Configuring ARP ACEs

Use ACE ARP entries to have the filter look for ARP requests or responses.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has ARP attributes.

Procedure steps

1. To configure an ACE for ARP packets:

   ```
   config filter acl <acl-id> ace <ace-id> arp operation <ace-
   op> <arp-oper-type>
   ```

2. Ensure the configuration is correct:

   ```
   show filter acl arp [<acl-id>] [<ace-id>]
   ```

Variable definitions

Use the following table to use the **`config filter acl <acl-id> ace <ace-id> arp`** command.

Variable	Value
`delete <arp-attributes>`	Deletes ARP attributes.
`info`	Displays ARP status information for the ACE.
`operation <ace-op> <arp-oper-type>`	Specifies the following:

Variable	Value
	• **<ace-op>** specifies an operator for a field match operation (eq). • **<arp-oper-type>** specifies an operation type: arpRequest or arpResponse. For ARP, only one attribute exists—operation.

Configuring an Ethernet ACE

Use Ethernet ACEs to filter on Ethernet parameters.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has Ethernet attributes.
- You can select a port or a VLAN ID, but not both.

Procedure steps

1. Configure an ACE with Ethernet header attributes:

   ```
   config filter acl <acl-id> ace <ace-id> ethernet
   ```

2. Ensure the configuration is correct:

   ```
   show filter acl ethernet [<acl-id>] [<ace-id>]
   ```

Variable definitions

Use the following table to help you use the **config filter acl <acl-id> ace <ace-id> ethernet** command.

Comments? infodev@avaya.com

Variable	Value
`delete <ethernet-attributes>`	Specifies Ethernet ACE attributes to delete. The <*ethernet-attributes*> parameter is a list of Ethernet attributes {<attr>,<attr>,<attr>-} where attr is • none • srcMac, dstMac, etherType, <port|vlan>, or vlanTagPrio You cannot select other attributes if you select **none**.
`dst-mac <ace-op> <dst-mac-list>`	The **<ace-op>** parameter specifies an operator for a field match condition: eq, ne, le, ge. The **<dst-mac-list>** parameter specifies a list of destination MAC addresses separated by a comma, or a range of MAC addresses specified from low to high; for example, [a:b:c:d:e:f, (x:y:z:w:v:u- a:b:c:d:e:f)]. You cannot use an asterisk (*) after <*ace-op*>.
`ether-type <ace-op> <ether-type>`	The **<ace-op>** parameter specifies an operator for a field match condition: eq, ne. The **<ether-type>** parameter specifies an ether-type name or number: • 0–65563 • ip, arp, ipx802dot3, ipx802dot2, ipxSnap, ipxEthernet2, appleTalk, decLat, decOther, sna802dot2, snaEthernet2, netBios, xns, vines, ipv6, rarp, or PPPoE.
`info`	Displays Ethernet header status information for the ACE.
`port <ace-op> <ports>`	The **<ace-op>** parameter specifies an operator for a field match condition (eq). The **<ports>** parameter specifies a port list [slot/port].
`src-mac <ace-op> <src-mac-list>`	The **<ace-op>** parameter specifies an operator for a field match condition: eq, ne, le, ge. The **<src-mac-list>** parameter specifies a list of source MAC addresses separated by a comma, or a range of MAC addresses specified from low to high; for example, [a:b:c:d:e:f, (x:y:z:w:v:u- a:b:c:d:e:f)].
`vlan-id <ace-op> <vid>`	The **<ace-op>** parameter specifies an operator for a field match condition (eq). The **<vid>** parameter specifies a list of VLAN IDs from 0–4096.
`vlan-tag-prio <ace-op> <vlan-tag-prio>`	The **<ace-op>** parameter specifies an operator for a field match condition: eq, ne. The **<vlan-tag-prio>** parameter specifies a VLAN tag priority from 0–7 or undefined.

Example of configuring an Ethernet ACE

1. Specify a specific destination MAC address:

   ```
   ERS-8610:6# config filter acl 1 ace 12 ethernet dst-mac eq
   08:00:69:02:01:FC
   ```

Configuring an IP ACE

Use IP ACEs to filter on the source IP address, destination IP address, DiffServ Code Point (DSCP), protocol, IP options, and IP fragmentation parameters.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has IP attributes.

Procedure steps

1. Configure an ACE with IP header attributes:

   ```
   config filter acl <acl-id> ace <ace-id> ip
   ```

2. Ensure the configuration is correct:

   ```
   show filter acl ip [<acl-id>] [<ace-id>]
   ```

Variable definitions

Use the following table to help you use the **`config filter acl <acl-id> ace <ace-id> ip`** command.

Variable	Value
`delete <ip-attributes>`	Specifies a list of IP ACE attributes to delete:

Comments? infodev@avaya.com

Variable	Value
	• none • srcIp, dstIp, ipFragFlag, ipOptions, ipProtoType, or dscp You cannot select other attributes if you select none.
`dst-ip <ace-op> <dst-ip-list>`	The **`<ace-op>`** parameter specifies an operator for a field match condition: eq, ne, le, ge. The **`<dst-ip-list>`** parameter specifies the destination IP address list in one of the following format: a.b.c.d, [w.x.y.z-p.q.r.s], [l.m.n.o/mask], [a.b.c.d/len]. You cannot use an asterisk (*) after <*ace-op*>.
`dscp <ace-op> <dscp-list>`	The **`<ace-op>`** parameter specifies an operator for a field match condition: eq, ne. **`<dscp-list>`** specifies the PHB: • disable • phbcs0, phbcs1, phbaf11, phbaf12, phbaf13, phbcs2, phbaf21, phbaf22, phbaf23, phbcs3, phbaf31, phbaf32, phbaf33, phbcs4, phbaf41, phbaf42, phbaf43, phbcs5, phbcs6, phbef, or phbcs
`ip-frag-flag <ace-op> <ip-frag-flag>`	The **`<ace-op>`** parameter specifies an operator for a field match condition: eq, ne. The **`<ip-frag-flag>`** parameter specifies a match option for IP fragments (0, 2, 4), or noFragment, moreFragment, lastFragment, anyFragment.
`ip-options <ace-op>`	Specifies an operator for a field match condition (any is the only option).
`info`	Displays IP header status information for the ACE.
`ip-protocol-type <ace-op> <ip-protocol-type>`	The **`<ace-op>`** parameter specifies an operator for a field match condition: eq, ne. The **`<ip-protocol-type>`** parameter specifies one or more IP protocol types: (1–256), or undefined, icmp, tcp, udp, ipsecesp, ipsecah, ospf, vrrp, snmp.
`src-ip <ace-op> <src-ip-list>`	The **`<ace-op>`** parameter specifies an operator for a field match condition: eq, ne, le, ge. The **`<src-ip-list>`** parameter specifies a source IP address list in one of the following format: a.b.c.d, [w.x.y.z-p.q.r.s], [l.m.n.o/mask], [a.b.c.d/len].

Example of configuring an IP ACE

1. Specify a destination IP address:

```
ERS-8610:6# config filter acl 1 ace 12 ip dst-ip eq 131.205.3.4
```

Configuring a protocol ACE

Use protocol ACEs to filter on the TCP source port, UDP source port, TCP destination port, UDP destination port, ICMP message type, and TCP flags.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has protocol attributes.

Procedure steps

1. Configure an ACE with protocol attributes:

   ```
   config filter acl <acl-id> ace <ace-id> protocol
   ```

   The tcp-flags and icmp-msg-type command options support lists.
2. Ensure the configuration is correct:

   ```
   show filter acl protocol [<acl-id>] [<ace-id>]
   ```

Variable definitions

Use the information in the following table to use the **`config filter acl <acl-id> ace <ace-id> protocol`** command.

Variable	Value
`delete <protocol-attributes>`	Specifies protocol ACE attributes to delete • none • tcpSrcPort, udpSrcPort ,tcpDstPort, udpDstPort, tcpFlags, or icmpMsgType You cannot select other attributes if you select none .
`icmp-msg-type <ace-op> <icmp-msg-type>`	The **`<ace-op>`** parameter specifies an operator for a field match condition: eq, ne.

Comments? infodev@avaya.com

Variable	Value
	The **<icmp-msg-type>** parameter specifies one or more IP protocol types (0–255), or echoreply, destunreach, sourcequench, redirect, echo-request, routeradv, routerselect, time-exceeded, param-problem, timestamp-request, timestamp-reply, addressmask-request, addressmask-reply, or traceroute. You cannot select an asterisk (*) after *<ace-op>*.
info	Displays IP header status information for the ACE.
tcp-dst-port <ace-op> <tcp-portlist>	The **<ace-op>** parameter specifies an operator for a field match condition: eq, ne, le, ge. The default is eq (equals). The **<tcp-portlist>** parameter specifies the destination port for the TCP protocol: (0–65535), or echo, ftpdata, ftpcontrol, ssh, telnet, dns, http, bgp, hdot323, or undefined.
tcp-flags <ace-op> <tcp-flags>	The **<ace-op>** parameter specifies an operator for a field match condition: matchAny, matchAll **<tcp-flags>** specifies one or more TCP flags: none, fin, syn, rst, push, ack, urg, or undefined.
tcp-src-port <ace-op> <tcp-portlist>	The **<ace-op>** parameter specifies an operator for a field match condition: eq, ne, le, ge. The default is eq (equals). The **<tcp-portlist>** parameter specifies the destination port for the TCP protocol (0–65535), or echo, dns, bootpServer, bootpClient, tftp, rip, rtp, rtcp, or undefined.
udp-dst-port <ace-op> <udp-portlist>	The **<ace-op>** parameter specifies an operator for a field match condition: eq, ne, le, ge. The default is eq. The **<udp-portlist>** parameter specifies the destination port for the UDP protocol (0–65535), or echo, dns, bootpServer, bootpClient, tftp, rip, rtp, rtcp, or undefined.
udp-src-port <ace-op> <udp-portlist>	The **<ace-op>** parameter specifies an operator for a field match condition: eq, ne, le, ge. The default is eq. The **<udp-portlist>** parameter specifies the source port for the UDP protocol (0–65535), or echo, dns, bootpServer, bootpClient, tftp, rip, rtp, rtcp, or undefined.

Example of configuring a protocol ACE

1. Specify ICMP packets:

```
ERS-8610:6# config filter acl 1 ace 12 protocol icmp-msg-type
eq destunreach
```

Configuring a custom ACE

You can use a custom ACE to define your own match patterns.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has pattern attributes.

Procedure steps

1. Add an ACE for patterns that you define:

   ```
   config filter acl <acl-id> ace <ace-id> advanced
   ```

2. Ensure that your configuration is correct:

   ```
   show filter acl advanced [<acl-id>] [<ace-id>]
   ```

Variable definitions

Use the following table to use the **`config filter acl <acl-id> ace <ace-id> advanced`** command.

Variable	Value
`custom-filter1 <pattern1-name> <ace-op> <value>`	Specifies the following information for custom filter 1: • <*pattern1-name*>—a descriptive name for pattern 1 that uses 0–32 characters. • <*ace-op*>—an operator for a field match condition (eq, le, ge). The ace-op ne does not apply to an ACE pattern. • <*value*>—a hexadecimal number equal to the pattern template length.

Comments? infodev@avaya.com

Variable	Value
`custom-filter2 <pattern2-name> <ace-op> <value>`	Specifies the following information for custom filter 2: • <*pattern2-name*>—a descriptive name for pattern 2 that uses 0–32 characters. • <*ace-op*>—an operator for a field match condition (eq, le, ge). The ace-op ne does not apply to an ACE pattern. • <*value*>—a hexadecimal number equal to the pattern template length.
`custom-filter3 <pattern3-name> <ace-op> <value>`	Specifies the following information for custom filter 3: • <*pattern3-name*>—a descriptive name for pattern 3 that uses 0–32 characters. • <*ace-op*>—an operator for a field match condition (eq, le, ge). The ace-op ne does not apply to an ACE pattern. • <*value*>—a hexadecimal number equal to the pattern template length.
`delete <pattern-attributes>`	Deletes user-defined patterns for an ACE: • none • custom-filter1, custom-filter2, custom-filter3
`info`	Displays user-defined pattern status information for the ACE.

Example of configuring a custom ACE

1. Add an ACE for patterns that you define:

   ```
   ERS-8610:6# config filter acl 1 ace 12 advanced custom-filter1
   Pattern1 eq 0x12
   ```

Configuring an IPv6 ACE

Use an IPv6 ACE to filter on IPv6 attributes.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has IPv6 attributes.

Procedure steps

1. Add an ACE with IP header attributes:

   ```
   config filter acl <acl-id> ace <ace-id> ipv6
   ```

2. Ensure that your configuration is correct:

   ```
   show filter acl ipv6 [<acl-id>] [<ace-id>]
   ```

Variable definitions

Use the information in the following table to use the **`config filter acl <acl-id> ace <ace-id> ipv6`** command.

Variable	Value
`delete <ipv6-attributes>`	Deletes the specified IPv6 ACE attributes. You cannot select other attributes if you select none.
`dst-ipv6 <ace-op> <dst-ipv6-list>`	The **`<ace-op>`** parameter specifies an operator for a field match condition: eq, ne. The **`<dst-ipv6-list>`** parameter specifies the list of destination IPv6 addresses, separated by commas. You cannot select an asterisk (*) after *<ace-op>*.
`info`	Displays the current level parameter settings and the next level directories.
`nxt-hdr <ace-op> <nxt-hdr>`	The **`<ace-op>`** parameter specifies an operator for a field match condition: eq, ne. The **`<nxt-hdr>`** parameter specifies hop-by-hop, tcp, udp, routing, fragment, ipsecesp, ipsecah, icmpv6, noHdr, or undefined.
`src-ipv6 <ace-op> <src-ipv6-list>`	The **`<ace-op>`** parameter specifies an operator for a field match condition: eq, ne. The **`<src-ipv6-list>`** parameter specifies the list of source IPv6 addresses, separated by commas.

Viewing ACL and ACE configuration data

Review your configuration to ensure that it is correct.

Procedure steps

1. View a list of executed commands:

   ```
   show filter acl config [<acl-id>] [<ace-id>]
   ```

Variable definitions

Use the information in the following table to use the **`show filter acl config`** command.

Variable	Value
`<ace-id>`	Specifies an ACE ID from 1–1000.
`<acl-id>`	Specifies an ACL ID from 1–4096.

Comments? infodev@avaya.com

Chapter 14: CLI configuration examples

This section provides configuration examples for common Quality of Service (QoS) and filtering tasks and includes the command line interface (CLI) commands you use to create the sample configurations.

For more information, see the configuration examples in *Filters and QoS for ERS 8800/8600 R-Series Modules Technical Configuration Guide, NN48500-541*. You can find this Technical Configuration Guide at http://www.avaya.com/supportwith the rest of the ERS8800/8600 documentation.

Delivering subrate IP service using policy-based policers

The example that follows shows how to provision subrate IP service. A gigabit link extends from an Avaya Ethernet Routing Switch 8800/8600 to a client, see Figure 31: Subrate IP service delivery on page 218. The configuration limits client throughput to 200 Mb/s. Traffic that exceeds the configured rate limit is dropped.

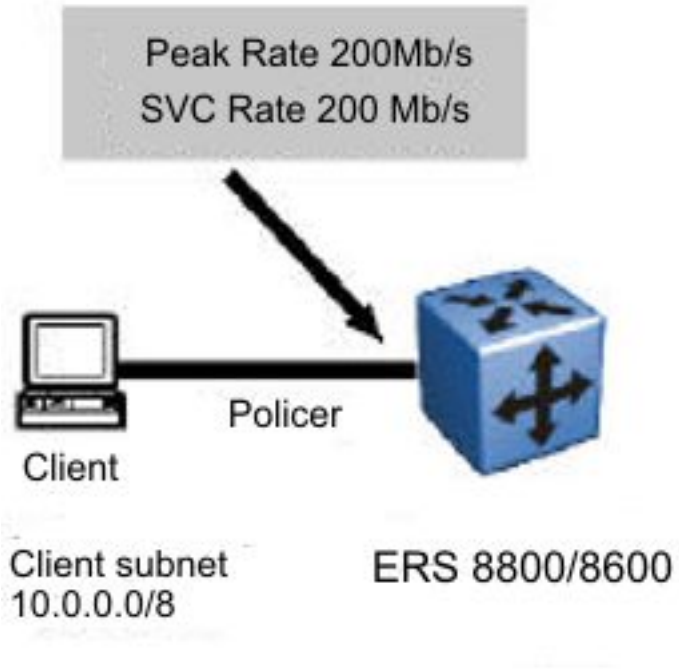


Figure 31: Subrate IP service delivery

If you need additional bandwidth, you can increase the rate by performing a soft configuration on the Avaya Ethernet Routing Switch 8800/8600. In this configuration, IP traffic from a source affects the filter action policer that is bound to the policy.

The switch drops packets above the peak rate, and you can configure the policer on an individual lane basis as required.

Procedure steps

1. Create a QoS traffic policy:

   ```
   ERS-8606:5# config qos policy 1

   ERS-8606:5# config qos policy 1 create peak rate 200000 svc-rate 200000

   ERS-8606:5/config/qos/policy/1# name ClientA

   ERS-8606:5# info

                       Id  : 1 Status    : Entry is created Name : "ClientA" peak-rate : 200000 svc-rate : 200000 lanes : 2/1,2/2
   ```

2. Create an ACT:

Comments? infodev@avaya.com

```
ERS-8605:5# config filter act 1 create name "Source"
ERS-8606:5# config filter act 1 ip srcip ERS-8606:5# config
filter act 1 apply
```

3. Create an ACL:

```
ERS-8606:5# config filter acl 1 create inPort act 1 name
"Policer1" ERS-8606:5# config filter acl 1 port add 2/11,2/13
```

4. Create an ACE and bind it to the traffic policy:

```
ERS-8606:5# config filter acl 1 ace 1 create ERS-8606:5#
config filter acl 1 ace 1 action permit police 1 ERS-8606:5#
config filter acl 1 ace 1 ip scr-ip eq
10.0.0.0-10.255.255.255 ERS-8606:5# config filter acl 1 ace 1
enable
```

   You can also configure the ACE in one line:

```
config filter acl 1 ace 1 create; action police 1; ip srcr-ip
eq 10.0.0.0-10.255.255.255; enable
```

Policing multiple flows using VLAN-based ACLs

In the following example, you classify incoming traffic at VLAN 100, see Figure 32: Multiple flow policing using VLAN-based ACLs on page 220, and police different flows according to the peak and service rate requirements shown in the following table.

Table 27: Flow requirements

Traffic type	Peak rate	Service rate
Web HTTP	200 Mb/s	100 Mb/s
FTP file transfer	100 Mb/s	50 Mb/s
UDP RTP	80 Mb/s	60 Mb/s
Other TCP port	50 Mb/s	40 Mb/s

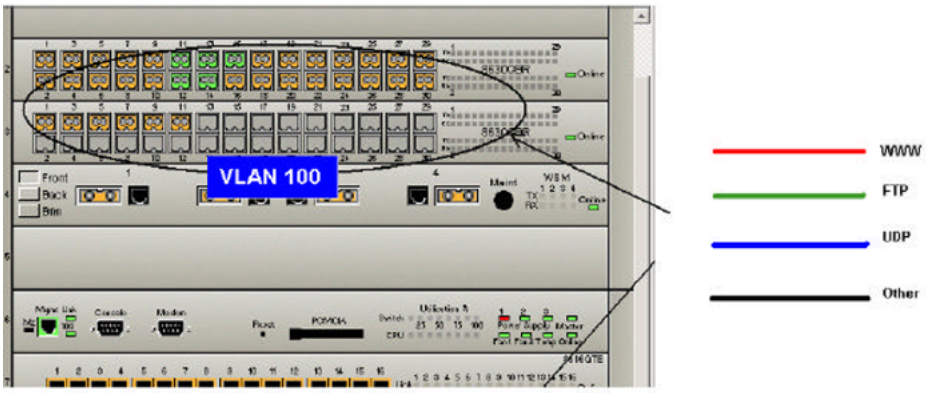


Figure 32: Multiple flow policing using VLAN-based ACLs

Procedure steps

1. Configure a WWW policy.

   ```
   ERS-8606:5# config qos policy 11 create peak-rate 200000 svc-rate 10000

   ERS-8606:5/config/qos/policy/11# lanes add 1/1,1/2,1/3

   ERS-8606:5/config/qos/policy/11# name WWW
   ```

   The name is optional. Use the optional lane parameter to apply the policy only to slot 1.

2. Display the policy configuration:

   ```
   ERS-8606:5# show qos config policy policy 11
   ```

   ```
   ERS-8610:5# show qos config policy policy 11

   ================================================================================
                R-Module QOS Policer Table
   ================================================================================
   PolicerID     Name       peak-rate     svc-rate      lanes
   --------------------------------------------------------------------------------
   11            WWW        200000        100000        1/1,1/2,1/3
   ```

3. Configure a policy for File Transfer Protocol (FTP):

   ```
   ERS-8605:5# config qos policy 12 create peak-rate 100000 svc-rate 50000

   ERS-8606:5/config/qos/policy/12# lanes add 1/1,1/2,1/3

   ERS-8606:5/config/qos/policy/12# name FTP
   ```

4. Display the policy configuration:

   ```
   ERS-8606:5/show/qos/config/policy/12# policy 12
   ```

```
ERS-8610:5# show qos config policy policy 12

================================================================================
             R-Module QOS Policer Table
================================================================================
PolicerID    Name      peak-rate    svc-rate     lanes
--------------------------------------------------------------------------------
12           FTP       100000       50000        1/1,1/2,1/3
```

5. Configure a policy for User Datagram Protocol (UDP):

   ```
   ERS-8606:5# config qos policy 13 create peak-rate 800000 svc-rate 60000
   ```

   ```
   ERS-8606:5/config/qos/policy/13# lanes add 1/1,1/2,1/3
   ```

   ```
   ERS-8606:5/config/qos/policy/13# name UDP
   ```

6. Display the policy configuration:

   ```
   ERS-8606:5/show/qos/config/policy/13# policy 13
   ```

```
ERS-8610:5# show qos config policy policy 13

================================================================================
             R-Module QOS Policer Table
================================================================================
PolicerID    Name      peak-rate    svc-rate     lanes
--------------------------------------------------------------------------------
13           UDP       80000        60000        1/1,1/2,1/3
```

7. Configure a policy for all other traffic:

   ```
   ERS-8606:5# config qos policy 14 create peak-rate 500000 svc-rate 40000
   ```

   ```
   ERS-8606:5/config/qos/policy/14# lanes add 1/1,1/2,1/3
   ```

   ```
   ERS-8606:5/config/qos/policy/14# name Other
   ```

8. Display the policy configuration:

   ```
   ERS-8606:5/show/qos/config/policy/13# policy 13
   ```

```
ERS-8610:5# show qos config policy policy 14

================================================================================
             R-Module QOS Policer Table
================================================================================
PolicerID    Name      peak-rate    svc-rate     lanes
--------------------------------------------------------------------------------
14           Other     50000        40000        1/1,1/2,1/3
```

9. Create filters and bind them to policies. Create an ACT:

   ```
   ERS-8606:5/config# filter act 100 create name "TCPIP"
   ```

   ```
   ERS-8606:5/config# filter act 100 ip scrip, dstip
   ```

```
ERS-8606:5/config# filter act 100 protocol
tcpSrcPort,udpSrcPort,tcpDstPort,udpDstPort

ERS-8606:5/config# filter act 100 apply
```

10. Create an ACL:

```
ERS-8606:5/config# filter acl 100 create inVlan act 100

ERS-8606:5/config# filter acl 100 vlan add 100
```

11. Create an ACE. Classify HTTP and the binding policy:

```
ERS-8606:5/config# filter acl 100 ace 1 create

ERS-8606:5/config# filter acl 100 ace 1 action permit police
11

ERS-8606:5/config# filter acl 100 ace 1 protocol tcp-dst-port
eq http

ERS-8606:5/config# filter acl 100 ace 1 enable
```

12. Classify FTP (control and data packets) and the binding policy:

```
ERS-8606:5/config# filter acl 100 ace 2 create

ERS-8606:5/config# filter acl 100 ace 2 action permit police
12

ERS-8606:5/config# filter acl 100 ace 2 protocol tcp-dst-port
eq ftpcontrol

ERS-8606:5/config# filter acl 100 ace 2 enable

ERS-8606:5/config# filter acl 100 ace 3 create

ERS-8606:5/config# filter acl 100 ace 3 action permit police
12

ERS-8606:5/config# filter acl 100 ace 3 protocol tcp-dst-port
eq ftpdata

ERS-8606:5/config# filter acl 100 ace 3 enable
```

13. Classify RTP and the binding policy:

```
ERS-8606:5/config# filter acl 100 ace 4 create

ERS-8606:5/config# filter acl 100 ace 4 action permit police
13

ERS-8606:5/config# filter acl 100 ace 4 protocol udp-dst-port
eq rtp

ERS-8606:5/config# filter acl 100 ace 4 enable
```

14. Configure the TCP port and binding policy:

```
ERS-8606:5/config# filter acl 100 ace 5 create
```

```
ERS-8606:5/config# filter acl 100 ace 5 action permit police
14

ERS-8606:5/config# filter acl 100 ace 5 protocol tcp-dst-port
eq 0

ERS-8606:5/config# filter acl 100 ace 5 enable
```

Mirroring using ACLs

For more information about port mirroring and remote port mirroring, see *Avaya Ethernet Routing Switch 8800/8600 Troubleshooting,* (NN46205-703).

This configuration example shows how to perform the following tasks:

- Enable port mirroring (RxFilter mode) for a port on VLAN 220.
- Use port 3/48 as the monitoring port.
- Configure an ACL so that TCP traffic from ports 20 to 500, and ICMP frames are mirrored to the monitoring port; see Figure 33: Switch configuration for port mirroring example on page 223.

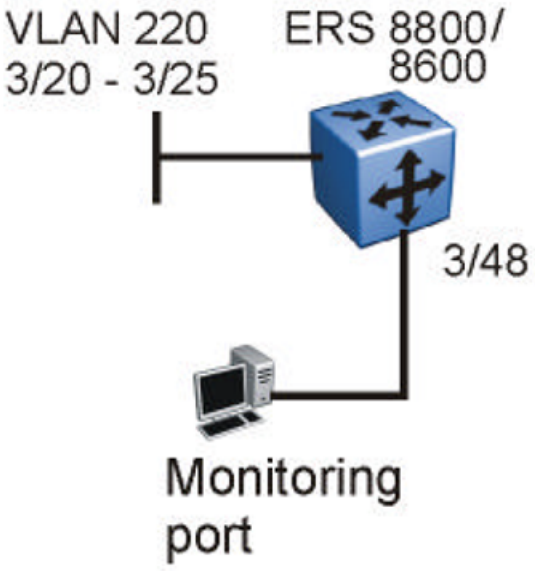


Figure 33: Switch configuration for port mirroring example

Procedure steps

1. Create a new ACT to filter on ICMP frames and TCP destination ports. Configure a new ACT with ID = 2:

```
ERS-8610:5# config filter act 2 create
```

2. Select the IP attributes of the IP protocol type:

```
ERS-8610:5# config filter act 2 ip ipProtoType
```

3. Select protocol attributes of TCP source port, TCP destination port, and UDP destination port

```
ERS-8610:5# config filter act 2 protocol tcpDstPort
```

4. Enable ACT 2:

```
ERS-8610:5# config filter act 2 apply
```

5. Create ACL 1 with type ingress VLAN:

```
ERS-8610:5# config filter acl 1 create inVlan act 2
```

6. Add ingress VLAN of 220 to ACL 1:

```
ERS-8610:5# config filter acl 1 vlan add 220
```

7. Add ACE 1 with action of permit to mirror ICMP traffic:

```
ERS-8610:5# config filter acl 1 ace 1 create name icmp

ERS-8610:5# config filter acl 1 ace 1 action permit

ERS-8610:5# config filter acl 1 ace 1 debug mirror enable
mirroring-dst-ports 3/48

ERS-8610:5# config filter acl 1 ace 1 ip ip-protocol-type eq
icmp

ERS-8610:5# config filter acl 1 ace 1 enable
```

8. Add ACE 2 with action of permit to mirror TCP traffic with a destination port range from 20 to 500:

```
ERS-8610:5# config filter acl 1 ace 2 create name tcp_range

ERS-8610:5# config filter acl 1 ace 2 action permit

ERS-8610:5# config filter acl 1 ace 2 debug mirror enable
mirroring-dst-ports 3/48

ERS-8610:5# config filter acl 1 ace 2 ip ip-protocol-type eq
tcp

ERS-8610:5# config filter acl 1 ace 2 protocol tcp-dst-port
eq 20-500

ERS-8610:5# config filter acl 1 ace 2 enable
```


Comments? infodev@avaya.com

Asymmetric downlink and uplink using policy-based policers and port-based shapers

The example that follows shows how to provision asymmetric downlink and uplink using the policer and a traffic shaper. A gigabit link extends from an Avaya Ethernet Routing Switch 8800/8600 to a client; see the following figure.

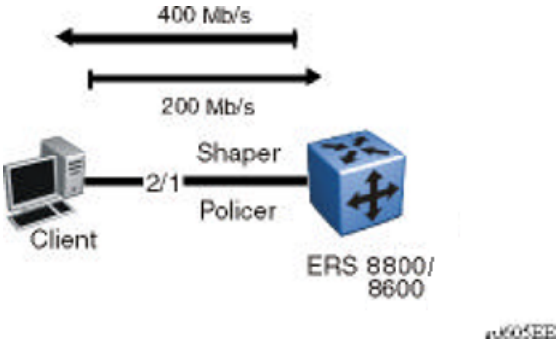


Figure 34: Asymmetric downlink and uplink

The client requirement is

- downlink of 400 Mb/s (shaped)
- uplink of 200 Mb/s (policed)

Procedure steps

1. Configure the port shaper for downlinking by configuring the shaper for a 400 Mb/s rate:

   ```
   ERS-8606:5# config ethernet 2/1 shape 400000 enable
   ```

2. Configure a QoS traffic policy:

   ```
   ERS-8606:5# config qos policy 1 create peak-rate 200000 svc-rate 200000 lanes 2/1,2/2

   ERS-8606:5# config qos policy 1 name ClientA
   ```

3. Configure an ACT:

   ```
   ERS-8606:5# config filter act 1 create name “SourceIP”

   ERS-8606:5# config filter act 1 ip srcip

   ERS-8606:5# config filter act 1 apply
   ```

4. Configure an ACL:

```
ERS-8606:5# config filter acl 1 create inPort act 1 name
“Policer1”

ERS-8606:5# config filter acl 1 port add 2/1
```

5. Configure an ACE and bind it to the traffic policy:

```
ERS-8606:5# config filter acl 1 ace 1 create

ERS-8606:5# config filter acl 1 ace 1 action permit policy 1

ERS-8606:5# config filter acl 1 ace 1 ip src-ip eq
10.0.0.0-10.255.255.255

ERS-8606:5# config filter acl 1 ace 1 enable
```

Chapter 15: Basic DiffServ configuration using the ACLI

Use DiffServ to provide appropriate Quality of Service (QoS) to specific traffic types.

Job aid

The following roadmap lists some of the QoS commands and the parameters that you can use to perform the procedures in this section.

Table 28: Roadmap of QoS ACLI commands

Command	Parameter
Global Configuration mode	
`vlan mac-address-entry <1-4094> qos-level <H.H.H> <0-6> status <other|invalid|learned|self|mgmt>`	—
`vlan mac-address-filter <1-4094> <H.H.H> <portList> <0-6>`	—
`vlan mac-address-static <1-4094> <H.H.H> <portList> qos <0-6>`	—
Interface Configuration mode	
`access-diffserv [port <portList>] [enable]`	—
`enable-diffserv [port <portList>] [enable]`	—
`qos`	`802.1p-override [enable]`
	`level [port <portList>] <0-6>`

Enabling DiffServ on a port

Enable DiffServ so that the switch provides DiffServ-based QoS on that port.

Prerequisites

- Access Interface Configuration mode.

Procedure steps

1. Enable DiffServ:

   ```
   enable-diffserv [port <portList>] [enable]
   ```

Variable definitions

Use the data in the following table to use the **enable-diffserv** command.

Variable	Value
enable	Enables DiffServ for the specified port. The default is disabled. To use the default configuration, use the default option in the command **default enable-diffserv [enable]** To delete the current configuration, use the no option in the command**no enable-diffserv [enable]**
port <portList>	Specifies the slot and port, or slot and port list. To delete the current configuration, use the no option in the command **no enable-diffserv [port <portList>]**

Configuring Layer 3 trusted or untrusted ports

Configure a port as trusted or untrusted to determine the Layer 3 QoS actions the switch performs. A trusted port honors incoming DSCP markings. An untrusted port overrides DSCP markings.

Prerequisites

- Access Interface Configuration mode.
- DiffServ is enabled.

Procedure steps

1. Configure the port as Layer 3 untrusted:

   ```
   access-diffserv [port <portList>] [enable]
   ```

   To configure the port as Layer 3 trusted, use the **no access-diffserv enable** command.

Variable definitions

Use the data in the following table to use the **access-diffserv** commands.

Variable	Value
enable	If enabled, specifies an access port and overrides incoming DSCP bits. If disabled, specifies a core port and honors and handles incoming DSCP bits. The default is disabled. To use the default configuration, use the default option in the command **default access-diffserv [enable]** To delete the current configuration, use the no option in the command**no access-diffserv [enable]**
port <portList>	Specifies the slot and port, or slot and port list. To delete the current configuration, use the no option in the command **no access-diffserv [port <portList>]**

Configuring Layer 2 trusted or untrusted ports

Configure a port as trusted or untrusted to determine the Layer 2 QoS actions the switch performs. A trusted port (override disabled) honors incoming 802.1p bit markings. An untrusted port (override enabled) overrides 802.1p bit markings.

Prerequisites

- Access Interface Configuration mode.
- DiffServ is enabled.

Procedure steps

1. Configure the port as Layer 2 untrusted:

   ```
   qos 802.1p-override [enable]
   ```

   To configure the port as Layer 2 trusted, use the **`no qos 802.1p-override`** command.

Variable definitions

Use the data in the following table to youuse the **`qos 802.1p-override`** command.

Variable	Value
`enable`	If you configure this variable, it overrides incoming 802.1p bits; if you do not configure this variable, it honors and handles incoming 802.1p bits. The default is disable (Layer 2 trusted). To use the default configuration, use the default option in the command **`default qos 802.1p-override [enable]`** To delete the current configuration, use the no option in the command**`no qos 802.1p-override [enable]`**

Configuring the port QoS level

Use the default port QoS level to assign a default QoS level for all traffic (providing the packet does not match an ACL that re-marks the packet).

Comments? infodev@avaya.com

Prerequisites

- Access Interface Configuration mode.

Procedure steps

1. Configure the port QoS level:

   ```
   qos level [port <portList>] <0-6>
   ```

Variable definitions

Use the data in the following table to use the **`qos level`** command.

Variable	Value
`<0-6>`	Specifies the default QoS level for the port traffic. QoS level 7 is reserved for network control traffic. The default is 1. To use the default configuration, use the default option in the command **`default qos level`**
`port <portList>`	Specifies the slot and port, or slot and port list.

Configuring the VLAN QoS level

You can change the default port or VLAN QoS levels to assign a default QoS level for all traffic, providing the packet does not match an ACL that re-marks the packet.

Prerequisites

- Access VLAN Interface Configuration mode.
- The VLAN exists.

Procedure steps

1. Configure the VLAN level:

   ```
   qos level <0-6>
   ```

Variable definitions

Use the data in the following table to use the **qos level** command.

Variable	Value
<0-6>	Specifies the default QoS level for the VLAN traffic. QoS level 7 is reserved for network control traffic. The default is 1. To use the default configuration, use the default option in the command**default qos level**

Configuring the QoS level for a MAC address

Apply a QoS level to traffic from specific VLAN MAC addresses to provide special QoS treatment to the packets and to modify the QoS level providing that the packet does not match an ACL that re-marks the packet.

For more information about the VLAN commands, see *Avaya Ethernet Routing Switch 8800/8600 Configuration — VLANs and Spanning Tree,* (NN46205-517).

Prerequisites

- Access Global Configuration mode.
- The VLAN exists.

Procedure steps

1. Configure the source MAC QoS level for a dynamically learned address:

Comments? infodev@avaya.com

```
vlan mac-address-entry <1-4094> qos-level <H.H.H> <0-6>
status <other|invalid|learned|self|mgmt>
```

2. Configure the source MAC QoS level for a bridge static address:

   ```
   vlan mac-address-static <1-4094> <H.H.H> <portList> qos <0-6>
   ```

3. Configure the source MAC QoS level for a bridge filter address:

   ```
   vlan mac-address-filter <1-4094> <H.H.H> <portList> <0-6>
   ```

Variable definitions

Use the data in the following table to use the commands in this procedure.

Variable	Value
`<0-6>`	Specifies the QoS level. The default is 1. To use the default configuration, use the default option in the command.
`<1-4094>`	Specifies the VLAN ID.
`<H.H.H>`	Specifies the MAC address in the format 0x00:0x00:0x00:0x00:0x00:0x00
`<portList>`	Specifies the slot and port, or slot and port list.
`status <other|invalid|learned|self|mgmt>`	Specifies the FDB status (other|invalid|learned| self|mgmt)

Example of setting a QoS level for a MAC address

Procedure steps

1. To change the source MAC QoS level to 2 for the MAC address 00:00:00:00:01:0a on VLAN 2 through port 7/26, enter the following command:

   ```
   ERS-8610:5# vlan mac-address-static 2 00:00:00:00:01:0a 7/26
   qos 2
   ```

Comments? infodev@avaya.com

Chapter 16: QoS configuration using the ACLI

Use the procedures in this section to configure Quality of Service (QoS) on the Avaya Ethernet Routing Switch 8800/8600.

For information about statistics, see *Avaya Ethernet Routing Switch 8800/8600 Performance Management,* (NN46205-704)

Job aid

The following roadmap lists some of the QoS commands and the parameters that you can use to perform the procedures in this section.

Table 29: Roadmap of QoS ACLI commands

Command	Parameter
Privileged EXEC mode	
`qos apply egress-queue-set <1-386>`	—
`show qos 802.1p-override`	`fastEthernet <portList>`
	`GigabitEthernet <portList>`
	`vlan <1-4094>`
`show qos egress-queue-set`	`<1-386> [queue <0-63>]`
	`port <portList>`
`show qos egressmap`	`1p [<0-7>]`
	`ds [<0-7>]`
	`exp [<0-7>]`
`show qos eqmap <1-10>`	—
`show qos ingressmap`	`1p [<0-7>]`
	`ds [<0-63>]`
	`exp [<0-7>]`

Command	Parameter
show qos policer	interface fastEthernet <portList>
	interface gigabitEthernet <portList>
show qos policy-config [<0-16383>] [lane <WORD 1-128>] [port <portList>]	—
show qos queue [<0-7>]	—
show qos shaper	interface fastEthernet <portList>
	interface gigabitEthernet <portList>
	interface vlan <1-4094>
show qos statistics	egress-queue-set [<1-386>] [interface-type <fastEthernet|gigabitEthernet> <portList>] [detail]
	policy [<0-20000>] [lane <WORD 1-128>] [port <portList>]
Global Configuration mode	
qos egress-queue-set	<1-386> <portList>
	qmax <1-386> <8|64> [balanced-queues <0-48>] [hipri-queues <0-64>] [lopri-queues <0-8>] [name <WORD 0-32>]
qos egress-queue-set queue <1-386> <0-63>	max-length <0-32760>
	max-rate <0-100>
	min-rate <0-100>
	name <WORD 0-32>
qos egressmap	1p <0-7> <0-7>
	ds <0-7> <WORD 1-6>
	exp <0-7> <0-7>
qos ingressmap	1p <0-7> <0-7>
	ds <0-63> <0-7>
	exp <0-7> <0-7>

Comments? infodev@avaya.com

Command	Parameter
qos policy <1-16383>	peak-rate <250-10000000> svc-rate <250-10000000>
	lanes <WORD 1-128>
	name <WORD 1-32>
qos threshold <0–3>	
Interface Configuration mode	
bandwidth-limit	[port <portList>] broadcast <250-2147483647>
	[port <portList>] multicast <250-2147483647>
qos	if-policer [port <portList>] police-rate <1000–10000000>
	if-shaper [port <portList>] shape-rate <1000–10000000>
rate-limit	
GigabitEthernet Interface Configuration Mode	
enable-diffserv	[port <portlist>] enable
no access-diffserv	[port <portlist>] enable
qos 802.1p-override	enable

Configuring broadcast and multicast bandwidth limiting

Use broadcast and multicast bandwidth limiting to restrict the amount of ingress broadcast and multicast traffic on a port. The switch drops traffic that violates the bandwidth limit.

Prerequisites

- Access Interface Configuration mode.

Procedure steps

1. Configure broadcast bandwidth limiting:

   ```
   bandwidth-limit [port <portList>] broadcast <250-2147483647>
   ```

2. Configure multicast bandwidth limiting:

   ```
   bandwidth-limit [port <portList>] multicast <250-2147483647>
   ```

Variable definitions

Use the data in the following table to use the **`bandwidth-limit`** commands.

Variable	Value
`broadcast <250-2147483647>`	Specifies the bandwidth limit for broadcast traffic from 250–2147483647 Kb/s. To delete the current configuration, use the no option in the command: **`no bandwidth-limit [port <portList>] broadcast`** To use the default configuration, use the default option in the command: **`default bandwidth-limit broadcast`**. The default is disabled.
`multicast <250-2147483647>`	Specifies the bandwidth limit for multicast traffic from 250–2147483647 Kb/s. To delete the current configuration, use the no option in the command: **`no bandwidth-limit [port <portList>] multicast`** To use the default configuration, use the default option in the command: **`default bandwidth-limit multicast`**. The default is disabled.
`port <portList>`	Specifies the slot and port, or a list of slots and ports. To delete the current configuration, use the no option in the command: **`no bandwidth-limit port <portList>`**

Comments? infodev@avaya.com

Variable	Value
	To use the default configuration, use the default option in the command: **default bandwidth-limit port <portList>**

Configuring the port-based shaper

Use port-based shaping to rate-limit all outgoing traffic to a specific rate.

For information about configuring queue-based shaping, see Configuring an egress queue set queue on page 173.

Prerequisites

- Access Interface Configuration mode.

Procedure steps

1. Configure port-based shaping:

   ```
   qos if-shaper [port <portList>] shape-rate <1000–10000000>
   ```

Variable definitions

Use the data in the following table to use the **qos if-shaper** command.

Variable	Value
`port <portList>`	Specifies the slot and port, or slot and portlist.
`shape-rate <1000-10000000>`	Configures the shaping rate from 1000–10000000 Kb/s.

Configuring a port-based policer for RS and 8800 modules

Use a port policer to bandwidth-limit incoming traffic. The switch drops or re-marks violating traffic. Only RS and 8800 modules support this policer.

Prerequisites

- Access Interface Configuration mode.

Procedure steps

1. Assign the policing limit:

   ```
   qos if-policer [port <portList>] police-rate <1000–10000000>
   ```

Variable definitions

Use the data in the following table to use the **`qos if-policer`** command.

Variable	Value
`police-rate <1000–10000000>`	Specifies the ingress rate limit (policing limit) in Kb/s. The range is 1000–10000000.
`port <portList>`	Specifies the slot and port or slot and portlist.

Configuring a policy-based policer

Use a QoS policy to configure peak and service policing rates for specific lane members.

Comments? infodev@avaya.com

Prerequisites

- Access Global Configuration mode.

Procedure steps

1. Configure a policer (traffic policy):

   ```
   qos policy <1-16383> peak-rate <250-10000000> svc-rate
   <250-10000000> [lanes <WORD 1-128>] [name <WORD 1-32>]
   ```

2. Ensure that your configuration is correct:

   ```
   show qos policy-config [<0-16383>] [lane <WORD 1-128>] [port
   <portList>]
   ```

Variable definitions

Use the information in the following table to use the commands in this procedure.

Variable	Value
`<1-16383>`	Specifies the policer ID number.
`peak-rate <250-10000000>`	Configures the policer peak rate in Kb/s.
`srv-rate <250-10000000>`	Configures the policer service rate in Kb/s.
`lanes <WORD 1-128>`	Specifies the lanes to which the policer applies: • all • slot/lane [-slot/lane][,-]
`name <WORD 1-32>`	Names the policer template.
`port <portList>`	Specifies the slot and port, or slot and port list.

Job aid

The following table describes the headings in the show command output.

Table 30: show qos policy-config output

Field	Description
PolicerID	Specifies the policer ID number.
Name	Specifies the name of the policer.
peak-rate	Specifies a policer peak rate in Kb/s.
svc-rate	Specifies a local policer service rate in Kb/s.
lanes	Specifies the lane numbers associated with the policy.

Configuring an egress queue set

Configure an egress queue set to apply the same egress queue configuration (a template) to a group (set) of ports. Base shapers on egress queue sets.

Prerequisites

- Access Global Configuration mode.

Procedure steps

1. Configure the egress queue set template:

   ```
   qos egress-queue-set qmax <1-386> <8|64> [balanced-queues
   <0-48>] [hipri-queues <0-64>] [lopri-queues <0-8>] [name
   <WORD 0-32>]
   ```

2. Associate ports with the egress queue set:

   ```
   qos egress-queue-set <1-386> <portList>
   ```

   The system verifies that the requested port types support the number of queues in the egress queue set. If you add ports to an applied template, the system sends additional messages to the relevant module control processors and configures the hardware accordingly.

3. Ensure the configuration is correct:

   ```
   show qos statistics egress-queue-set <1-386> [detail]
   ```


Comments? infodev@avaya.com

4. To configure the egress queue set queues, do so now, before you apply the egress queue set.
5. To apply all configuration changes, exit Global Configuration mode, and then in Privileged EXEC mode, enter:

   ```
   qos egress-queue-set <1-386> apply
   ```

Variable definitions

Use the information in the following table to use the **`qos egress-queue-set qmax <1-386> <8|64>`** commands.

Variable	Value
`<1-386>`	Identifies the egress queue template.
`apply`	Applies the egress queue set when you issue the command. When you create an egress queue set, apply occurs when you issue the command. When you modify a queue set, apply occurs after you save the configuration and boot the switch. This command is available only in Privileged EXEC mode.
`balanced-queues <0-48>`	Specifies the maximum number of balanced queues in the egress queue set.
`hipri-queues <0-64>`	Specifies the maximum number of high-priority queues in the egress queue set.
`lopri-queues <0-8>`	Specifies the maximum number of low-priority queues in the egress queue set.
`name <WORD 0-32>`	Names the egress queue set template.
`qmax <8|64>`	Specifies the maximum number of queues, either 8 or 64. The sum of the number of queues for balanced, hipri, and lopri queues must be less than or equal to qmax.

Use the information in the following table to youuse the **`qos egress-queue-set <1-386> <portList>`** command.

Variable	Value
`<1-386>`	Identifies the egress queue set.
`<portList>`	Specifies the list of ports. To remove ports to an egress queue set, use the following command:

Variable	Value
	`no qos egress-queue-set <1-386> <portList>`

Job aid

The following table describes the headings in the show command output.

Table 31: Description of terms in show command output

Field	Description
Qid	Queue offset from the base queue
Q-name	Name of the queue
Q-Style	Queuing style: low priority; high priority; or balanced
min-rate	Minimum guaranteed rate
max-rate	Maximum data rate
max-q-length	Maximum queue length
TemplateID	Template ID
Name	Name of the template
Total Qs	Total number of queues
BalQs	Number of balanced queues
Hi-priQs	Number of high-priority queues
lo-priQs	Number of low-priority queues
Total pages	Total pages offered to the queue
Dropped pages	Total pages dropped by the queue
Utilization	Percent of queue usage

Configuring an egress queue set queue

Configure an egress queue set queue to customize shaping behavior.

When you create a new custom queue, you MUST re-configure the default values provided for the new queue to suit customer QoS requirements.

Comments? infodev@avaya.com

Caution:

Risk of packet loss

If you modify an egress queue set queue, you must restart the switch.

Important:

For each Balanced queue, you must specify a desired minimum rate (min-rate) guarantee and a maximum-rate (max-rate) limit.

For Priority queues (either high or low priority), a minimum rate guarantee does not apply. Configure only a rate limit (max-rate).

The sum of minimum rate guarantees must be less than the port line rate minus the sum of high-priority queue rate limits. If this condition is not met, minimum rates are not guaranteed.

Prerequisites

- Access Global Configuration mode.

Procedure steps

1. Configure the QoS egress queue set queue:

   ```
   qos egress-queue-set queue <1-386> <0-63> [max-length <0-32760>] [max-rate <0-100>] [min-rate <0-100>] [name <WORD 0-32>]
   ```

2. To apply the changes to the queue set, exit Global Configuration mode, and then in Privileged EXEC mode, enter:

   ```
   qos apply egress-queue-set <1-386>
   ```

   If you modify an existing queue set, save the configuration, and then restart the switch.

Variable definitions

Use the information in the following table to use the **`qos egress-queue-set queue`** commands.

Variable	Value
`<0-63>`	Identifies the queue.

Variable	Value
`<1-386>`	Identifies the egress queue template.
`max-length <0-32760>`	Specifies the limit to which a queue can grow. The queue length does not imply that a queue has a fixed number of buffers. For example, a queue can grow to full memory size of 32 K buffers.
`max-rate <0-100>`	Specifies the maximum line rate in percent to accommodate various port speeds in the same template. The max-rate maximum is 100 percent. For example, if a 20 percent rate applies to a 10 and 1 Gb/s Ethernet port, the result is a 2 Gb/s bandwidth allocation for 10 Gb/s Ethernet and 200 Mb/s for a 1 Gb/s Ethernet port.
`min-rate <0-100>`	Specifies the minimum line rate in percent to accommodate various port speeds in the same template.
`name <WORD 0-32>`	Names the egress queue.

Modifying an egress queue set or egress queue set queue

Modify a queue set or queue to change shaping behavior.

Caution:
Risk of packet loss

If you modify an egress queue set, you must restart the switch.

Prerequisites

- Access Global Configuration mode.

Procedure steps

1. After you apply a queue set, you can modify the queue min-rate and max-rate parameters:

   ```
   qos egress-queue-set queue <1-386> <0-63> [max-length
   <0-32760>] [max-rate <0-100>] [min-rate <0-100>] [name <WORD
   0-32>]
   ```


Comments? infodev@avaya.com

2. Modify associated ports with the egress queue set:

   ```
   qos egress-queue-set <1-386> <portList>
   ```

   Remove ports to an egress queue set:

   ```
   no qos egress-queue-set <1-386> <portList>
   ```

3. You cannot modify other queue set parameters. If you require different queue set parameters, you must delete the queue set and configure another. If you attempt to change another parameter, the following message appears:

   ```
   Error: Modification of ADSSC Egress QSet values not allowed. Only Queue
   Min/Max rate modification allowed.
   ```

4. Ensure the configuration is correct:

   ```
   show qos egress-queue-set [<1-386>] [detail]
   ```

5. To apply all configuration changes, exit Global Configuration mode, and then in Privileged EXEC mode, enter:

   ```
   qos apply egress-queue-set <1-386>
   ```

   The following message appears:

   ```
   WARNING: The egress-queue-set QoS change made will take effect only after
   the configuration is saved and the chassis is rebooted.
   ```

6. Save the configuration as required:

   ```
   save config
   ```

   ```
   save config standby config.cfg
   ```

   ```
   save bootconfig
   ```

   ```
   save bootconfig standby boot.cfg
   ```

7. Restart the switch:

   ```
   boot -y
   ```

8. Verify the changes:

   ```
   show qos egress-queue-set [<1-386>]
   ```

Variable definitions

Use the information in the following table to use the commands in this procedure.

Variable	Value
<1-386>	Identifies the egress queue template.

Configuring ingress mappings

You can modify the ingress mappings to change traffic priorities. However, Avaya recommends that you use the default mappings.

Prerequisites

- Access Global Configuration mode.

Procedure steps

1. Configure MPLS to QoS ingress mappings:

   ```
   qos ingressmap exp <0-7> <0-7>
   ```

2. Configure DSCP to QoS ingress mappings:

   ```
   qos ingressmap ds <0-63> <0-7>
   ```

3. Configure 802.1p bit to QoS ingress mappings:

   ```
   qos ingressmap 1p <0-7> <0-7>
   ```

4. Ensure the configuration is correct:

   ```
   show qos ingressmap
   ```

Variable definitions

Use the information in the following table to use the **`qos ingressmap`** commands.

Variable	Value
`1p <0-7> <0-7>`	Maps the IEEE 802.1p bit to QoS level. Each QoS level has a default IEEE 1P value: • level 0—1 • level 1—0 • level 2—2 • level 3—3 • level 4—4

Comments? infodev@avaya.com

Variable	Value
	• level 5—5 • level 6—6 • level 7—7 To use the default configuration, use the default option in the command**`default qos ingressmap 1p`**
`ds <0-63> <0-7>`	Maps the DS byte to QoS level.
`exp <0-7> <0-7>`	Maps the MPLS EXP bit to a QoS level. Each option has a range from 0–7.

Configuring egress mappings

You can modify the egress mappings to change traffic priorities. However, Avaya recommends that you use the default mappings.

Prerequisites

- Access Global Configuration mode.

Procedure steps

1. Configure QoS to MPLS egress mappings:

   ```
   qos egressmap exp <0-7> <0-7>
   ```

2. Configure QoS to DSCP egress mappings:

   ```
   qos egressmap ds <0-7> <WORD 1-6>
   ```

3. Configure QoS to 802.1p bit egress mappings:

   ```
   qos egressmap 1p <0-7> <0-7>
   ```

4. Ensure the configuration is correct:

   ```
   show qos egressmap
   ```

Variable definitions

Use the information in the following table to use the **qos egressmap** commands.

Variable	Value
1p <0-7> <0-7>	Maps the QoS level to IEEE 802.1p priority. Each QoS level has a default IEEE 1P value: • level 0—1 • level 1—0 • level 2—2 • level 3—3 • level 4—4 • level 5—5 • level 6—6 • level 7—7 To use the default configuration, use the default option in the command**default qos ingressmap 1p**
ds <0-7> <WORD 1-6>	Maps the QoS level to DS byte. You can specify the DSCP in either hexadecimal, binary, or decimal.
exp <0-7> <0-7>	Maps the QoS level to MPLS EXP level.

Configuring Avaya Automatic QoS

Configure the Avaya Automatic QoS to automatically recognize the DSCP values that Avaya voice applications use and to associate them with the proper egress queues.

Prerequisites

Log on to the Interface Configuration mode in the ACLI.

Procedure steps

1. Enable diffserv on a port by using the following command:

Comments? infodev@avaya.com

```
enable-diffserv [port <portlist>] enable
```

2. Enable a port as a trusted core port by using the following CLI command:

   ```
   no access-diffserv [port <portlist>] enable
   ```

3. For tagged ports, enable 802.1p override by using the following command:

   ```
   qos 802.1p-override enable
   ```

Comments? infodev@avaya.com

Chapter 17: Traffic filter configuration using the ACLI

Use traffic filtering to block unwanted traffic or to prioritize desired traffic.

Traffic filter configuration procedures

This task flow shows you the sequence of procedures you perform to configure traffic filters.

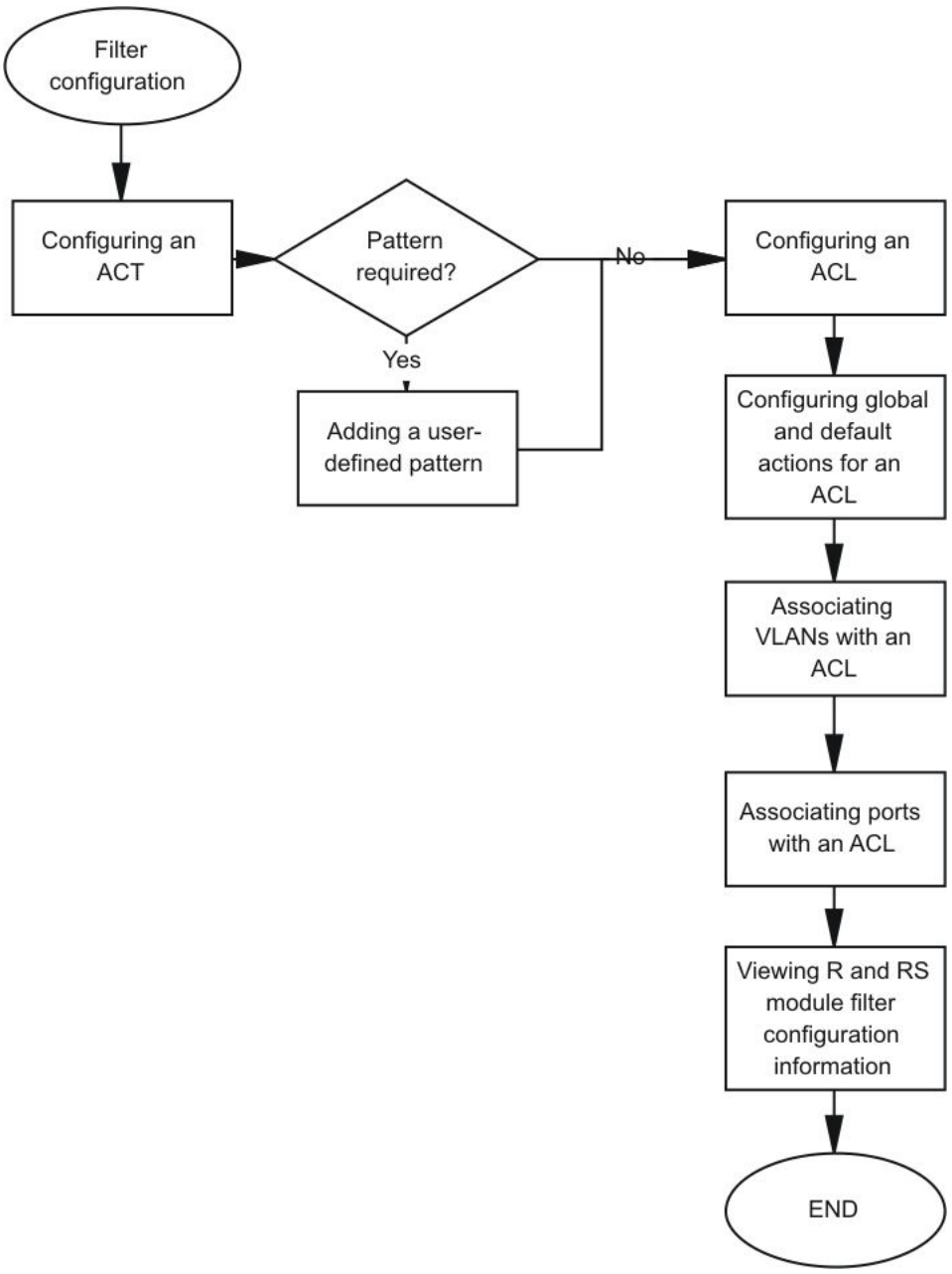


Figure 35: Traffic filter configuration procedures

Job aid

The following roadmap lists traffic filter commands that you can use to perform the procedures in this section.

Table 32: Roadmap of traffic filter ACLI commands

Command	Parameters
Privileged EXEC mode	
clear filter acl statistics	default [<1-4096>]
	port [<1-4096> [<1-1000> [<portList>]]]
show filter acl	<1-4096>
	ace [<1-4096>] [<1-1000>]
	action [<1-4096>] [<1-1000>]
	advanced [<1-4096>] [<1-1000>]
	arp [<1-4096>] [<1-1000>]
	config [<1-4096>] [<1-1000>]
	debug [<1-4096>] [<1-1000>]
	ethernet [<1-4096>] [<1-1000>]
	ip [<1-4096>] [<1-1000>]
	ipv6 [<1-4096>] [<1-1000>]
	protocol [<1-4096>] [<1-1000>]
	statistics default [<1-4096>]
	statistics port [<1-4096> [<1-1000> [<portList>]]]
show filter act [<1-4096>]	—
show filter act-pattern [<1-4096>]	—
Global Configuration mode	
filter acl <1-4096>	enable
	name <WORD 0-32>
	type <inVlan|outVlan|inPort| outPort> act <1-4096> [pktType <ipv4|ipv6>] [name <WORD 0-32>]
filter acl port <1-4096> <portList>	—
filter acl set <1-4096>	default-action <deny|permit>

Command	Parameters
	`global-action <count|count-ipfix|ipfix|mirror|mirror-count|mirror-count-ipfix|mirror-ipfix>`
`filter acl vlan <1-4096> <1-4094>`	—
`filter act <1-4096>`	`arp operation`
	`ethernet <srcMac|dstMac|ethertype|<port|vlan>|vlanTagPrio>`
	`ip <srcip|dstIp|ipFragFlag|ipOptions|ipProtoType|dscp>`
	`ipv6 <srcipv6|dstIpv6|nextHdr>`
	`name <WORD 0-32>`
	`protocol <tcpSrcPort|udpSrcPort|tcpDstPort|udpDstPort|tcpFlags|icmpMsgType>`
`filter act pattern <1-4096> <WORD 0-32> <base> <0-76800> <1-56>`	—
`filter apply act <1-4096>`	—

Configuring an ACT

Use an access control template (ACT) to specify all possible match fields for an access control list (ACL).

Prerequisites

- Enter Global Configuration mode.
- To add a pattern, the ACT must be inactive (Apply = false).

Comments? infodev@avaya.com

Procedure steps

1. Create the ACT:

   ```
   filter act <1-4096> [name <WORD 0-32>]
   ```

   <1-4096> specifies an ACT ID from 1 to 4096.

2. Configure the required ACT attributes: ARP, IP, IPv6, protocol, and Ethernet. You can specify ACE attributes only for the attributes that you specify in the ACT.
3. Optionally, add a pattern.
4. Ensure the configuration is correct:

   ```
   show filter act [<1-4096>]
   ```

5. Apply (commit) your changes:

   ```
   filter apply act <1-4096>
   ```

   After you issue the apply command, you cannot modify the ACT. If you require different attributes or patterns, you must delete the ACT and create a new one.

Variable definitions

Use the information in the following table to use the **`filter act <1-4096>`** commands.

Variable	Value
`apply`	Applies or commits the ACT. After you issue the **`apply`** command, to change the ACT, you must delete it (if no ACLs are associated with it) and recreate it.
`arp <operation>`	Specifies the permitted ARP attributes for the ACT. The only option is operation.
`ip <ip-attributes>`	Specifies the permitted IP attributes for the ACT. Separate the list of attributes by commas: srcIp, dstIp, ipFragFlag, ipOptions, ipProtoType, or dscp. The default is none. To use the default configuration, use the default option in the command: **`default filter act <1-4096> ip`**
`ethernet <srcMac|dstMac|ethertype|<port|vlan>|vlanTagPrio>`	Specifies the permitted Ethernet attributes for the ACT. Separate the list of attributes by commas: srcMac, dstMac, etherType, <port|vlan>, or vlanTagPrio. The default is none.

Variable	Value
	To use the default configuration, use the default option in the command: **default filter act <1-4096> ethernet**
ipv6 <srcipv6|dstIpv6|nextHdr>	Specifies the permitted IPv6 attributes. Separate the list of allowed attributes by commas: srcIpv6, dstIpv6, or nextHdr.
name <WORD 0-32>	Specifies an optional name for the ACT that uses 0–32 characters. If you do not enter a name, the switch generates a default name. You can change the name at any time, even after you issue the **apply** command.
protocol <tcpSrcPort|udpSrcPort|tcpDstPort|udpDstPort|tcpFlags|icmpMsgType>	Specifies the permitted protocol attributes for the ACT. Separate the list of attributes by commas: tcpSrcPort, udpSrcPort, tcpDstPort, udpDstPort, tcpFlags, or icmpMsgFlags. The default is none. To use the default configuration, use the default option in the command: **default filter act <1-4096> protocol**

Adding a user-defined pattern

Add a user-defined pattern to which the ACT can match. An ACT can have a maximum of three associated patterns.

Prerequisites

- You can insert a pattern into an ACT only if it is inactive.
- Enter Global Configuration mode.

Procedure steps

1. Create a template for patterns within an ACT:

   ```
   filter act pattern <1-4096> <WORD 0-32> <base> <0-76800>
   <1-56>
   ```

2. Ensure the configuration is correct:

   ```
   show filter act-pattern [<act-id>]
   ```


Comments? infodev@avaya.com

Variable definitions

Use the information in the following table to use the pattern commands.

Variable	Value
<0-76800>	The *<0-76800>* parameter specifies the offset: the number of bits from the base where the pattern starts.
<1-56>	The *<1-56>* parameter specifies the length in bits of the user-defined field from 1–56.
<base>	The *<base>* parameter specifies the base. The base and the offset together determine the beginning of the pattern. Permitted values for the base include ether-begin, mac-dst-begin, mac-srcbegin, ethTypeLen-begin, arp-begin, ip-hdr-begin, ip-options-begin, ip-payload-begin, ip-tos-begin, ip-proto-begin, ip-src-begin, ip-dst-begin, ipv6-hdr-begin, tcp-begin, tcp-srcport-begin, tcp-dstport-begin, tcp-flags-end, udp-begin, udp-srcport-begin, udp-dstport-begin, ether-end, ip-hdr-end, icmp-msg-begin, tcp-end, or udp-end.
<WORD 0-32>	Names the pattern with a new name that you define. Each of the three patterns must have a unique name.

Configuring an ACL

Use an ACL to specify an ordered list of ACEs, or filter rules. The ACEs provide specific actions for the filter to perform.

When you create an ACL with the type inVlan that uses an ACT based on the source IP address, the ACL no longer works after the ARP aging time elapses. This does not cause a security breach. For a solution to this issue, see Workaround for inVlan, srcIp ACL on page 351.

Prerequisites

- An ACT exists.
- You cannot use an ACL to reference an ACT until you apply the ACT.
- Enter Global Configuration mode.

Procedure steps

1. Create and configure an ACL:

   ```
   filter acl <1-4096> type <inVlan|outVlan|inPort|outPort> act
   <1-4096> [pktType <ipv4|ipv6>] [name <WORD 0-32>]
   ```

   <1-4096> specifies a unique identifier (1 to 4096) for this ACL; **act <1-4096>** specifies an ACT ID from 1 to 4096.

2. Ensure the configuration is correct:

   ```
   show filter acl info [<1-4096>]
   ```

3. Associate ports or VLANs to the ACL as required.
4. Configure the ACL actions as required.
5. Ensure that the ACL is enabled:

   ```
   filter acl <1-4096> enable
   ```

Variable definitions

Use the information in the following table to use the **filter acl <1-4096>** command.

Variable	Value
`enable`	Enables the ACL state, and all associated ACEs. Enable is the default state.
`name <WORD 0-32>`	Specifies an optional descriptive name for the ACL.
`pktType <ipv4|ipv6>`	Specifies the IP version. The default is IPv4.
`type <inVlan| outVlan|inPort| outPort>`	Specifies the ACL type. **inVlan** and **inPort** are ingress ACLs, and **outVlan** and **outPort** are egress ACLs.

Configuring global and default actions for an ACL

Configure the default packet treatment when a packet does not match an ACE.

Configure the global packet treatment when a packet does match an ACE.

Comments? infodev@avaya.com

Prerequisites

- The ACL exists.
- Enter Global Configuration mode.

Procedure steps

1. Configure the global action for an ACL:

   ```
   filter acl set <1-4096> global-action <count|count-ipfix|
   ipfix|mirror|mirror-count|mirror-count-ipfix|mirror-ipfix>
   ```

2. Configure the default action for an ACL:

   ```
   filter acl set <1-4096> default-action <permit|deny>
   ```

Variable definitions

Use the information in the following table to use the **`filter acl set <1-4096>`** commands.

Variable	Value
`default-action <deny|permit>`	Specifies the default action to take when no ACEs match. Options include **`<deny|permit>`**. The default is permit.
`global-action <count|count-ipfix| ipfix|mirror| mirror-count| mirror-count-ipfix| mirror-ipfix>`	Specifies the global action for matching ACEs: mirror, count, mirror-count, ipfix, mirror-ipfix, count-ipfix, or mirror-count-ipfix. If you enable mirroring, ensure you specify the source or destination mirroring ports: • For R modules in Tx mode, use **`mirror-by-port`** commands to specify mirroring ports. • For RS and 8800 modules, or R modules in Rx mode, use the **`filter acl ace debug`** commands to specify mirroring ports. The default is none. To use the default configuration, use the default option in the command **`default filter acl set <1-4096> global-action`**

Associating VLANs with an ACL

Associate VLANs with, or remove VLANs from, an ACL so that filters do or do not apply to VLAN traffic, respectively.

Prerequisites

- The ACL exists.
- Enter Global Configuration mode.

Procedure steps

1. Associate VLANs with an ACL:

   ```
   filter acl vlan <1-4096> <1-4094>
   ```

2. Remove VLANs from an ACL:

   ```
   no filter acl vlan <1-4096> <1-4094>
   ```

Variable definitions

Use the information in the following table to use the commands in this procedure.

Variable	Value
<1-4096>	Specifies an ACL ID from 1–4096.
<1-4094>	Specifies the VLAN IDs from 1–4094.

Associating ports with an ACL

Associate ports with, or remove ports from, an ACL so that filters do or do not apply to port traffic, respectively.

Prerequisites

- The ACL exists.
- Enter Global Configuration mode.

Procedure steps

1. Associate ports with an ACL:

   ```
   filter acl port <1-4096> <portList>
   ```

2. Remove ports from an ACL:

   ```
   no filter acl port <1-4096> <portList>
   ```

Variable definitions

Use the information in the following table to use the commands in this procedure.

Variable	Value
<1-4096>	Specifies an ACL ID from 1–4096.
<portList>	Specifies ports in one of the following formats: [<slot/port>] or [<slot/port-slot/port>].

Viewing filter configuration information

View configuration information for ACL-based filters.

Procedure steps

1. View configuration information about ACLs:

   ```
   show filter acl
   ```

2. View configuration information about ACTs:

```
show filter act
```

3. View configuration information about ACT patterns:

```
show filter act-pattern
```

Variable definitions

Use the information in the following table to use the **show** command.

Variable	Value
`mode <value>`	Shows filter configuration output in either CLI or ACLI mode. **<value>** is cli or acli.
`verbose`	Shows detailed output.

Job aid

This sections shows the **show config module filter** command output.

```
ERS-8606:5# show config module filter
Preparing to Display Configuration... #
# MON APR 14 11:05:31 2008 UTC
# box type : ERS-8006
# software version : REL4.2.0.0_B157
# monitor version : 4.2.0.0/157
# cli mode : 8600 CLI
#
#
# Asic Info :
# SlotNum|Name |CardType |MdaType |Parts Description
#
# Slot 1 -- 0x00000001 0x00000000
# Slot 2 -- 0x00000001 0x00000000
# Slot 3 8630GBR 0x2432511e 0x00000000 RSP=25 CLUE=2 F2I=1 F2E=1
FTMUX=17 CC= 3 FOQ=266 DPC=184 BMC=776 PIM=257 MAC=4
# Slot 4 8648GTR 0x24220130 0x00000000 RSP=25 CLUE=2 F2I=1 F2E=1
FTMUX=0 CC=3 FOQ=266 DPC=6 BMC=776 PIM=257 MAC=4
# Slot 5 8692SF 0x200e0100 0x00000000 CPU: CPLD=19 MEZZ=4 SFM:
OP=3 TMUX=2 SWIP=23 FAD=16 CF=28
# Slot 6 -- 0x00000001 0x00000000 config
#
# R-MODULE FILTER CONFIGURATION
#
filter act 1 create name "ACT-1ADV"
filter act 1 ethernet srcMac
filter act 1 ip srcIp
filter act 1 protocol tcpSrcPort
filter act 1 apply filter act 2 create name "ACT-2AD VS"
filter act 2 pattern kelie add ip-hdr-begin 0 1
filter act 2 apply
filter acl 1 create inPort act 1
filter acl 1 set global-action mirror-count
filter acl 1 ace 1 create name "Adv"
filter acl 1 ace 1 action permit filter acl 1 ace 1 debug
copytoprimarycp enable
filter acl 2 create inPort act 2
```


Comments? infodev@avaya.com

```
filter acl 2 ace 1 create name "KB"
filter acl 2 ace 1 action permit remark-dot1p five
back
ERS-8606:5#
```

Comments? infodev@avaya.com

Chapter 18: Access control entry configuration using the ACLI

Use an ACE to provide an ordered list of traffic filtering rules.

Job aid

The following roadmap lists traffic filter commands that you can use to perform the procedures in this section.

Table 33: Roadmap of traffic filter ACLI commands

Command	Parameters
Global Configuration mode	
`filter acl ace <1-4096> <1-1000>`	`enable`
	`name <WORD 0-32>`
`filter acl ace action <1-4096> <1-1000> <deny|permit>`	`egress-queue <0-64>`
	`egress-queue-adssc <bronze|critical|custom|gold|platimum|premium|silver|standard>`
	`ipfix enable`
	`mlt-index <0-8>`
	`police <0-16383>`
	`redirect-next-hop <WORD 1-15>`
	`remark-dot1p <0-8>|zero|one|two|three|four|five|six|seven>`
	`remark-dscp <0-256>|phbcs0|phbcs1|phbaf11|phbaf12|phbaf13|phbcs2| phbaf21|phbaf22|phbaf23|phbcs3|phbaf31|phbaf32|phbaf33|phbcs4|phbaf41|phbaf42|phbaf43|phbcs5|phbef|phbcs6|phbcs7>`

Command	Parameters
	stop-on-match enable
	unreachable <deny|permit>
filter acl ace advanced <1-4096> <1-1000>	custom-filter1 <WORD 0-32> <eq|le|ge> <WORD 1-1024>
	custom-filter2 <WORD 0-32> <eq|le|ge> <WORD 1-1024>
	custom-filter3 <WORD 0-32> <eq|le|ge> <WORD 1-1024>
filter acl ace arp <1-4096> <1-1000> operation eq <arprequest|arpresponse>	–
filter acl ace ethernet <1-4096> <1-1000>	dst-mac <eq|ne|le|ge> <WORD 1-1024>
	ether-type <eq|ne> <WORD 1-200>
	port <eq> <portList>
	src-mac <eq|ne|le|ge> <WORD 1-1024>
	vlan-id <eq> <1..4094[,<1..4094>...]>
	vlan-tag-prio <eq|ne> <0-7>
filter acl ace ip <1-4096> <1-1000>	dscp <eq|ne> <0-256>|phbcs0|phbcs1|phbaf11|phbaf12|phbaf13|phbcs2|phbaf21|phbaf22|phbaf23|phbcs3|phbaf31|phbaf32|phbaf33|phbcs4|phbaf41|phbaf42|phbaf43|phbcs5|phbcs6|phbef|phbcs7>
	dst-ip <eq|ne|le|ge> <WORD 1-1024>
	ip-frag-flag <eq> <noFragment|anyFragment|moreFragment|lastFragment>
	ip-options any
	ip-protocol-type <eq|ne> <WORD 1-256>
	src-ip <eq|ne|le|ge> <WORD 1-1024>

Comments? infodev@avaya.com

Command	Parameters
filter acl ace ipv6 <1-4096> <1-1000>	dst-ipv6 <eq> <WORD 0-255>
	nxt-hdr <eq|ne> <fragment|hop-by-hop|ipsecesp|ipsecah|icmpv6|noHdr|routing|tcp|udp|undefined>
	src-ipv6 <eq> <WORD 0-255>
filter acl ace protocol <1-4096> <1-1000>	icmp-msg-type <eq|ne> <WORD 1-200>
	tcp-dst-port <eq|ne|le|ge> <WORD 1-60>
	tcp-flags <match-any|match-all> <fin|syn|rst|push|ack|urg>
	tcp-src-port <eq|ne|le|ge> <WORD 0-65535>
	udp-dst-port <eq|ne|le|ge> <WORD 1-200>
	udp-src-port <eq|ne|le|ge> <WORD 0-65535>
filter acl ace debug <1-4096> <1-1000>	copy-to-primary-cp enable
	copy-to-secondary-cp enable
	count enable
	mirror enable
	monitor-dst-ports <portList>
	monitor-dst-vlan <0-4094>
	monitor-dst-mlt <1-256>

Configuring ACEs

Use an access control entry (ACE) to define a packet pattern and the desired behavior for packets that carry the pattern.

ACEs of type inVlan with an ACT that includes srcIp, and with an ACL default action of deny, require additional configuration to function properly. See Workaround for inVlan, srcIp ACL on page 351 for the CLI commands for this special configuration.

Alternatively, Avaya recommends that you create ACLs with a default action of permit, and with an ACE mode of deny. For deny and permit ACLs and ACEs, the default action and the mode must be opposite for the ACE (filter) to have meaning.

Prerequisites

- The ACL exists.
- Enter Global Configuration mode.

Procedure steps

1. Create and configure an access control entry :

   ```
   filter acl ace <1-4096> <1-1000> [name <WORD 0-32]
   ```

   The ACE ID determines ACE precedence (that is, the lower the ID, the higher the precedence).

   <1-1000> specifies an ACE ID from 1 to 1000; **<1-4096>** specifies an ACL ID from 1 to 4096.

2. Configure the ACE action mode as deny or permit:

   ```
   filter acl ace action <1-4096> <1-1000> <deny|permit>
   ```

3. Configure ACE actions as required.
4. Ensure the configuration is correct:

   ```
   show filter acl ace [<1-4096>] [<1-1000>]
   ```

5. Ensure the filter is enabled:

   ```
   filter acl ace <1-4096> <1-1000> enable
   ```

Variable definitions

Use the information in the following table to use the **filter acl ace <1-4096> <1-1000>** and the **filter acl ace action <1-4096> <1-1000>** commands.

Variable	Value
<deny|permit>	Configures the action mode. The default is deny. To use the default configuration, use the default option in the command **default filter acl ace action <1-4096> <1-1000>**

Comments? infodev@avaya.com

Variable	Value
`debug`	Updates desired debug parameters for ACEs.
`enable`	Enables an ACE within an ACL. After you enable an ACE, to make changes, first disable it.
`name <WORD 0-32>`	Specifies an optional descriptive name for the ACE that uses 0–32 characters.

Configuring ACE actions

Actions determine the process that occurs when a packet matches an ACE.

Prerequisites

- The ACE exists.
- Enter Global Configuration mode.
- To use a policer, a policy exists.

Procedure steps

1. Configure ACE actions:

   ```
   filter acl ace action <1-4096> <1-1000> <deny|permit>
   ```

2. Ensure the configuration is correct:

   ```
   show filter acl action [<1-4096>] [<1-1000>]
   ```

Variable definitions

Use the information in the following table to use the **`filter acl ace action <1-4096> <1-1000> <deny|permit>`** commands.

Variable	Value
`egress-queue <0-63>`	Specifies the offset from the base queue number (0–63). **`<0-63>`** can be one, two, or three values.. The first value specifies the Egress Queue ID for the 8648GTR, 8648GTRS, 8848GT, 8648GBRS, 8848GB, and

Variable	Value
	gigabit ports of the 8634XGRS and 8834XG modules. The second value specifies the Egress Queue ID for the 8630GBR, 8612XLRS, 8812XL, and 10 Gb ports of the 8634XGRS and the 8834XG modules. The third specifies the Egress Queue ID for 8683XLR and 8683XZR modules. If you specify only one value, the same value applies to all module types. If you specify two values, the first value applies to 8648GTR, 8648GTRS, 8848GT, 8648GBRS, 8848GB and gigabit ports of 8634XGRS, 8834XG, and the second value applies to 8630GBR, 8612XLRS, 8812XL, and 10 Gb ports of the 8634XGRS and the 8834XG modules. If you specify all three values, the three values apply to the respective module types as explained in the preceding paragraph.
`egress-queue-adssc <bronze|critical|custom|gold|platimum|premium|silver|standard>`	Specifies the ADSSC egress queue value.
`ipfix enable`	Enables IPFIX. The default is disabled. To use the default configuration, use the default option in the command **`default filter acl ace action <1-4096> <1-1000> ipfix enable`**
`mlt-index <0-8>`	If you specify this action, the ACE overrides the mlt-index chosen by the MLT algorithm for packets sent on MLT ports. The MLT index ranges from 0–8. If three ports exist in an MLT (for example, A, B, and C) and you specify an index of 6, the Avaya Ethernet Routing Switch 8800/8600 applies the MOD function and chooses port C. If port C becomes nonoperational, the filtered packets exit from port B. Multicast traffic does not support the MLT index.
`police <0-16383>`	Specifies the policy ID of the policer (0–16383). A policy must exist.
`redirect-next-hop <WORD 1-15>`	Specifies the next-hop IP address for redirect mode (a.b.c.d). If you specify the next-hop IPv6 address for redirect mode, enter 0.0.0.0 <IPv6 address>.
`remark-dscp <WORD 0-256>`	Specifies the new Per-Hop Behavior for matching packets: phbcs0, phbcs1, phbaf11, phbaf12, phbaf13, phbcs2, phbaf21, phbaf22, phbaf23, phbcs3, phbaf31, phbaf32, phbaf33, phbcs4, phbaf41, phbaf42, phbaf43, phbcs5, phbef, phbcs6, phbcs7.

Variable	Value
`remark-dot1p <WORD 0-256>`	Specifies the new 802.1 priority bit for matching packets: zero, one, two, three, four, five, six, or seven.
`stop-on-match enable`	Enables the stop-on-match option. This option specifies whether to stop or continue after an ACE matches the packet. After this ACE matches, the switch does not attempt a match on other ACEs with lower priority.
`unreachable <deny|permit>`	Denies or permits packet dropping when the next-hop for the packet is unreachable. The default is deny. To use the default configuration, use the default option in the command **`default filter acl ace action <1-4096> <1-1000> unreachable`**

Example of configuring ACE actions

1. Configure actions:

   ```
   ERS-8610:6# filter acl ace action 1 1 permit ipfix enable
   remark-dscp phbaf22
   ```

Configuring ACE debug actions

Use debug actions to use filters for troubleshooting or monitoring procedures.

Caution:

Risk of packet loss

Avaya recommends that you do not select copyToPrimaryCp or copyToSecondaryCp. If you select the copyToPrimaryCp parameter, the switch sends packets to the CP, which can overload it. You can use the Packet Capture Tool (PCAP), rather than select the parameter copyToPrimaryCp.

If you use the mirror action, ensure that you specify the mirroring destination: MLTs, ports, or VLANs.

Prerequisites

- The ACE exists.
- Enter Global Configuration mode.

Procedure steps

1. Configure debug actions for an ACE:

   ```
   filter acl ace debug <1-4096> <1-1000> [count enable] [copy-to-primary-cp enable] [copy-to-secondary-cp enable] [mirror enable] [monitor-dst-ports <portList>] [monitor-dst-vlan <0-4094>] [monitor-dst-mlt <1-256>]
   ```

2. Ensure the configuration is correct:

   ```
   show filter acl debug [<1-4096>] [<1-1000>]
   ```

Variable definitions

Use the information in the following table to use the **`filter acl ace debug <1-4096> <1-1000>`** commands.

Variable	Value
`copy-to-primary-cp enable`	Enables the ability to copy matching packets to the primary (Master) CPU. The default is disabled. To use the default configuration, use the default option in the command **`default filter acl ace debug <1-4096> <1-1000> copy-to-primary-cp enable`**
`copy-to-secondary-cp enable`	Enables the ability to copy matching packets to the secondary (Standby) CPU. The default is disabled. To use the default configuration, use the default option in the command **`default filter acl ace debug <1-4096> <1-1000> copy-to-secondary-cp enable`**
`count enable`	Enables the ability to count matching packets. The default is disabled. To use the default configuration, use the default option in the command **`default filter acl ace debug <1-4096> <1-1000> count enable`**
`mirror enable`	Enables mirroring. If you enable mirroring, ensure that you configure the appropriate parameters:

Comments? infodev@avaya.com

Variable	Value
	• For R, RS, and 8800 modules in Rx mode, and for RS and 8800 modules, use**monitor-dst-ports**, **monitor-dst-vlan**, or **monitor-dst-mlt**. • For R modules in Tx mode, use the **mirror-by-port** commands to specify the mirroring source or destination. The default is disabled. To use the default configuration, use the default option in the command **default filter acl ace debug <1-4096> <1-1000> mirror enable**
`monitor-dst-ports <portList>`	Configures mirroring to a destination port or ports.
`monitor-dst-mlt <1-256>`	Configures mirroring to a destination MLT group.
`monitor-dst-vlan <0-4094>`	Configures mirroring to a destination VLAN.

Configuring ARP ACEs

Use ACE ARP entries so that the filter looks for ARP requests or responses.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has ARP attributes.
- Enter Global Configuration mode.

Procedure steps

1. Configure an ACE for ARP packets:

   ```
   filter acl ace arp <1-4096> <1-1000> operation eq
   <arprequest|arpresponse>
   ```

2. Ensure the configuration is correct:

```
show filter acl arp [<1-4096>] [<1-1000>]
```

Variable definitions

Use the following table to use the **`filter acl ace arp`** commands.

Variable	Value
`operation eq <arprequest|arpresponse>`	Specifies an ARP operation type of arpRequest or arpResponse. For ARP, only one operator and attribute exist (eq and operation).

Configuring an Ethernet ACE

Use Ethernet ACEs to filter on Ethernet parameters.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has Ethernet attributes.
- Enter Global Configuration mode.

Procedure steps

1. Configure an ACE with Ethernet header attributes:

   ```
   filter acl ace ethernet <1-4096> <1-1000>
   ```

2. Ensure the configuration is correct:

   ```
   show filter acl ethernet [<1-4096>] [<1-1000>]
   ```

Variable definitions

Use the following table to use the **`filter acl ace ethernet <1-4096> <1-1000>`** commands.

Comments? infodev@avaya.com

Variable	Value
`dst-mac <eq|ne|le|ge> <WORD 1-1024>`	The **<eq|ne|le|ge>** parameter specifies an operator for a field match condition: equal to, not equal to, less than or equal to, greater than or equal to. The **<WORD 1-1024>** parameter specifies a list of destination MAC addresses separated by a comma, or a range of MAC addresses specified from low to high; for example, [a:b:c:d:e:f, (x:y:z:w:v:u-a:b:c:d:e:f)].
`ether-type <eq|ne> <WORD 1-200>`	The **<eq|ne>** parameter specifies an operator for a field match condition: equal to or not equal to. The **<WORD 1-200>** parameter specifies an ether-type name or number: • 0–65563 • ip, arp, ipx802dot3, ipx802dot2, ipxSnap, ipxEthernet2, appleTalk, decLat, decOther, sna802dot2, snaEthernet2, netBios, xns, vines, ipv6, rarp, or PPPoE
`port eq <portList>`	Specifies ports to which to match, where **<portList>** specifies the ports.
`src-mac <eq|ne|le|ge> <WORD 1-1024>`	The **<eq|ne|le|ge>** parameter specifies an operator for a field match condition: equal to, not equal to, less than or equal to, greater than or equal to. The **<WORD 1-1024>** parameter specifies a list of source MAC addresses separated by separated by a comma, or a range of MAC addresses specified from low to high; for example, [a:b:c:d:e:f, (x:y:z:w:v:u- a:b:c:d:e:f)].
`vlan-id eq <1-4094>`	Specifies VLANs to match, where **<1-4094>** specifies the VLAN IDs.
`vlan-tag-prio <eq|ne> <0-7>`	The **<eq|ne>** parameter specifies an operator for a field match condition: equal to or not equal to. The **<vlan-tag-prio>** parameter specifies a VLAN tag priority from 0–7 or undefined.

Example of configuring an Ethernet ACE

1. Specify a specific destination MAC address:

```
ERS-8610:6# filter acl ace ethernet 1 12 dst-mac eq
08:00:69:02:01:FC
```

Configuring an IP ACE

Use IP ACEs to filter on the source IP address, destination IP address, DiffServ Code Point (DSCP), protocol, IP options, and IP fragmentation parameters.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has IP attributes.
- Enter Global Configuration mode.

Procedure steps

1. Configure an ACE with IP header attributes:

   ```
   filter acl ace ip <1-4096> <1-1000>
   ```

2. Ensure the configuration is correct:

   ```
   show filter acl ip [<1-4096>] [<1-1000>]
   ```

Variable definitions

Use the following table to use the **`filter acl ace ip <1-4096> <1-1000>`** commands.

Variable	Value
`dst-ip <eq|ne|le|ge> <WORD 1-1024>`	The **`<eq|ne|le|ge>`** parameter specifies an operator for a field match condition: equal to, not equal to, less than or equal to, greater than or equal to. The **`<WORD 1-1024>`** parameter specifies the destination IP address list in one of the following formats: a.b.c.d, [w.x.y.z-p.q.r.s], [l.m.n.o/mask], [a.b.c.d/len].

Comments? infodev@avaya.com

Variable	Value
`dscp <eq|ne> <WORD 0-256>`	The **<eq|ne>** parameter specifies an operator for a field match condition: equal to or not equal to. The **<WORD 0-256>** parameter specifies the PHB name or DSCP value {0 to 256}, or phbcs0, phbcs1, phbaf11, phbaf12, phbaf13, phbcs2, phbaf21, phbaf22, phbaf23, phbcs3, phbaf31, phbaf32, phbaf33, phbcs4, phbaf41, phbaf42, phbaf43, phbcs5, phbcs6, phbef, or phbcs.
`ip-frag-flag eq <noFragment| anyFragment| moreFragment| lastFragment>`	The **eq** parameter specifies an operator for a field match condition: equal to. The **ip-frag-flag** parameter specifies a match option for IP fragments (0, 2, or 4), or noFragment, anyFragment, moreFragment, lastFragment.
`ip-options any`	Matches to an IP option. Any is the only option.
`ip-protocol-type <eq|ne> <WORD 1-256>`	The **<eq|ne>** parameter specifies an operator for a field match condition: equal to or not equal to. The **<WORD 1-256>** parameter specifies one or more IP protocol types: (1–256), or undefined, icmp, tcp, udp, ipsecesp, ipsecah, ospf, vrrp, snmp.
`src-ip <eq|ne|le| ge> <WORD 1-1024>`	The **<eq|ne|le|ge>** parameter specifies an operator for a field match condition: equal to, not equal to, less than or equal to, greater than or equal to. The **<WORD 1-1024>** parameter specifies a source IP address list in one of the following formats: a.b.c.d, [w.x.y.z-p.q.r.s], [l.m.n.o/mask], [a.b.c.d/len].

Example of configuring an IP ACE

1. Specify a specific destination IP address:

   ```
   ERS-8610:6# filter acl ace ip 1 12 dst-ip eq 121.202.2.3
   ```

Configuring a protocol ACE

Use protocol ACEs to filter on the TCP source port, UDP source port, TCP destination port, UDP destination port, ICMP message type, and TCP flags.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has protocol attributes.
- Enter Global Configuration mode.

Procedure steps

1. Configure an ACE with protocol attributes:

   ```
   filter acl ace protocol <1-4096> <1-1000>
   ```

2. Ensure the configuration is correct:

   ```
   show filter acl protocol [<1-4096>] [<1-1000>]
   ```

Variable definitions

Use the information in the following table to use the **`filter acl ace protocol <1-4096> <1-1000>`** commands.

Variable	Value
`icmp-msg-type <eq|ne> <WORD 1-200>`	The **`<eq|ne>`** parameter specifies an operator for a field match condition: equal to or not equal to. The **`<WORD 1-200>`** parameter specifies one or more IP protocol types (0–255), or echoreply, destunreach, sourcequench, redirect, echo-request, routeradv, routerselect, time-exceeded, param-problem, timestamp-request, timestamp-reply, addressmask-request, addressmask-reply, or traceroute.
`tcp-dst-port <eq|ne|le|ge> <WORD 1-60>`	The **`<eq|ne|le|ge>`** parameter specifies an operator for a field match condition: equal to, not equal to, less than or equal to, greater than or equal to. The **`<WORD 1-60>`** parameter specifies the destination port for the TCP protocol: (0–65535), or echo, ftpdata, ftpcontrol, ssh, telnet, dns, http, bgp, hdot323, or undefined.

Comments? infodev@avaya.com

Variable	Value
`tcp-flags <match-any|match-all> <WORD>`	Specifies matchAny or matchAll operators for a field match condition. The **<WORD>** parameter specifies one or more TCP flags: none, fin, syn, rst, push, ack, urg, undefined. The tcp-flags and icmp-msg-type command options support lists.
`tcp-src-port <eq|ne|le|ge> <WORD 0-65535>`	The **<eq|ne|le|ge>** parameter specifies an operator for a field match condition: equal to, not equal to, less than or equal to, greater than or equal to. The **<WORD 0-65535>** parameter specifies the destination port for the TCP protocol (0–65535), or echo, dns, bootpServer, bootpClient, tftp, rip, rtp, rtcp, or undefined.
`udp-dst-port <eq|ne|le|ge> <WORD 1-200>`	The **<eq|ne|le|ge>** parameter specifies an operator for a field match condition: equal to, not equal to, less than or equal to, greater than or equal to. The **<WORD 1-200>** parameter specifies the destination port for the UDP protocol (0–65535), or echo, dns, bootpServer, bootpClient, tftp, rip, rtp, rtcp, or undefined.
`udp-src-port <eq|ne|le|ge> <WORD 0-65535>`	The **<eq|ne|le|ge>** parameter specifies an operator for a field match condition: equal to, not equal to, less than or equal to, greater than or equal to. The **<WORD 0-65535>** parameter specifies the source port for the UDP protocol (0–65535), or [ ].

Example of configuring a protocol ACE

1. Specify ICMP packets:

   ```
   ERS-8610:6# filter acl ace protocol 1 12 icmp-msg-type eq echo-request
   ```

Configuring a custom ACE

You can use a custom ACE to define your own match patterns.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has pattern attributes.
- Enter Global Configuration mode.

Procedure steps

1. Add an ACE for patterns that you define:

   ```
   filter acl ace advanced <1-4096> <1-1000>
   ```

2. Ensure that your configuration is correct:

   ```
   show filter acl advanced [<1-4096>] [<1-1000>]
   ```

Variable definitions

Use the following table to use the **`filter acl ace advanced <1-4096> <1-1000>`** commands.

Variable	Value
`custom-filter1 <WORD 0-32> <eq|le|ge> <WORD 1-1024>`	Creates a custom filter 1: • **`<WORD 0-32>`** specifies a descriptive name for the pattern that uses 0–32 characters. • **`<eq|le|ge>`** specifies the operators equal to, less than or equal to, or greater than or equal to. The ace-op ne does not apply to an ACE pattern. • **`<WORD 1-1024>`** specifies a hexadecimal number equal to the pattern template length.
`custom-filter2 <WORD 0-32> <eq|le|ge> <WORD 1-1024>`	Creates custom filter 2.
`custom-filter3 <WORD 0-32> <eq|le|ge> <WORD 1-1024>`	Creates custom filter 3.

Example of configuring a custom ACE

1. Add an ACE for patterns that you define:

   ```
   ERS-8610:6# filter acl ace advanced 1 12 custom-filter1
   PatternName eq 0x12
   ```

Configuring an IPv6 ACE

Use an IPv6 ACE to filter on IPv6 attributes.

Prerequisites

- The ACE exists.
- The ACL exists.
- The ACT has IPv6 attributes.
- Enter Global Configuration mode.

Procedure steps

1. Add an ACE with IP header attributes:

   ```
   filter acl ace ipv6 <1-4096> <1-1000>
   ```

2. Ensure that your configuration is correct:

   ```
   show filter acl ipv6 [<1-4096>] [<1-1000>]
   ```

Variable definitions

Use the information in the following table to use the **`filter acl ace ipv6 <1-4096> <1-1000>`** commands.

Variable	Value
`dst-ipv6 <eq> <WORD 0-255>`	The **<eq|ne>** parameter specifies an operator for a field match condition: equal to or not equal to.

Variable	Value
	The **<WORD 0-255>** parameter specifies a list of destination IPv6 addresses, separated by commas. An example IPv6 address is 3ffe:1900:4545:3:200:f8ff:fe21:67cf.
`nxt-hdr <eq|ne> <nxt-hdr>`	The **<eq|ne>** parameter specifies an operator for a field match condition: equal to or not equal to. **<nxt-hdr>** specifies hop-by-hop, tcp, udp, routing, fragment, ipsecesp, ipsecah, icmpv6, noHdr, or undefined.
`src-ipv6 <eq> <WORD 0-255>`	The **<eq|ne>** parameter specifies an operator for a field match condition: equal to or not equal to. The **<WORD 0-255>** parameter specifies a list of source IPv6 addresses, separated by commas. An example IPv6 address is 3ffe:1900:4545:3:200:f8ff:fe21:67cf.

Example of configuring an IPv6 ACE

1. Add an ACE with IP header attributes:

   ```
   ERS-8610:6# filter acl ace ipv6 1 12 dst-ipv6 eq 3ffe:
   1900:4545:3:200:f8ff:fe21:67cf
   ```

Viewing ACL and ACE configuration data

Review your configuration to ensure that it is correct.

Prerequisites

- Enter Privileged EXEC mode.

Procedure steps

1. View a list of executed commands:

Comments? infodev@avaya.com

```
show filter acl config [<1-4096>] [<1-1000>]
```

Variable definitions

Use the data in the following table to use the **`show filter acl config`** command.

Variable	Value
`<1-1000>`	Specifies an ACE ID from 1–1000.
`<1-4096>`	Specifies an ACL ID from 1–4096.

Comments? infodev@avaya.com

Chapter 19: Safety messages

This section describes the various precautionary notices used in this document. This section also contains precautionary notices that you must read for safe operation of the Avaya Ethernet Routing Switch 8800/8600.

Notices

Notice paragraphs alert you about issues that require your attention. The following sections describe the types of notices.

Attention notice

Important:

An attention notice provides important information regarding the installation and operation of Avaya products.

Caution ESD notice

Electrostatic alert:

ESD

ESD notices provide information about how to avoid discharge of static electricity and subsequent damage to Avaya products.

Electrostatic alert:

ESD (décharge électrostatique)

La mention ESD fournit des informations sur les moyens de prévenir une décharge électrostatique et d'éviter d'endommager les produits Avaya.

Electrostatic alert:

ACHTUNG ESD

ESD-Hinweise bieten Information dazu, wie man die Entladung von statischer Elektrizität und Folgeschäden an Avaya-Produkten verhindert.

Electrostatic alert:

PRECAUCIÓN ESD (Descarga electrostática)

El aviso de ESD brinda información acerca de cómo evitar una descarga de electricidad estática y el daño posterior a los productos Avaya.

Electrostatic alert:

CUIDADO ESD

Os avisos do ESD oferecem informações sobre como evitar descarga de eletricidade estática e os conseqüentes danos aos produtos da Avaya.

Electrostatic alert:

ATTENZIONE ESD

Le indicazioni ESD forniscono informazioni per evitare scariche di elettricità statica e i danni correlati per i prodotti Avaya.

Caution notice

Caution:

Caution notices provide information about how to avoid possible service disruption or damage to Avaya products.

Caution:

ATTENTION

La mention Attention fournit des informations sur les moyens de prévenir une perturbation possible du service et d'éviter d'endommager les produits Avaya.

Caution:

ACHTUNG

Achtungshinweise bieten Informationen dazu, wie man mögliche Dienstunterbrechungen oder Schäden an Avaya-Produkten verhindert.

Caution:

PRECAUCIÓN

Los avisos de Precaución brindan información acerca de cómo evitar posibles interrupciones del servicio o el daño a los productos Avaya.

Caution:

CUIDADO

Os avisos de cuidado oferecem informações sobre como evitar possíveis interrupções do serviço ou danos aos produtos da Avaya.

 **Caution:**

ATTENZIONE

Le indicazioni di attenzione forniscono informazioni per evitare possibili interruzioni del servizio o danni ai prodotti Avaya.

Comments? infodev@avaya.com

Chapter 20: Customer Service

Visit the Avaya Web site to access the complete range of services and support that Avaya provides. Go to www.avaya.com or go to one of the pages listed in the following sections.

Getting technical documentation

To download and print selected technical publications and release notes directly from the Internet, go to www.avaya.com/support.

Getting product training

Ongoing product training is available. For more information or to register, you can access the Web site at www.avaya.com/support. From this Web site, you can locate the Training contacts link on the left-hand navigation pane.

Getting help from a distributor or reseller

If you purchased a service contract for your Avaya product from a distributor or authorized reseller, contact the technical support staff for that distributor or reseller for assistance.

Getting technical support from the Avaya Web site

The easiest and most effective way to get technical support for Avaya products is from the Avaya Technical Support Web site at www.avaya.com/support.

Appendix A: Advanced filter examples

This appendix gives a detailed Advanced filter configuration example.

ACE filters for secure networks

The following example shows filters configured for two Layer 2 switched hosts and two Layer 3 routed hosts for an IP phone and computer VLAN network.

These filters apply after an analysis of the traffic types flowing on the network. The filters provide security by permitting legitimate traffic and denying (dropping) all other traffic. Filters redirect certain traffic to another IP address. Further, use IPFIX and counting for reporting and monitoring. The filters can also determine which traffic to permit on which parts of the network.

The ACEs named DENY ANY or DENY ANY ANY are the cleanup filters. These filters drop traffic that does not match other ACEs.

Through the use of Ethereal, you determine that ACEs permit (this is not an exhaustive list) the following traffic types:

- DNS traffic
- ICMP traffic
- IGMP traffic
- VRRP traffic (in certain areas)
- BootStrap Protocol server and client traffic
- DHCP traffic
- NetBIOS traffic (in certain areas)
- TCP traffic with the Established flag set
- traffic with specific IP addresses
- Microsoft Operations Manager 2005 agent (MOM 2005) traffic
- HTTP, HTTP proxy, and HTTPS traffic
- remote desktop traffic
- ISAKMP and Internet Key Exchange (IKE) traffic
- SQL database system traffic

Other ACEs deny (drop) the following traffic types:

- VRRP traffic (in certain areas)
- NetBIOS traffic (UDP destination ports 137, 138)
- specific multicast traffic (UDP destination ports 61011, 64046)
- specific UDP traffic
- instant messaging traffic (UDP destination port 1900)

This section shows the filters configured for the first Layer 2 switched host.

```
#
# R-MODULE FILTER CONFIGURATION
#
filter act 1 create name "BUSINESS 1"
filter act 1 ip srcIp,dstIp,ipOptions,ipProtoType
filter act 1 protocol
tcpSrcPort,udpSrcPort,tcpDstPort,udpDstPort,tcpFlags,icmpMsgType
filter act 1 apply

filter acl 1 create outPort act 1 name "VRRP_Drop"
filter acl 1 port add 4/24-4/25,8/37
filter acl 1 ace 1 create name "VRRP"
filter acl 1 ace 1 action deny stop-on-match true
filter acl 1 ace 1 debug count enable
filter acl 1 ace 1 ip ip-protocol-type eq vrrp
filter acl 1 ace 1 enable
filter acl 1 ace 2 create name "NetbIOS_Drop"
filter acl 1 ace 2 action deny stop-on-match true
filter acl 1 ace 2 debug count enable
filter acl 1 ace 2 ip ip-protocol-type eq udp
filter acl 1 ace 2 protocol udp-dst-port eq 137
filter acl 1 ace 2 enable
filter acl 1 ace 3 create name "NetbIOS2_Drop"
filter acl 1 ace 3 action deny stop-on-match true
filter acl 1 ace 3 debug count enable
```

```
filter acl 1 ace 3 ip ip-protocol-type eq udp
filter acl 1 ace 3 protocol udp-dst-port eq 138
filter acl 1 ace 3 enable
filter acl 1 ace 4 create name "WL_Multicast1_Drop"
filter acl 1 ace 4 action deny stop-on-match true
filter acl 1 ace 4 debug count enable
filter acl 1 ace 4 ip ip-protocol-type eq udp
filter acl 1 ace 4 protocol udp-dst-port eq 61011
filter acl 1 ace 4 enable
filter acl 1 ace 5 create name "WL_Multicast2_Drop"
filter acl 1 ace 5 action deny stop-on-match true
filter acl 1 ace 5 debug count enable
filter acl 1 ace 5 ip ip-protocol-type eq udp
filter acl 1 ace 5 protocol udp-dst-port eq 64046
filter acl 1 ace 5 enable
filter acl 1 ace 6 create name "UDP_1100_Drop"
filter acl 1 ace 6 action deny stop-on-match true
filter acl 1 ace 6 ip dst-ip eq 100.20.100.255
filter acl 1 ace 6 ip ip-protocol-type eq udp
filter acl 1 ace 6 protocol udp-dst-port eq 1100
filter acl 1 ace 6 enable
filter acl 1 ace 7 create name "UDP_67_Drop"
filter acl 1 ace 7 action deny stop-on-match true
filter acl 1 ace 7 ip ip-protocol-type eq udp
filter acl 1 ace 7 protocol udp-dst-port eq 67
filter acl 1 ace 7 enable
filter acl 1 ace 8 create name "Messenger"
filter acl 1 ace 8 action deny stop-on-match true
filter acl 1 ace 8 ip ip-protocol-type eq udp
filter acl 1 ace 8 protocol udp-dst-port eq 1900
```

```
filter acl 1 ace 8 enable filter acl 20 create inVlan act 1 name
"Symantec-Drop"

filter acl 20 vlan add 2
filter acl 20 ace 10 create name "Othello-drop"
filter acl 20 ace 10 action deny stop-on-match true
filter acl 20 ace 10 debug count enable
filter acl 20 ace 10 ip src-ip eq 100.20.2.47
filter acl 20 ace 10 ip ip-protocol-type eq tcp
filter acl 20 ace 10 protocol tcp-src-port eq 80
filter acl 20 ace 10 enable
filter acl 20 ace 15 create name "Macbeth-drop"
filter acl 20 ace 15 action deny stop-on-match true
filter acl 20 ace 15 debug count enable
filter acl 20 ace 15 ip src-ip eq 100.20.2.29
filter acl 20 ace 15 ip ip-protocol-type eq tcp
filter acl 20 ace 15 protocol tcp-src-port eq 80

filter acl 902 create inVlan act 1 name "ITD_REMOTE_in"
filter acl 902 vlan add 902 filter acl 902 disable
filter acl 902 ace 5 create name "ITD_TO_ITD"
filter acl 902 ace 5 action permit stop-on-match true
filter acl 902 ace 5 ip dst-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 5 enable
filter acl 902 ace 10 create name "ICMP_PERMIT"
filter acl 902 ace 10 action permit stop-on-match true
filter acl 902 ace 10 ip ip-protocol-type eq icmp
filter acl 902 ace 10 enable
filter acl 902 ace 20 create name "IGMP_PERMIT"
filter acl 902 ace 20 action permit stop-on-match true
filter acl 902 ace 20 ip ip-protocol-type eq 2
filter acl 902 ace 20 enable
```

```
filter acl 902 ace 30 create name "VRRP_PERMIT"
filter acl 902 ace 30 action permit stop-on-match true
filter acl 902 ace 30 ip ip-protocol-type eq vrrp
filter acl 902 ace 30 enable
filter acl 902 ace 35 create name "BOOTPS"
filter acl 902 ace 35 action permit stop-on-match true
filter acl 902 ace 35 protocol udp-dst-port eq 67
filter acl 902 ace 35 enable filter acl 902 ace 36 create name
"BOOTPC"
filter acl 902 ace 36 action permit stop-on-match true
filter acl 902 ace 36 protocol udp-dst-port eq 68
filter acl 902 ace 36 enable
filter acl 902 ace 40 create name "DNS_PERMIT"
filter acl 902 ace 40 action permit stop-on-match true
filter acl 902 ace 40 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 40 protocol udp-dst-port eq dns
filter acl 902 ace 40 enable filter acl 902 ace 43 create name
"Netbios_Erisim"
filter acl 902 ace 43 action permit stop-on-match true
filter acl 902 ace 43 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 43 protocol udp-dst-port eq 135
filter acl 902 ace 43 enable
filter acl 902 ace 45 create name "ESTABLISHED"
filter acl 902 ace 45 action permit stop-on-match true
filter acl 902 ace 45 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 45 ip ip-protocol-type eq tcp
filter acl 902 ace 45 protocol tcp-dst-port ge 1023
filter acl 902 ace 45 protocol tcp-flags match-any rst,ack
filter acl 902 ace 45 enable filter acl 902 ace 50 create name "DC-
EXCH-DNS"
filter acl 902 ace 50 action permit stop-on-match true
filter acl 902 ace 50 ip src-ip eq 100.20.103.65-100.20.103.78
```

```
filter acl 902 ace 50 ip dst-ip eq 100.20.104.0-100.20.105.255
filter acl 902 ace 50 enable filter acl 902 ace 55 create name "DC-
EXCH-DNS_OPC"
filter acl 902 ace 55 action permit stop-on-match true
filter acl 902 ace 55 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 55 ip dst-ip eq 100.6.105.0-100.6.105.15
filter acl 902 ace 55 enable filter acl 902 ace 60 create name
"Filesharing_Erisim"
filter acl 902 ace 60 action permit stop-on-match true
filter acl 902 ace 60 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 60 ip dst-ip eq 100.20.103.71-100.20.103.72
filter acl 902 ace 60 enable
filter acl 902 ace 65 create name "Filesharing_Erisim_Ek"
filter acl 902 ace 65 action permit stop-on-match true
filter acl 902 ace 65 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 65 ip dst-ip eq 10.10.230.6
filter acl 902 ace 65 enable filter acl 902 ace 70 create name
"IBPSQL_Erisim"
filter acl 902 ace 70 action permit stop-on-match true
filter acl 902 ace 70 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 70 ip dst-ip eq 100.20.100.176
filter acl 902 ace 70 ip ip-protocol-type eq tcp
filter acl 902 ace 70 protocol tcp-dst-port eq 4450
filter acl 902 ace 70 enable
filter acl 902 ace 75 create name "CTI_Erisim"
filter acl 902 ace 75 action permit stop-on-match true
filter acl 902 ace 75 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 75 ip dst-ip eq 100.6.100.161
filter acl 902 ace 75 ip ip-protocol-type eq tcp
filter acl 902 ace 75 protocol tcp-dst-port eq 1433
filter acl 902 ace 75 enable
filter acl 902 ace 80 create name "PVA_ERISIM"
```

```
filter acl 902 ace 80 action permit stop-on-match true
filter acl 902 ace 80 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 80 ip dst-ip eq 100.6.100.138
filter acl 902 ace 80 ip ip-protocol-type eq tcp
filter acl 902 ace 80 protocol tcp-dst-port eq 1521
filter acl 902 ace 80 enable
filter acl 902 ace 85 create name "PWC_ERISIM"
filter acl 902 ace 85 action permit stop-on-match true
filter acl 902 ace 85 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 85 ip dst-ip eq 100.6.100.113
filter acl 902 ace 85 ip ip-protocol-type eq tcp
filter acl 902 ace 85 protocol tcp-dst-port eq 1521
filter acl 902 ace 85 enable
filter acl 902 ace 90 create name "OASIS_ERISIM"
filter acl 902 ace 90 action permit stop-on-match true
filter acl 902 ace 90 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 90 ip dst-ip eq 100.6.100.112
filter acl 902 ace 90 ip ip-protocol-type eq tcp
filter acl 902 ace 90 protocol tcp-dst-port eq 1521
filter acl 902 ace 90 enable
filter acl 902 ace 95 create name "AV-YAMA_YONETIM__9968"
filter acl 902 ace 95 action permit stop-on-match true
filter acl 902 ace 95 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 95 ip ip-protocol-type eq tcp
filter acl 902 ace 95 protocol tcp-dst-port eq 9968
filter acl 902 ace 95 enable
filter acl 902 ace 100 create name "AV-YAMA_YONETIM_2967"
filter acl 902 ace 100 action permit stop-on-match true
filter acl 902 ace 100 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 100 ip ip-protocol-type eq tcp
filter acl 902 ace 100 protocol tcp-dst-port eq 2967
```

```
filter acl 902 ace 100 enable
filter acl 902 ace 105 create name "AV-YAMA_YONETIM_UDP_2967"
filter acl 902 ace 105 action permit stop-on-match true
filter acl 902 ace 105 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 105 ip ip-protocol-type eq udp
filter acl 902 ace 105 protocol udp-dst-port eq 2967
filter acl 902 ace 105 enable
filter acl 902 ace 108 create name "AV-YAMA_YONETIM_SOURCE_9968"
filter acl 902 ace 108 action permit stop-on-match true
filter acl 902 ace 108 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 108 ip ip-protocol-type eq udp
filter acl 902 ace 108 protocol udp-src-port eq 9968
filter acl 902 ace 108 enable
filter acl 902 ace 110 create name "ALERT_MOM_SMS_ERISIM_TCP_1270"
filter acl 902 ace 110 action permit stop-on-match true
filter acl 902 ace 110 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 110 ip dst-ip eq 100.6.140.10-100.6.140.11
filter acl 902 ace 110 ip ip-protocol-type eq tcp
filter acl 902 ace 110 protocol tcp-dst-port eq 1270
filter acl 902 ace 110 enable
filter acl 902 ace 120 create name "ALERT_MOM_SMS_ERISIM_UDP_1270"
filter acl 902 ace 120 action permit stop-on-match true
filter acl 902 ace 120 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 120 ip dst-ip eq 100.6.140.10-100.6.140.11
filter acl 902 ace 120 ip ip-protocol-type eq udp
filter acl 902 ace 120 protocol udp-dst-port eq 1270
filter acl 902 ace 120 enable
filter acl 902 ace 130 create name "ALERT_MOM_SMS_ERISIM_HTTP"
filter acl 902 ace 130 action permit stop-on-match true
filter acl 902 ace 130 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 130 ip dst-ip eq 100.6.140.13
```

```
filter acl 902 ace 130 ip ip-protocol-type eq tcp
filter acl 902 ace 130 protocol tcp-dst-port eq 80
filter acl 902 ace 130 enable
filter acl 902 ace 135 create name "ALERT_MOM_SMS_ERISIM_HTTP2"
filter acl 902 ace 135 action permit stop-on-match true
filter acl 902 ace 135 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 135 ip dst-ip eq 100.6.106.92
filter acl 902 ace 135 ip ip-protocol-type eq tcp
filter acl 902 ace 135 protocol tcp-dst-port eq 80
filter acl 902 ace 135 enable
filter acl 902 ace 140 create name "ALERT_MOM_SMS_ERISIM_1521"
filter acl 902 ace 140 action permit stop-on-match true
filter acl 902 ace 140 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 140 ip dst-ip eq 100.6.100.126
filter acl 902 ace 140 ip ip-protocol-type eq tcp
filter acl 902 ace 140 protocol tcp-dst-port eq 1521
filter acl 902 ace 140 enable
filter acl 902 ace 150 create name "ALERT_MOM_SMS_ERISIM_1521x"
filter acl 902 ace 150 action permit stop-on-match true
filter acl 902 ace 150 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 150 ip dst-ip eq 100.20.100.47
filter acl 902 ace 150 ip ip-protocol-type eq tcp
filter acl 902 ace 150 protocol tcp-dst-port eq 1521
filter acl 902 ace 150 enable
filter acl 902 ace 155 create name "FULL_ERISIM"
filter acl 902 ace 155 action permit stop-on-match true
filter acl 902 ace 155 ip dst-ip eq 100.20.100.149
filter acl 902 ace 155 enable
filter acl 902 ace 160 create name "LOGLAMAK_ICIN"
filter acl 902 ace 160 action permit redirect-next-hop 100.20.150.34
stop-on-match true
```

```
filter acl 902 ace 160 ip src-ip ge 0.0.0.0
filter acl 902 ace 170 create name "DENY_ANY_ANY"
filter acl 902 ace 170 action deny stop-on-match true
filter acl 902 ace 170 ip src-ip ge 0.0.0.0
filter acl 902 ace 170 ip dst-ip ge 0.0.0.0
filter acl 902 ace 170 enable
```

The following section provides details about the filter configuration for the second switched Layer 2 host.

```
#
# R-MODULE FILTER CONFIGURATION
#
filter act 1 create name "BUSINESS 1"
filter act 1 ip srcIp,dstIp,ipOptions,ipProtoType
filter act 1 protocol
tcpSrcPort,udpSrcPort,tcpDstPort,udpDstPort,tcpFlags,icmpMsgType
filter act 1 apply

filter acl 1 create outPort act 1 name "VRRP Drop"
filter acl 1 port add 4/24-4/25,8/37
filter acl 1 ace 1 create name "VRRP"
filter acl 1 ace 1 action deny stop-on-match true
filter acl 1 ace 1 ip ip-protocol-type eq vrrp
filter acl 1 ace 1 enable
filter acl 1 ace 2 create name "NetbIOS_Drop"
filter acl 1 ace 2 action deny stop-on-match true
filter acl 1 ace 2 ip ip-protocol-type eq udp
filter acl 1 ace 2 protocol udp-dst-port eq 137
filter acl 1 ace 2 enable
filter acl 1 ace 3 create name "NetbIOS2_Drop"
filter acl 1 ace 3 action deny stop-on-match true
filter acl 1 ace 3 ip ip-protocol-type eq udp
filter acl 1 ace 3 protocol udp-dst-port eq 138
```

```
filter acl 1 ace 3 enable filter acl 1 ace 4 create name
"WL_Multicast1_Drop"

filter acl 1 ace 4 action deny stop-on-match true

filter acl 1 ace 4 ip ip-protocol-type eq udp

filter acl 1 ace 4 protocol udp-dst-port eq 61011

filter acl 1 ace 4 enable

filter acl 1 ace 5 create name "WL_Multicast2_Drop"

filter acl 1 ace 5 action deny stop-on-match true

filter acl 1 ace 5 ip ip-protocol-type eq udp

filter acl 1 ace 5 protocol udp-dst-port eq 64046

filter acl 1 ace 5 enable filter acl 20 create inVlan act 1 name
"Symantec-Drop"

filter acl 20 vlan add 2

filter acl 20 ace 10 create name "Othello-drop"

filter acl 20 ace 10 action deny stop-on-match true

filter acl 20 ace 10 debug count enable

filter acl 20 ace 10 ip src-ip eq 100.20.2.47

filter acl 20 ace 10 ip ip-protocol-type eq tcp

filter acl 20 ace 10 protocol tcp-src-port eq 80

filter acl 20 ace 10 enable

filter acl 20 ace 15 create name "Macbeth-drop"

filter acl 20 ace 15 action deny stop-on-match true

filter acl 20 ace 15 debug count enable

filter acl 20 ace 15 ip src-ip eq 100.20.2.29

filter acl 20 ace 15 ip ip-protocol-type eq tcp

filter acl 20 ace 15 protocol tcp-src-port eq 80

filter acl 902 create inVlan act 1 name "ITD_REMOTE_in"

filter acl 902 vlan add 902 filter acl 902 disable

filter acl 902 ace 5 create name "ITD_TO_ITD"

filter acl 902 ace 5 action permit stop-on-match true

filter acl 902 ace 5 ip dst-ip eq 100.20.103.65-100.20.103.78
```

```
filter acl 902 ace 5 enable
filter acl 902 ace 10 create name "ICMP_PERMIT"
filter acl 902 ace 10 action permit stop-on-match true
filter acl 902 ace 10 ip ip-protocol-type eq icmp
filter acl 902 ace 10 enable
filter acl 902 ace 20 create name "IGMP_PERMIT"
filter acl 902 ace 20 action permit stop-on-match true
filter acl 902 ace 20 ip ip-protocol-type eq 2
filter acl 902 ace 20 enable filter acl 902 ace 30 create name
"VRRP_PERMIT"
filter acl 902 ace 30 action permit stop-on-match true
filter acl 902 ace 30 ip ip-protocol-type eq vrrp
filter acl 902 ace 30 enable
filter acl 902 ace 35 create name "BOOTPS"
filter acl 902 ace 35 action permit stop-on-match true
filter acl 902 ace 35 protocol udp-dst-port eq 67
filter acl 902 ace 35 enable
filter acl 902 ace 36 create name "BOOTPC"
filter acl 902 ace 36 action permit stop-on-match true
filter acl 902 ace 36 protocol udp-dst-port eq 68
filter acl 902 ace 36 enable
filter acl 902 ace 40 create name "DNS_PERMIT"
filter acl 902 ace 40 action permit stop-on-match true
filter acl 902 ace 40 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 40 protocol udp-dst-port eq dns
filter acl 902 ace 40 enable
filter acl 902 ace 43 create name "Netbios_Erisim"
filter acl 902 ace 43 action permit stop-on-match true
filter acl 902 ace 43 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 43 protocol udp-dst-port eq 135
filter acl 902 ace 43 enable
```

```
filter acl 902 ace 45 create name "ESTABLISHED"
filter acl 902 ace 45 action permit stop-on-match true
filter acl 902 ace 45 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 45 ip ip-protocol-type eq tcp
filter acl 902 ace 45 protocol tcp-dst-port ge 1023
filter acl 902 ace 45 protocol tcp-flags match-any rst,ack
filter acl 902 ace 45 enable
filter acl 902 ace 50 create name "DC-EXCH-DNS"
filter acl 902 ace 50 action permit stop-on-match true
filter acl 902 ace 50 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 50 ip dst-ip eq 100.20.104.0-100.20.105.255
filter acl 902 ace 50 enable
filter acl 902 ace 55 create name "DC-EXCH-DNS_OPC"
filter acl 902 ace 55 action permit stop-on-match true
filter acl 902 ace 55 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 55 ip dst-ip eq 100.6.105.0-100.6.105.15
filter acl 902 ace 55 enable
filter acl 902 ace 60 create name "Filesharing_Erisim"
filter acl 902 ace 60 action permit stop-on-match true
filter acl 902 ace 60 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 60 ip dst-ip eq 100.20.103.71-100.20.103.72
filter acl 902 ace 60 enable
filter acl 902 ace 65 create name "Filesharing_Erisim_Ek"
filter acl 902 ace 65 action permit stop-on-match true
filter acl 902 ace 65 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 65 ip dst-ip eq 10.10.230.6
filter acl 902 ace 65 enable
filter acl 902 ace 70 create name "IBPSQL_Erisim"
filter acl 902 ace 70 action permit stop-on-match true
filter acl 902 ace 70 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 70 ip dst-ip eq 100.20.100.176
```

```
filter acl 902 ace 70 ip ip-protocol-type eq tcp
filter acl 902 ace 70 protocol tcp-dst-port eq 4450
filter acl 902 ace 70 enable
filter acl 902 ace 75 create name "CTI_Erisim"
filter acl 902 ace 75 action permit stop-on-match true
filter acl 902 ace 75 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 75 ip dst-ip eq 100.6.100.161
filter acl 902 ace 75 ip ip-protocol-type eq tcp
filter acl 902 ace 75 protocol tcp-dst-port eq 1433
filter acl 902 ace 75 enable
filter acl 902 ace 80 create name "PVA_ERISIM"
filter acl 902 ace 80 action permit stop-on-match true
filter acl 902 ace 80 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 80 ip dst-ip eq 100.6.100.138
filter acl 902 ace 80 ip ip-protocol-type eq tcp
filter acl 902 ace 80 protocol tcp-dst-port eq 1521
filter acl 902 ace 80 enable
filter acl 902 ace 85 create name "PWC_ERISIM"
filter acl 902 ace 85 action permit stop-on-match true
filter acl 902 ace 85 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 85 ip dst-ip eq 100.6.100.113
filter acl 902 ace 85 ip ip-protocol-type eq tcp
filter acl 902 ace 85 protocol tcp-dst-port eq 1521
filter acl 902 ace 85 enable
filter acl 902 ace 90 create name "OASIS_ERISIM"
filter acl 902 ace 90 action permit stop-on-match true
filter acl 902 ace 90 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 90 ip dst-ip eq 100.6.100.112
filter acl 902 ace 90 ip ip-protocol-type eq tcp
filter acl 902 ace 90 protocol tcp-dst-port eq 1521
filter acl 902 ace 90 enable
```

```
filter acl 902 ace 95 create name "AV-YAMA_YONETIM__9968"
filter acl 902 ace 95 action permit stop-on-match true
filter acl 902 ace 95 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 95 ip ip-protocol-type eq tcp
filter acl 902 ace 95 protocol tcp-dst-port eq 9968
filter acl 902 ace 95 enable
filter acl 902 ace 100 create name "AV-YAMA_YONETIM_2967"
filter acl 902 ace 100 action permit stop-on-match true
filter acl 902 ace 100 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 100 ip ip-protocol-type eq tcp
filter acl 902 ace 100 protocol tcp-dst-port eq 2967
filter acl 902 ace 100 enable
filter acl 902 ace 105 create name "AV-YAMA_YONETIM_UDP_2967"
filter acl 902 ace 105 action permit stop-on-match true
filter acl 902 ace 105 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 105 ip ip-protocol-type eq udp
filter acl 902 ace 105 protocol udp-dst-port eq 2967
filter acl 902 ace 105 enable
filter acl 902 ace 108 create name "AV-YAMA_YONETIM_SOURCE_9968"
filter acl 902 ace 108 action permit stop-on-match true
filter acl 902 ace 108 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 108 ip ip-protocol-type eq udp
filter acl 902 ace 108 protocol udp-src-port eq 9968
filter acl 902 ace 108 enable
filter acl 902 ace 110 create name "ALERT_MOM_SMS_ERISIM_TCP_1270"
filter acl 902 ace 110 action permit stop-on-match true
filter acl 902 ace 110 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 110 ip dst-ip eq 100.6.140.10-100.6.140.11
filter acl 902 ace 110 ip ip-protocol-type eq tcp
filter acl 902 ace 110 protocol tcp-dst-port eq 1270
filter acl 902 ace 110 enable
```

```
filter acl 902 ace 120 create name "ALERT_MOM_SMS_ERISIM_UDP_1270"
filter acl 902 ace 120 action permit stop-on-match true
filter acl 902 ace 120 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 120 ip dst-ip eq 100.6.140.10-100.6.140.11
filter acl 902 ace 120 ip ip-protocol-type eq udp
filter acl 902 ace 120 protocol udp-dst-port eq 1270
filter acl 902 ace 120 enable
filter acl 902 ace 130 create name "ALERT_MOM_SMS_ERISIM_HTTP"
filter acl 902 ace 130 action permit stop-on-match true
filter acl 902 ace 130 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 130 ip dst-ip eq 100.6.140.13
filter acl 902 ace 130 ip ip-protocol-type eq tcp
filter acl 902 ace 130 protocol tcp-dst-port eq 80
filter acl 902 ace 130 enable
filter acl 902 ace 135 create name "ALERT_MOM_SMS_ERISIM_HTTP2"
filter acl 902 ace 135 action permit stop-on-match true
filter acl 902 ace 135 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 135 ip dst-ip eq 100.6.106.92
filter acl 902 ace 135 ip ip-protocol-type eq tcp
filter acl 902 ace 135 protocol tcp-dst-port eq 80
filter acl 902 ace 135 enable
filter acl 902 ace 140 create name "ALERT_MOM_SMS_ERISIM_1521"
filter acl 902 ace 140 action permit stop-on-match true
filter acl 902 ace 140 ip src-ip eq 100.20.103.65-100.20.103.78
filter acl 902 ace 140 ip dst-ip eq 100.6.100.126
filter acl 902 ace 140 ip ip-protocol-type eq tcp
filter acl 902 ace 140 protocol tcp-dst-port eq 1521
filter acl 902 ace 140 enable
filter acl 902 ace 150 create name "ALERT_MOM_SMS_ERISIM_1521x"
filter acl 902 ace 150 action permit stop-on-match true
filter acl 902 ace 150 ip src-ip eq 100.20.103.65-100.20.103.78
```

```
filter acl 902 ace 150 ip dst-ip eq 100.20.100.47
filter acl 902 ace 150 ip ip-protocol-type eq tcp
filter acl 902 ace 150 protocol tcp-dst-port eq 1521
filter acl 902 ace 150 enable
filter acl 902 ace 155 create name "FULL_ERISIM"
filter acl 902 ace 155 action permit stop-on-match true
filter acl 902 ace 155 ip dst-ip eq 100.20.100.149
filter acl 902 ace 155 enable
filter acl 902 ace 160 create name "LOGLAMAK_ICIN"
filter acl 902 ace 160 action permit redirect-next-hop 100.20.150.34
stop-on-match true
filter acl 902 ace 160 ip src-ip ge 0.0.0.0
filter acl 902 ace 170 create name "DENY_ANY_ANY"
filter acl 902 ace 170 action deny stop-on-match true
filter acl 902 ace 170 ip src-ip ge 0.0.0.0
filter acl 902 ace 170 ip dst-ip ge 0.0.0.0
filter acl 902 ace 170 enable
```

The following section provides details about the filter configuration for the first core Layer 3 host.

```
#
# R-MODULE FILTER CONFIGURATION
#
filter act 1 create name "BUSINESS 1"
filter act 1 ip srcIp,dstIp,ipOptions,ipProtoType
filter act 1 protocol
tcpSrcPort,udpSrcPort,tcpDstPort,udpDstPort,tcpFlags,icmpMsgType
filter act 1 apply

filter acl 1 create outPort act 1 name "VRRP_Drop_ACL"
filter acl 1 port add 4/46
filter acl 1 ace 1 create name "Vrrp"
filter acl 1 ace 1 action deny stop-on-match true
filter acl 1 ace 1 ip ip-protocol-type eq vrrp
```

```
filter acl 1 ace 1 enable
filter acl 171 create inVlan act 1 name "TOPLANTI_VE_EGITIM_ACL"
filter acl 171 vlan add 171
filter acl 171 disable filter acl 171 ace 10 create name
"ICMP_PERMIT"
filter acl 171 ace 10 action permit stop-on-match true
filter acl 171 ace 10 ip ip-protocol-type eq icmp
filter acl 171 ace 10 enable
filter acl 171 ace 20 create name "IGMP_PERMIT"
filter acl 171 ace 20 action permit stop-on-match true
filter acl 171 ace 20 ip ip-protocol-type eq 2
filter acl 171 ace 20 enable
filter acl 171 ace 30 create name "VRRP_PERMIT"
filter acl 171 ace 30 action permit stop-on-match true
filter acl 171 ace 30 ip ip-protocol-type eq vrrp
filter acl 171 ace 30 enable
filter acl 171 ace 40 create name "DNS_PERMIT"
filter acl 171 ace 40 action permit stop-on-match true
filter acl 171 ace 40 ip src-ip eq 100.20.171.0-100.20.171.255
filter acl 171 ace 40 ip dst-ip eq 100.20.104.0-100.20.104.255
filter acl 171 ace 40 protocol udp-dst-port eq dns
filter acl 171 ace 40 enable
filter acl 171 ace 50 create name "ESTABLISHED"
filter acl 171 ace 50 action permit stop-on-match true
filter acl 171 ace 50 ip src-ip eq 100.6.172.0-100.6.172.255
filter acl 171 ace 50 ip ip-protocol-type eq tcp
filter acl 171 ace 50 protocol tcp-dst-port ge 1023
filter acl 171 ace 50 protocol tcp-flags match-any rst,ack
filter acl 171 ace 50 enable
filter acl 171 ace 60 create name "DHCP_PERMIT"
filter acl 171 ace 60 action permit stop-on-match true
```

```
filter acl 171 ace 60 protocol udp-dst-port eq bootpServer
filter acl 171 ace 60 enable
filter acl 171 ace 80 create name "DC_DNS_EXC_PERMIT"
filter acl 171 ace 80 action permit stop-on-match true
filter acl 171 ace 80 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 171 ace 80 ip dst-ip eq 100.20.104.0-100.20.105.255
filter acl 171 ace 80 enable
filter acl 171 ace 90 create name "HTTP_PERMIT"
filter acl 171 ace 90 action permit stop-on-match true
filter acl 171 ace 90 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 171 ace 90 protocol tcp-dst-port eq 80
filter acl 171 ace 90 enable
filter acl 171 ace 100 create name "HTTPS_PERMIT"
filter acl 171 ace 100 action permit stop-on-match true
filter acl 171 ace 100 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 171 ace 100 protocol tcp-dst-port eq 443
filter acl 171 ace 100 enable
filter acl 171 ace 110 create name "PROXY_8080_PERMIT"
filter acl 171 ace 110 action permit stop-on-match true
filter acl 171 ace 110 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 171 ace 110 ip dst-ip eq 100.20.189.0-100.20.189.255
filter acl 171 ace 110 protocol tcp-dst-port eq 8080
filter acl 171 ace 110 enable
filter acl 171 ace 120 create name "CITRIX_Conn"
filter acl 171 ace 120 action permit stop-on-match true
filter acl 171 ace 120 protocol tcp-dst-port eq 1494
filter acl 171 ace 120 protocol udp-dst-port eq 1604
filter acl 171 ace 120 enable
filter acl 171 ace 130 create name "PWC_VPN_ERISIM"
filter acl 171 ace 130 action permit stop-on-match true
filter acl 171 ace 130 ip src-ip eq 100.20.172.0-100.20.172.255
```

```
filter acl 171 ace 130 protocol tcp-dst-port eq 11160
filter acl 171 ace 130 enable
filter acl 171 ace 140 create name "Microsoft_FileSharing_PERMIT"
filter acl 171 ace 140 action permit stop-on-match true
filter acl 171 ace 140 debug count enable
filter acl 171 ace 140 protocol tcp-dst-port eq 135-139
filter acl 171 ace 140 protocol udp-dst-port eq 135-139
filter acl 171 ace 140 enable
filter acl 171 ace 150 create name "Microsoft_FileSharing_PERMIT"
filter acl 171 ace 150 action permit stop-on-match true
filter acl 171 ace 150 debug count enable
filter acl 171 ace 150 protocol tcp-dst-port eq 445
filter acl 171 ace 150 protocol udp-dst-port eq 445
filter acl 171 ace 150 enable

filter acl 172 create inVlan act 1 name "MISAFIR_ACL"
filter acl 172 vlan add 172
filter acl 172 disable
filter acl 172 ace 5 create name "Misafir_to_Misafir"
filter acl 172 ace 5 action permit stop-on-match true
filter acl 172 ace 5 ip dst-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 5 enable
filter acl 172 ace 10 create name "ICMP_PERMIT"
filter acl 172 ace 10 action permit stop-on-match true
filter acl 172 ace 10 ip ip-protocol-type eq icmp
filter acl 172 ace 10 enable
filter acl 172 ace 20 create name "IGMP_PERMIT"
filter acl 172 ace 20 action permit stop-on-match true
filter acl 172 ace 20 ip ip-protocol-type eq 2
filter acl 172 ace 20 enable
filter acl 172 ace 30 create name "VRRP_PERMIT"
```

```
filter acl 172 ace 30 action permit stop-on-match true
filter acl 172 ace 30 ip ip-protocol-type eq vrrp
filter acl 172 ace 30 enable
filter acl 172 ace 40 create name "DNS_PERMIT"
filter acl 172 ace 40 action permit stop-on-match true
filter acl 172 ace 40 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 40 ip dst-ip eq 100.20.104.0-100.20.104.255
filter acl 172 ace 40 protocol udp-dst-port eq dns filter acl 172 ace
40 enable
filter acl 172 ace 50 create name "ESTABLISHED"
filter acl 172 ace 50 action permit stop-on-match true
filter acl 172 ace 50 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 50 ip ip-protocol-type eq tcp
filter acl 172 ace 50 protocol tcp-dst-port ge 1023
filter acl 172 ace 50 protocol tcp-flags match-any rst,ack
filter acl 172 ace 50 enable
filter acl 172 ace 60 create name "DHCP_PERMIT"
filter acl 172 ace 60 action permit stop-on-match true
filter acl 172 ace 60 protocol udp-dst-port eq bootpServer
filter acl 172 ace 60 enable
filter acl 172 ace 80 create name "DC_DNS_EXC_PERMIT"
filter acl 172 ace 80 action permit stop-on-match true
filter acl 172 ace 80 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 80 ip dst-ip eq 100.20.104.0-100.20.105.255
filter acl 172 ace 80 enable
filter acl 172 ace 90 create name "HTTP_PERMIT"
filter acl 172 ace 90 action permit stop-on-match true
filter acl 172 ace 90 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 90 ip ip-protocol-type eq tcp
filter acl 172 ace 90 protocol tcp-dst-port eq 80
filter acl 172 ace 90 enable
```

```
filter acl 172 ace 100 create name "HTTPS_PERMIT"
filter acl 172 ace 100 action permit stop-on-match true
filter acl 172 ace 100 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 100 ip ip-protocol-type eq tcp
filter acl 172 ace 100 protocol tcp-dst-port eq 443
filter acl 172 ace 100 enable
filter acl 172 ace 105 create name "REMDESKTOP_PERMIT"
filter acl 172 ace 105 action permit stop-on-match true
filter acl 172 ace 105 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 105 ip ip-protocol-type eq tcp
filter acl 172 ace 105 protocol tcp-dst-port eq 3389
filter acl 172 ace 105 enable
filter acl 172 ace 106 create name "NORKOM_PERMIT"
filter acl 172 ace 106 action permit stop-on-match true
filter acl 172 ace 106 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 106 ip dst-ip eq
100.6.106.0-100.6.106.255,100.20.24.0-100.20.24.255
filter acl 172 ace 106 enable
filter acl 172 ace 107 create name "SPECTRUM_PERMIT"
filter acl 172 ace 107 action permit stop-on-match true
filter acl 172 ace 107 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 107 ip dst-ip eq 100.20.17.0-100.20.17.255
filter acl 172 ace 107 enable
filter acl 172 ace 110 create name "PROXY_8080_PERMIT"
filter acl 172 ace 110 action permit stop-on-match true
filter acl 172 ace 110 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 110 ip dst-ip eq 100.20.189.0-100.20.189.255
filter acl 172 ace 110 ip ip-protocol-type eq tcp
filter acl 172 ace 110 protocol tcp-dst-port eq 8080
filter acl 172 ace 110 enable filter acl 172 ace 120 create name
"CITRIX_Conn-tcp"
filter acl 172 ace 120 action permit stop-on-match true
```

```
filter acl 172 ace 120 ip ip-protocol-type eq tcp
filter acl 172 ace 120 protocol tcp-dst-port eq 1494
filter acl 172 ace 120 enable
filter acl 172 ace 121 create name "CITRIX_Conn-udp"
filter acl 172 ace 121 action permit stop-on-match true
filter acl 172 ace 121 ip ip-protocol-type eq udp
filter acl 172 ace 121 protocol udp-dst-port eq 1604
filter acl 172 ace 121 enable
filter acl 172 ace 128 create name "VOIP_VLAN_PERMIT"
filter acl 172 ace 128 action permit stop-on-match true
filter acl 172 ace 128 ip dst-ip eq 10.201.0.0-10.201.31.255
filter acl 172 ace 128 enable filter acl 172 ace 129 create name
"GANYMEDE-PERMIT"
filter acl 172 ace 129 action permit stop-on-match true
filter acl 172 ace 129 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 129 ip dst-ip eq 100.6.100.225
filter acl 172 ace 129 enable
filter acl 172 ace 130 create name "PWC_VPN_ERISIM"
filter acl 172 ace 130 action permit stop-on-match true
filter acl 172 ace 130 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 130 ip ip-protocol-type eq tcp
filter acl 172 ace 130 protocol tcp-dst-port eq 11160
filter acl 172 ace 130 enable
filter acl 172 ace 131 create name "ISAKMP"
filter acl 172 ace 131 action permit stop-on-match true
filter acl 172 ace 131 ip ip-protocol-type eq udp
filter acl 172 ace 131 protocol udp-dst-port eq 500
filter acl 172 ace 131 enable
filter acl 172 ace 132 create name "ESP"
filter acl 172 ace 132 action permit stop-on-match true
filter acl 172 ace 132 ip ip-protocol-type eq 50
```

```
filter acl 172 ace 132 enable

filter acl 172 ace 133 create name "LOGLAMAK_ICIN"

filter acl 172 ace 133 action permit redirect-next-hop 100.20.150.34
stop-on-match true ipfix enable

filter acl 172 ace 133 debug count enable

filter acl 172 ace 133 ip src-ip ge 0.0.0.0

filter acl 172 ace 140 create name "DENY_ANY_ANY"

filter acl 172 ace 140 action deny stop-on-match true

filter acl 172 ace 140 debug count enable

filter acl 172 ace 140 ip src-ip ge 0.0.0.0

filter acl 172 ace 140 ip dst-ip ge 0.0.0.0

filter acl 172 ace 140 enable

filter acl 802 create inVlan act 1 name "NICE-CLS_ACL-in"

filter acl 802 vlan add 802

filter acl 802 disable

filter acl 802 ace 1 create name "NICE_to_NICE"

filter acl 802 ace 1 action permit stop-on-match true

filter acl 802 ace 1 ip dst-ip eq 100.20.174.32-100.20.174.63

filter acl 802 ace 1 enable

filter acl 802 ace 10 create name "ICMP_PERMIT"

filter acl 802 ace 10 action permit stop-on-match true

filter acl 802 ace 10 ip ip-protocol-type eq icmp

filter acl 802 ace 10 enable

filter acl 802 ace 20 create name "IGMP_PERMIT"

filter acl 802 ace 20 action permit stop-on-match true

filter acl 802 ace 20 ip ip-protocol-type eq 2

filter acl 802 ace 20 enable filter acl 802 ace 30 create name
"VRRP_PERMIT"

filter acl 802 ace 30 action permit stop-on-match true

filter acl 802 ace 30 ip ip-protocol-type eq vrrp

filter acl 802 ace 30 enable

filter acl 802 ace 40 create name "DNS_PERMIT"
```

```
filter acl 802 ace 40 action permit stop-on-match true
filter acl 802 ace 40 ip src-ip eq 100.20.174.32-100.20.174.63
filter acl 802 ace 40 ip dst-ip eq 100.20.104.0-100.20.104.255
filter acl 802 ace 40 protocol udp-dst-port eq dns
filter acl 802 ace 40 enable
filter acl 802 ace 45 create name "DC-EXCH-DNS"
filter acl 802 ace 45 action permit stop-on-match true
filter acl 802 ace 45 ip dst-ip eq 100.20.104.0-100.20.105.255
filter acl 802 ace 45 enable
filter acl 802 ace 50 create name "ESTABLISHED"
filter acl 802 ace 50 action permit stop-on-match true
filter acl 802 ace 50 ip src-ip eq 100.20.174.32-100.20.174.63
filter acl 802 ace 50 ip ip-protocol-type eq tcp
filter acl 802 ace 50 protocol tcp-dst-port ge 1023
filter acl 802 ace 50 protocol tcp-flags match-any rst,ack
filter acl 802 ace 50 enable
filter acl 802 ace 51 create name "UDP_Permit"
filter acl 802 ace 51 action permit stop-on-match true
filter acl 802 ace 51 ip ip-protocol-type eq udp
filter acl 802 ace 51 enable
filter acl 802 ace 60 create name "NICE_Logging"
filter acl 802 ace 60 action permit stop-on-match true
filter acl 802 ace 60 ip src-ip eq 100.20.174.32-100.20.174.63
filter acl 802 ace 60 ip ip-protocol-type eq tcp
filter acl 802 ace 60 protocol tcp-dst-port eq 2011
filter acl 802 ace 60 enable
filter acl 802 ace 65 create name "RTS_Conn"
filter acl 802 ace 65 action permit stop-on-match true
filter acl 802 ace 65 ip dst-ip eq 100.20.152.20
filter acl 802 ace 65 enable
filter acl 802 ace 70 create name "CTI_Conn"
```

```
filter acl 802 ace 70 action permit stop-on-match true
filter acl 802 ace 70 ip src-ip eq 100.20.174.32-100.20.174.63
filter acl 802 ace 70 ip ip-protocol-type eq tcp
filter acl 802 ace 70 protocol tcp-dst-port eq 3750
filter acl 802 ace 70 enable
filter acl 802 ace 90 create name "LOGLAMA"
filter acl 802 ace 90 action permit redirect-next-hop 100.20.150.217
stop-on-match true
filter acl 802 ace 90 debug count enable
filter acl 802 ace 90 ip src-ip ge 0.0.0.0
filter acl 802 ace 100 create name "DENY_ANY"
filter acl 802 ace 100 action deny stop-on-match true
filter acl 802 ace 100 debug count enable
filter acl 802 ace 100 ip src-ip ge 0.0.0.0
filter acl 802 ace 100 ip dst-ip ge 0.0.0.0
filter acl 802 ace 100 enable

filter acl 804 create inVlan act 1 name "BASIM_LIMITED-in"
filter acl 804 vlan add 804
filter acl 804 ace 5 create name "Basim_to_Basim"
filter acl 804 ace 5 action permit stop-on-match true
filter acl 804 ace 5 ip dst-ip eq 100.20.174.96-100.20.174.127
filter acl 804 ace 5 enable
filter acl 804 ace 10 create name "ICMP_PERMIT"
filter acl 804 ace 10 action permit stop-on-match true
filter acl 804 ace 10 ip ip-protocol-type eq icmp
filter acl 804 ace 10 enable
filter acl 804 ace 20 create name "IGMP_PERMIT"
filter acl 804 ace 20 action permit stop-on-match true
filter acl 804 ace 20 ip ip-protocol-type eq 2
filter acl 804 ace 20 enable
filter acl 804 ace 30 create name "VRRP_PERMIT"
```

```
filter acl 804 ace 30 action permit stop-on-match true
filter acl 804 ace 30 ip ip-protocol-type eq vrrp
filter acl 804 ace 30 enable
filter acl 804 ace 40 create name "DNS_PERMIT"
filter acl 804 ace 40 action permit stop-on-match true
filter acl 804 ace 40 protocol udp-dst-port eq dns
filter acl 804 ace 40 enable
filter acl 804 ace 45 create name "DC-EXCH-DNS"
filter acl 804 ace 45 action permit stop-on-match true
filter acl 804 ace 45 ip dst-ip eq 100.20.104.0-100.20.105.255
filter acl 804 ace 45 enable
filter acl 804 ace 50 create name "ESTABLISHED"
filter acl 804 ace 50 action permit stop-on-match true
filter acl 804 ace 50 ip src-ip eq 100.20.174.97-100.20.174.127
filter acl 804 ace 50 ip ip-protocol-type eq tcp
filter acl 804 ace 50 protocol tcp-dst-port ge 1023
filter acl 804 ace 50 protocol tcp-flags match-any rst,ack
filter acl 804 ace 50 enable
filter acl 804 ace 60 create name "E-BANK_ERISIM"
filter acl 804 ace 60 action permit stop-on-match true
filter acl 804 ace 60 ip dst-ip eq 100.20.115.11
filter acl 804 ace 60 ip ip-protocol-type eq tcp
filter acl 804 ace 60 protocol tcp-dst-port eq 80
filter acl 804 ace 60 enable
filter acl 804 ace 70 create name "E-BANK_ERISIM_HTTPS"
filter acl 804 ace 70 action permit stop-on-match true
filter acl 804 ace 70 ip dst-ip eq 100.20.115.11
filter acl 804 ace 70 ip ip-protocol-type eq tcp
filter acl 804 ace 70 protocol tcp-dst-port eq 443
filter acl 804 ace 70 enable
filter acl 804 ace 80 create name "FRED_Erisim"
```

```
filter acl 804 ace 80 action permit stop-on-match true
filter acl 804 ace 80 ip dst-ip eq 100.20.100.145
filter acl 804 ace 80 enable
filter acl 804 ace 81 create name "BARNEY_Erisim"
filter acl 804 ace 81 action permit stop-on-match true
filter acl 804 ace 81 ip dst-ip eq 100.20.100.151
filter acl 804 ace 81 enable
filter acl 804 ace 90 create name "BUFFY_ERISIM"
filter acl 804 ace 90 action permit stop-on-match true
filter acl 804 ace 90 ip dst-ip eq 100.20.100.77
filter acl 804 ace 90 ip ip-protocol-type eq tcp
filter acl 804 ace 90 protocol tcp-dst-port eq 1433
filter acl 804 ace 90 enable
filter acl 804 ace 100 create name "ROMTest_ERISIM"
filter acl 804 ace 100 action permit stop-on-match true
filter acl 804 ace 100 ip dst-ip eq 100.20.24.77
filter acl 804 ace 100 ip ip-protocol-type eq tcp
filter acl 804 ace 100 protocol tcp-dst-port eq 1433
filter acl 804 ace 100 enable
filter acl 804 ace 101 create name "Mrksql-t0_ERISIM"
filter acl 804 ace 101 action permit stop-on-match true
filter acl 804 ace 101 ip dst-ip eq 100.20.20.77
filter acl 804 ace 101 ip ip-protocol-type eq tcp
filter acl 804 ace 101 protocol tcp-dst-port eq 1433
filter acl 804 ace 101 enable
filter acl 804 ace 110 create name "ROSETTA_ERISIM"
filter acl 804 ace 110 action permit stop-on-match true
filter acl 804 ace 110 ip dst-ip eq 172.17.1.100
filter acl 804 ace 110 enable
filter acl 804 ace 120 create name "PLAST_ERISIM"
filter acl 804 ace 120 action permit stop-on-match true
```

```
filter acl 804 ace 120 ip dst-ip eq 212.57.7.20
filter acl 804 ace 120 enable
filter acl 804 ace 130 create name "AV-Yama_YONETIM_2967"
filter acl 804 ace 130 action permit stop-on-match true
filter acl 804 ace 130 ip ip-protocol-type eq tcp
filter acl 804 ace 130 protocol tcp-dst-port eq 2967
filter acl 804 ace 130 enable
filter acl 804 ace 140 create name "AV-Yama_YONETIM_9968"
filter acl 804 ace 140 action permit stop-on-match true
filter acl 804 ace 140 ip ip-protocol-type eq tcp
filter acl 804 ace 140 protocol tcp-dst-port eq 9968
filter acl 804 ace 140 enable
filter acl 804 ace 150 create name "AV-Yama_YONETIM_UDP_2967"
filter acl 804 ace 150 action permit stop-on-match true
filter acl 804 ace 150 ip ip-protocol-type eq udp
filter acl 804 ace 150 protocol udp-dst-port eq 2967
filter acl 804 ace 150 enable
filter acl 804 ace 160 create name "AV-Yama_YONETIM_UDP_9968"
filter acl 804 ace 160 action permit stop-on-match true
filter acl 804 ace 160 ip ip-protocol-type eq udp
filter acl 804 ace 160 protocol udp-dst-port eq 9968
filter acl 804 ace 160 enable
filter acl 804 ace 170 create name "AV-Yama_YONETIM_UDP_Source"
filter acl 804 ace 170 action permit stop-on-match true
filter acl 804 ace 170 ip ip-protocol-type eq udp
filter acl 804 ace 170 protocol udp-src-port eq 9968
filter acl 804 ace 170 enable
filter acl 804 ace 210 create name "PROXY_ERISIM_EK"
filter acl 804 ace 210 action permit stop-on-match true
filter acl 804 ace 210 ip dst-ip eq 100.20.189.0-100.20.189.255
filter acl 804 ace 210 ip ip-protocol-type eq tcp
```

```
filter acl 804 ace 210 protocol tcp-dst-port eq 8080
filter acl 804 ace 210 enable
filter acl 804 ace 220 create name "LOGLAMA"
filter acl 804 ace 220 action permit redirect-next-hop 100.20.150.217
stop-on-match true
filter acl 804 ace 220 debug count enable
filter acl 804 ace 220 ip src-ip ge 0.0.0.0
filter acl 804 ace 230 create name "DENY_ANY"
filter acl 804 ace 230 action deny stop-on-match true
filter acl 804 ace 230 debug count enable
filter acl 804 ace 230 ip src-ip ge 0.0.0.0
filter acl 804 ace 230 ip dst-ip ge 0.0.0.0
filter acl 804 ace 230 enable

filter acl 805 create inVlan act 1 name "SBS-Remote"
filter acl 805 vlan add 805
filter acl 805 ace 5 create name "SBS-to-SBS"
filter acl 805 ace 5 action permit stop-on-match true
filter acl 805 ace 5 ip dst-ip eq 100.20.174.128-100.20.174.135
filter acl 805 ace enable
filter acl 805 ace 10 create name "ICMP_PERMIT"
filter acl 805 ace 10 action permit stop-on-match true
filter acl 805 ace 10 ip ip-protocol-type eq icmp
filter acl 805 ace 10 enable
filter acl 805 ace 20 create name "IGMP_PERMIT"
filter acl 805 ace 20 action permit stop-on-match true
filter acl 805 ace 20 ip ip-protocol-type eq 2
filter acl 805 ace 20 enable
filter acl 805 ace 30 create name "VRRP_PERMIT"
filter acl 805 ace 30 action permit stop-on-match true
filter acl 805 ace 30 ip ip-protocol-type eq vrrp
filter acl 805 ace 30 enable
```

```
filter acl 805 ace 40 create name "DNS_PERMIT"
filter acl 805 ace 40 action permit stop-on-match true
filter acl 805 ace 40 protocol udp-dst-port eq 53
filter acl 805 ace 40 enable
filter acl 805 ace 50 create name "ESTABLISHED"
filter acl 805 ace 50 action permit stop-on-match true
filter acl 805 ace 50 ip src-ip eq 100.20.174.128-100.20.174.134
filter acl 805 ace 50 ip ip-protocol-type eq tcp
filter acl 805 ace 50 protocol tcp-dst-port ge 1023
filter acl 805 ace 50 protocol tcp-flags match-any rst,ack
filter acl 805 ace 50 enable
filter acl 805 ace 80 create name "DC_DNS_EXCH_PERMIT"
filter acl 805 ace 80 action permit stop-on-match true
filter acl 805 ace 80 ip dst-ip eq 100.20.104.0-100.20.105.255
filter acl 805 ace 80 enable
filter acl 805 ace 90 create name "HTTP_PERMIT"
filter acl 805 ace 90 action permit stop-on-match true
filter acl 805 ace 90 ip ip-protocol-type eq tcp
filter acl 805 ace 90 protocol tcp-dst-port eq 80
filter acl 805 ace 90 enable
filter acl 805 ace 100 create name "HTTPS_PERMIT"
filter acl 805 ace 100 action permit stop-on-match true
filter acl 805 ace 100 ip ip-protocol-type eq tcp
filter acl 805 ace 100 protocol tcp-dst-port eq 443
filter acl 805 ace 100 enable
filter acl 805 ace 105 create name "REMDESKTOP_PERMIT"
filter acl 805 ace 105 action permit stop-on-match true
filter acl 805 ace 105 ip ip-protocol-type eq tcp
filter acl 805 ace 105 protocol tcp-dst-port eq 3389
filter acl 805 ace 105 enable
filter acl 805 ace 110 create name "PROXY_8080_PERMIT"
```

```
filter acl 805 ace 110 action permit stop-on-match true
filter acl 805 ace 110 ip dst-ip eq 100.20.189.0-100.20.189.255
filter acl 805 ace 110 ip ip-protocol-type eq tcp
filter acl 805 ace 110 protocol tcp-dst-port eq 8080
filter acl 805 ace 110 enable
filter acl 805 ace 120 create name "DAMEWARE_PERMIT" filter acl 805
ace 120 action permit
filter acl 805 ace 120 ip src-ip eq 100.20.174.128-100.20.174.134
filter acl 805 ace 120 protocol tcp-dst-port eq 445,6129
filter acl 805 ace 120 enable
filter acl 805 ace 140 create name "DENY_ANY_ANY"
filter acl 805 ace 140 action deny stop-on-match true
filter acl 805 ace 140 ip src-ip ge 0.0.0.0
filter acl 805 ace 140 ip dst-ip ge 0.0.0.0
filter acl 805 ace 140 enable

filter acl 1000 create inPort act 1 name "CS1K-RemDesk"
filter acl 1000 port add 4/33
filter acl 1000 ace 10 create name "ICMP"
filter acl 1000 ace 10 action permit stop-on-match true
filter acl 1000 ace 10 ip ip-protocol-type eq icmp
filter acl 1000 ace 10 enable
filter acl 1000 ace 15 create name "ESTABLISHED_PERMIT"
filter acl 1000 ace 15 action permit stop-on-match true
filter acl 1000 ace 15 protocol tcp-dst-port ge 1023
filter acl 1000 ace 15 protocol tcp-flags match-any rst,ack
filter acl 1000 ace 15 enable
filter acl 1000 ace 20 create name "LOGLAMAK_ICIN"
filter acl 1000 ace 20 action permit redirect-next-hop 10.201.12.8
stop-on-match true
filter acl 1000 ace 20 ip src-ip ge 0.0.0.0
filter acl 1000 ace 30 create name "DENY-ANY_ANY"
```

```
filter acl 1000 ace 30 action deny stop-on-match true
filter acl 1000 ace 30 ip src-ip ge 0.0.0.0
filter acl 1000 ace 30 enable

filter acl 1802 create outVlan act 1 name "NICE-CLS_ACL-out"
filter acl 1802 vlan add 802
filter acl 1802 disable
filter acl 1802 ace 10 create name "ICMP_PERMIT"
filter acl 1802 ace 10 action permit stop-on-match true
filter acl 1802 ace 10 ip ip-protocol-type eq icmp
filter acl 1802 ace 10 enable
filter acl 1802 ace 20 create name "IGMP_PERMIT"
filter acl 1802 ace 20 action permit stop-on-match true
filter acl 1802 ace 20 ip ip-protocol-type eq 2
filter acl 1802 ace 20 enable
filter acl 1802 ace 30 create name "VRRP_PERMIT"
filter acl 1802 ace 30 action permit stop-on-match true
filter acl 1802 ace 30 ip ip-protocol-type eq vrrp
filter acl 1802 ace 30 enable
filter acl 1802 ace 51 create name "UDP_Permit"
filter acl 1802 ace 51 action permit stop-on-match true
filter acl 1802 ace 51 ip ip-protocol-type eq udp
filter acl 1802 ace 51 enable
filter acl 1802 ace 60 create name "NICE_Logging"
filter acl 1802 ace 60 action permit stop-on-match true
filter acl 1802 ace 60 ip src-ip eq 100.20.174.32-100.20.174.63
filter acl 1802 ace 60 protocol tcp-dst-port eq 2011
filter acl 1802 ace 60 enable
filter acl 1802 ace 65 create name "RTS_Conn"
filter acl 1802 ace 65 action permit stop-on-match true
filter acl 1802 ace 100 create name "DENY_ANY"
```

```
filter acl 1802 ace 100 action deny stop-on-match true
filter acl 1802 ace 100 ip src-ip ge 0.0.0.0
filter acl 1802 ace 100 ip dst-ip ge 0.0.0.0
filter acl 1802 ace 100 enable

filter acl 1804 create outVlan act 1 name "BASIM_LIMITED-out"
filter acl 1804 vlan add 804
filter acl 1804 ace 5 create name "BASIM_to_BASIM"
filter acl 1804 ace 5 action permit stop-on-match true
filter acl 1804 ace 5 ip src-ip eq 100.20.174.96-100.20.174.127
filter acl 1804 ace 5 enable
filter acl 1804 ace 10 create name "ICMP_PERMIT"
filter acl 1804 ace 10 action permit stop-on-match true
filter acl 1804 ace 10 ip ip-protocol-type eq icmp
filter acl 1804 ace 10 enable
filter acl 1804 ace 20 create name "IGMP_PERMIT"
filter acl 1804 ace 20 action permit stop-on-match true
filter acl 1804 ace 20 ip ip-protocol-type eq 2
filter acl 1804 ace 20 enable
filter acl 1804 ace 30 create name "VRRP_PERMIT"
filter acl 1804 ace 30 action permit stop-on-match true
filter acl 1804 ace 30 ip ip-protocol-type eq vrrp
filter acl 1804 ace 30 enable
filter acl 1804 ace 40 create name "DNS_PERMIT"
filter acl 1804 ace 40 action permit stop-on-match true
filter acl 1804 ace 40 protocol udp-src-port eq 53
filter acl 1804 ace 40 enable
filter acl 1804 ace 45 create name "DC-EXCH-DNS"
filter acl 1804 ace 45 action permit stop-on-match true
filter acl 1804 ace 45 ip src-ip eq 100.20.104.0-100.20.105.255
filter acl 1804 ace 45 enable
```

```
filter acl 1804 ace 50 create name "ESTABLISHED"
filter acl 1804 ace 50 action permit stop-on-match true
filter acl 1804 ace 50 ip dst-ip eq 100.20.174.97-100.20.174.127
filter acl 1804 ace 50 ip ip-protocol-type eq tcp
filter acl 1804 ace 50 protocol tcp-dst-port ge 1023
filter acl 1804 ace 50 protocol tcp-flags match-any rst,ack
filter acl 1804 ace 50 enable
filter acl 1804 ace 80 create name "PWC_ERISIM"
filter acl 1804 ace 80 action permit stop-on-match true
filter acl 1804 ace 80 ip src-ip eq 100.20.100.145
filter acl 1804 ace 80 enable
filter acl 1804 ace 110 create name "ROSETTA_ERISIM"
filter acl 1804 ace 110 action permit stop-on-match true
filter acl 1804 ace 110 ip src-ip eq 172.17.1.100
filter acl 1804 ace 110 enable
filter acl 1804 ace 120 create name "PLAST_ERISIM"
filter acl 1804 ace 120 action permit stop-on-match true
filter acl 1804 ace 120 ip src-ip eq 212.57.7.20
filter acl 1804 ace 120 enable
filter acl 1804 ace 130 create name "AV-Yama_YONETIM_9968"
filter acl 1804 ace 130 action permit stop-on-match true
filter acl 1804 ace 130 ip ip-protocol-type eq tcp
filter acl 1804 ace 130 protocol tcp-dst-port eq 9968
filter acl 1804 ace 130 enable
filter acl 1804 ace 140 create name "AV-Yama_YONETIM_2967"
filter acl 1804 ace 140 action permit stop-on-match true
filter acl 1804 ace 140 ip ip-protocol-type eq tcp
filter acl 1804 ace 140 protocol tcp-dst-port eq 2967
filter acl 1804 ace 140 enable
filter acl 1804 ace 150 create name "AV-Yama_YONETIM_UDP_9968"
filter acl 1804 ace 150 action permit stop-on-match true
```

```
filter acl 1804 ace 150 ip ip-protocol-type eq udp
filter acl 1804 ace 150 protocol udp-dst-port eq 9968
filter acl 1804 ace 150 enable
filter acl 1804 ace 160 create name "AV-Yama_YONETIM_UDP_2967"
filter acl 1804 ace 160 action permit stop-on-match true
filter acl 1804 acl 160 ip ip-protocol-type eq udp
filter acl 1804 ace 160 protocol udp-dst-port eq 2967
filter acl 1804 ace 160 enable
filter acl 1804 ace 180 create name "SUNUCU_YONETIM"
filter acl 1804 ace 180 action permit stop-on-match true
filter acl 1804 ace 180 ip src-ip eq 100.20.150.80-100.20.150.95
filter acl 1804 ace 180 ip ip-protocol-type eq tcp
filter acl 1804 ace 180 protocol tcp-dst-port eq 3389
filter acl 1804 ace 180 enable
filter acl 1804 ace 200 create name "OTOMIZE_DEBIT_CARD_OPS"
filter acl 1804 ace 200 action permit stop-on-match true
filter acl 1804 ace 200 ip src-ip eq 100.20.114.0-100.20.114.255
filter acl 1804 ace 200 ip ip-protocol-type eq tcp
filter acl 1804 ace 200 protocol tcp-dst-port eq 445
filter acl 1804 ace 200 enable
filter acl 1804 ace 210 create name "OTOMIZE_DEBIT_CARD_OPS"
filter acl 1804 ace 210 action permit stop-on-match true
filter acl 1804 ace 210 ip src-ip eq 100.20.24.0-100.20.24.255
filter acl 1804 ace 210 ip ip-protocol-type eq tcp
filter acl 1804 ace 210 protocol tcp-dst-port eq 445
filter acl 1804 ace 210 enable
filter acl 1804 ace 220 create name "LOGLAMA"
filter acl 1804 ace 220 action permit
filter acl 1804 ace 220 debug count enable
filter acl 1804 ace 220 ip src-ip ge 0.0.0.0
filter acl 1804 ace 220 enable
```

```
filter acl 1804 ace 230 create name "DENY_ANY"
filter acl 1804 ace 230 action deny stop-on-match true
filter acl 1804 ace 230 debug count enable
filter acl 1804 ace 230 ip src-ip ge 0.0.0.0
filter acl 1804 ace 230 ip dst-ip ge 0.0.0.0
filter acl 1804 ace 230 enable
```

The following section provides details about the filter configuration for the second core Layer 3 host

```
#
# R-MODULE FILTER CONFIGURATION
#
filter act 1 create name "BUSINESS 1"
filter act 1 ip srcIp,dstIp,ipOptions,ipProtoType
filter act 1 protocol
tcpSrcPort,udpSrcPort,tcpDstPort,udpDstPort,tcpFlags,icmpMsgType
filter act 1 apply filter acl 1 create outPort act 1 name
"VRRP_Drop_ACL"
filter acl 1 port add 4/46
filter acl 1 ace 1 create name "Vrrp" filter acl 1 ace 1 action deny
stop-on-match true
filter acl 1 ace 1 debug count enable
filter acl 1 ace 1 ip ip-protocol-type eq vrrp
filter acl 1 ace 1 enable

filter acl 171 create inVlan act 1 name "TOPLANTI_VE_EGITIM_ACL"
filter acl 171 vlan add 171
filter acl 171 disable
filter acl 171 ace 10 create name "ICMP_PERMIT"
filter acl 171 ace 10 action permit stop-on-match true
filter acl 171 ace 10 ip ip-protocol-type eq icmp
filter acl 171 ace 10 enable filter acl 171 ace 20 create name
"IGMP_PERMIT"
filter acl 171 ace 20 action permit stop-on-match true
```

```
filter acl 171 ace 20 ip ip-protocol-type eq 2
filter acl 171 ace 20 enable
filter acl 171 ace 30 create name "VRRP_PERMIT"
filter acl 171 ace 30 action permit stop-on-match true
filter acl 171 ace 30 ip ip-protocol-type eq vrrp
filter acl 171 ace 30 enable
filter acl 171 ace 40 create name "DNS_PERMIT"
filter acl 171 ace 40 action permit stop-on-match true
filter acl 171 ace 40 ip src-ip eq 100.20.171.0-100.20.171.255
filter acl 171 ace 40 ip dst-ip eq 100.20.104.0-100.20.104.255
filter acl 171 ace 40 protocol udp-dst-port eq dns
filter acl 171 ace 40 enable
filter acl 171 ace 50 create name "ESTABLISHED"
filter acl 171 ace 50 action permit stop-on-match true
filter acl 171 ace 50 ip src-ip eq 100.6.172.0-100.6.172.255
filter acl 171 ace 50 ip ip-protocol-type eq tcp
filter acl 171 ace 50 protocol tcp-dst-port ge 1023
filter acl 171 ace 50 protocol tcp-flags match-any rst,ack
filter acl 171 ace 50 enable
filter acl 171 ace 60 create name "DHCP_PERMIT"
filter acl 171 ace 60 action permit stop-on-match true
filter acl 171 ace 60 protocol udp-dst-port eq bootpServer
filter acl 171 ace 60 enable
filter acl 171 ace 80 create name "DC_DNS_EXC_PERMIT"
filter acl 171 ace 80 action permit stop-on-match true
filter acl 171 ace 80 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 171 ace 80 ip dst-ip eq 100.20.104.0-100.20.105.255
filter acl 171 ace 80 enable filter acl 171 ace 90 create name
"HTTP_PERMIT"
filter acl 171 ace 90 action permit stop-on-match true
filter acl 171 ace 90 ip src-ip eq 100.20.172.0-100.20.172.255
```

```
filter acl 171 ace 90 protocol tcp-dst-port eq 80
filter acl 171 ace 90 enable
filter acl 171 ace 100 create name "HTTPS_PERMIT"
filter acl 171 ace 100 action permit stop-on-match true
filter acl 171 ace 100 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 171 ace 100 protocol tcp-dst-port eq 443
filter acl 171 ace 100 enable
filter acl 171 ace 110 create name "PROXY_8080_PERMIT"
filter acl 171 ace 110 action permit stop-on-match true
filter acl 171 ace 110 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 171 ace 110 ip dst-ip eq 100.20.189.0-100.20.189.255
filter acl 171 ace 110 protocol tcp-dst-port eq 8080
filter acl 171 ace 110 enable
filter acl 171 ace 120 create name "CITRIX_Conn"
filter acl 171 ace 120 action permit stop-on-match true
filter acl 171 ace 120 protocol tcp-dst-port eq 1494
filter acl 171 ace 120 protocol udp-dst-port eq 1604
filter acl 171 ace 120 enable
filter acl 171 ace 130 create name "PWC_VPN_ERISIM"
filter acl 171 ace 130 action permit stop-on-match true
filter acl 171 ace 130 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 171 ace 130 protocol tcp-dst-port eq 11160
filter acl 171 ace 130 enable
filter acl 171 ace 140 create name "Microsoft_FileSharing_PERMIT"
filter acl 171 ace 140 action permit stop-on-match true
filter acl 171 ace 140 debug count enable
filter acl 171 ace 140 protocol tcp-dst-port eq 135-139
filter acl 171 ace 140 protocol udp-dst-port eq 135-139
filter acl 171 ace 140 enable
filter acl 171 ace 150 create name "Microsoft_FileSharing_PERMIT"
filter acl 171 ace 150 action permit stop-on-match true
```

```
filter acl 171 ace 150 debug count enable
filter acl 171 ace 150 protocol tcp-dst-port eq 445
filter acl 171 ace 150 protocol udp-dst-port eq 445
filter acl 171 ace 150 enable

filter acl 172 create inVlan act 1 name "MISAFIR_ACL"
filter acl 172 vlan add 172
filter acl 172 disable
filter acl 172 ace 5 create name "Misafir_to_Misafir"
filter acl 172 ace 5 action permit stop-on-match true
filter acl 172 ace 5 ip dst-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 5 enable
filter acl 172 ace 10 create name "ICMP_PERMIT"
filter acl 172 ace 10 action permit stop-on-match true
filter acl 172 ace 10 ip ip-protocol-type eq icmp
filter acl 172 ace 10 enable
filter acl 172 ace 20 create name "IGMP_PERMIT"
filter acl 172 ace 20 action permit stop-on-match true
filter acl 172 ace 20 ip ip-protocol-type eq 2
filter acl 172 ace 20 enable
filter acl 172 ace 30 create name "VRRP_PERMIT"
filter acl 172 ace 30 action permit stop-on-match true
filter acl 172 ace 30 ip ip-protocol-type eq vrrp
filter acl 172 ace 30 enable
filter acl 172 ace 40 create name "DNS_PERMIT"
filter acl 172 ace 40 action permit stop-on-match true
filter acl 172 ace 40 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 40 ip dst-ip eq 100.20.104.0-100.20.104.255
filter acl 172 ace 40 protocol udp-dst-port eq dns
filter acl 172 ace 40 enable
filter acl 172 ace 50 create name "ESTABLISHED"
```

```
filter acl 172 ace 50 action permit stop-on-match true
filter acl 172 ace 50 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 50 ip ip-protocol-type eq tcp
filter acl 172 ace 50 protocol tcp-dst-port ge 1023
filter acl 172 ace 50 protocol tcp-flags match-any rst,ack
filter acl 172 ace 50 enable
filter acl 172 ace 60 create name "DHCP_PERMIT"
filter acl 172 ace 60 action permit stop-on-match true
filter acl 172 ace 60 protocol udp-dst-port eq bootpServer
filter acl 172 ace 60 enable
filter acl 172 ace 80 create name "DC_DNS_EXC_PERMIT"
filter acl 172 ace 80 action permit stop-on-match true
filter acl 172 ace 80 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 80 ip dst-ip eq 100.20.104.0-100.20.105.255
filter acl 172 ace 80 enable
filter acl 172 ace 90 create name "HTTP_PERMIT"
filter acl 172 ace 90 action permit stop-on-match true
filter acl 172 ace 90 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 90 ip ip-protocol-type eq tcp
filter acl 172 ace 90 protocol tcp-dst-port eq 80
filter acl 172 ace 100 create name "HTTPS_PERMIT"
filter acl 172 ace 100 action permit stop-on-match true
filter acl 172 ace 100 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 100 ip ip-protocol-type eq tcp
filter acl 172 ace 100 protocol tcp-dst-port eq 443
filter acl 172 ace 100 enable
filter acl 172 ace 105 create name "REMDESKTOP_PERMIT"
filter acl 172 ace 105 action permit stop-on-match true
filter acl 172 ace 105 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 105 ip ip-protocol-type eq tcp
filter acl 172 ace 105 protocol tcp-dst-port eq 3389
```

```
filter acl 172 ace 105 enable
filter acl 172 ace 106 create name "NORKOM_PERMIT"
filter acl 172 ace 106 action permit stop-on-match true
filter acl 172 ace 106 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 106 ip dst-ip eq
100.6.106.0-100.6.106.255,100.20.24.0-100.20.24.255
filter acl 172 ace 106 enable
filter acl 172 ace 107 create name "SPECTRUM_PERMIT"
filter acl 172 ace 107 action permit stop-on-match true
filter acl 172 ace 107 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 107 ip dst-ip eq 100.20.17.0-100.20.17.255
filter acl 172 ace 107 enable
filter acl 172 ace 110 create name "PROXY_8080_PERMIT"
filter acl 172 ace 110 action permit stop-on-match true
filter acl 172 ace 110 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 110 ip dst-ip eq 100.20.189.0-100.20.189.255
filter acl 172 ace 110 ip ip-protocol-type eq tcp
filter acl 172 ace 110 protocol tcp-dst-port eq 8080
filter acl 172 ace 110 enable
filter acl 172 ace 120 create name "CITRIX_Conn-tcp"
filter acl 172 ace 120 action permit stop-on-match true
filter acl 172 ace 120 ip ip-protocol-type eq tcp
filter acl 172 ace 120 protocol tcp-dst-port eq 1494
filter acl 172 ace 120 enable
filter acl 172 ace 121 create name "CITRIX_Conn-udp"
filter acl 172 ace 121 action permit stop-on-match true
filter acl 172 ace 121 ip ip-protocol-type eq udp
filter acl 172 ace 121 protocol udp-dst-port eq 1604
filter acl 172 ace 121 enable
filter acl 172 ace 128 create name "VOIP_VLAN_PERMIT"
filter acl 172 ace 128 action permit stop-on-match true
```

```
filter acl 172 ace 128 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 128 ip dst-ip eq 10.201.0.0-10.201.31.255
filter acl 172 ace 128 enable
filter acl 172 ace 129 create name "GANYMEDE_PERMIT"
filter acl 172 ace 129 action permit stop-on-match true
filter acl 172 ace 129 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 129 ip dst-ip eq 100.6.100.225
filter acl 172 ace 129 enable
filter acl 172 ace 130 create name "PWC_VPN_ERISIM"
filter acl 172 ace 130 action permit stop-on-match true
filter acl 172 ace 130 ip src-ip eq 100.20.172.0-100.20.172.255
filter acl 172 ace 130 ip ip-protocol-type eq tcp
filter acl 172 ace 130 protocol tcp-dst-port eq 11160
filter acl 172 ace 130 enable
filter acl 172 ace 131 create name "ISAKMP"
filter acl 172 ace 131 action permit stop-on-match true
filter acl 172 ace 131 ip ip-protocol-type eq udp
filter acl 172 ace 131 protocol udp-dst-port eq 500
filter acl 172 ace 131 enable
filter acl 172 ace 132 create name "ESP"
filter acl 172 ace 132 action permit stop-on-match true
filter acl 172 ace 132 ip ip-protocol-type eq 50
filter acl 172 ace 132 enable
filter acl 172 ace 133 create name "LOGLAMAK_ICIN"
filter acl 172 ace 133 action permit redirect-next-hop 100.20.150.34
stop-on-match true ipfix enable
filter acl 172 ace 133 debug count enable
filter acl 172 ace 133 ip src-ip eq 100.20.172.72
filter acl 172 ace 140 create name "DENY_ANY_ANY"
filter acl 172 ace 140 action deny stop-on-match true
filter acl 172 ace 140 debug count enable
```

```
filter acl 172 ace 140 ip src-ip ge 0.0.0.0
filter acl 172 ace 140 ip dst-ip ge 0.0.0.0
filter acl 172 ace 140 enable

filter acl 802 create inVlan act 1 name "NICE-CLS_ACL-in"
filter acl 802 vlan add 802
filter acl 802 disable
filter acl 802 ace 1 create name "NICE_to_NICE"
filter acl 802 ace 1 action permit stop-on-match true
filter acl 802 ace 1 ip dst-ip eq 100.20.174.32-100.20.174.63
filter acl 802 ace 1 enable
filter acl 802 ace 10 create name "ICMP_PERMIT"
filter acl 802 ace 10 action permit stop-on-match true
filter acl 802 ace 10 ip ip-protocol-type eq icmp
filter acl 802 ace 10 enable filter acl 802 ace 20 create name
"IGMP_PERMIT"
filter acl 802 ace 20 action permit stop-on-match true
filter acl 802 ace 20 ip ip-protocol-type eq 2
filter acl 802 ace 20 enable
filter acl 802 ace 30 create name "VRRP_PERMIT"
filter acl 802 ace 30 action permit stop-on-match true
filter acl 802 ace 30 ip ip-protocol-type eq vrrp
filter acl 802 ace 30 enable
filter acl 802 ace 40 create name "DNS_PERMIT"
filter acl 802 ace 40 action permit stop-on-match true
filter acl 802 ace 40 ip src-ip eq 100.20.174.32-100.20.174.63
filter acl 802 ace 40 ip dst-ip eq 100.20.104.0-100.20.104.255
filter acl 802 ace 40 protocol udp-dst-port eq dns
filter acl 802 ace 40 enable
filter acl 802 ace 45 create name "DC-EXCH-DNS"
filter acl 802 ace 45 action permit stop-on-match true
filter acl 802 ace 45 ip dst-ip eq 100.20.104.0-100.20.105.255
```

```
filter acl 802 ace 45 enable
filter acl 802 ace 50 create name "ESTABLISHED"
filter acl 802 ace 50 action permit stop-on-match true
filter acl 802 ace 50 ip src-ip eq 100.20.174.32-100.20.174.63
filter acl 802 ace 50 ip ip-protocol-type eq tcp
filter acl 802 ace 50 protocol tcp-dst-port ge 1023
filter acl 802 ace 50 protocol tcp-flags match-any rst,ack
filter acl 802 ace 50 enable
filter acl 802 ace 51 create name "UDP_Permit"
filter acl 802 ace 51 action permit stop-on-match true
filter acl 802 ace 51 ip ip-protocol-type eq udp
filter acl 802 ace 51 enable
filter acl 802 ace 60 create name "NICE_Logging"
filter acl 802 ace 60 action permit stop-on-match true
filter acl 802 ace 60 ip src-ip eq 100.20.174.32-100.20.174.63
filter acl 802 ace 60 ip ip-protocol-type eq tcp
filter acl 802 ace 60 protocol tcp-dst-port eq 2011
filter acl 802 ace 60 enable
filter acl 802 ace 65 create name "RTS_Conn"
filter acl 802 ace 65 action permit stop-on-match true
filter acl 802 ace 65 ip dst-ip eq 100.20.152.20
filter acl 802 ace 65 enable filter acl 802 ace 70 create name
"CTI_Conn"
filter acl 802 ace 70 action permit stop-on-match true
filter acl 802 ace 70 ip src-ip eq 100.20.174.32-100.20.174.63
filter acl 802 ace 70 ip ip-protocol-type eq tcp
filter acl 802 ace 70 protocol tcp-dst-port eq 3750
filter acl 802 ace 70 enable filter acl 802 ace 90 create name
"LOGLAMA"
filter acl 802 ace 90 action permit redirect-next-hop 100.20.150.217
stop-on-match true
filter acl 802 ace 90 debug count enable
```

```
filter acl 802 ace 90 ip src-ip ge 0.0.0.0
filter acl 802 ace 100 create name "DENY_ANY"
filter acl 802 ace 100 action deny stop-on-match true
filter acl 802 ace 100 debug count enable
filter acl 802 ace 100 ip src-ip ge 0.0.0.0
filter acl 802 ace 100 ip dst-ip ge 0.0.0.0
filter acl 802 ace 100 enable

filter acl 804 create inVlan act 1 name "BASIM_LIMITED-in"
filter acl 804 vlan add 804
filter acl 804 ace 5 create name "Basim_to_Basim"
filter acl 804 ace 5 action permit stop-on-match true
filter acl 804 ace 5 ip dst-ip eq 100.20.174.96-100.20.174.127
filter acl 804 ace 5 enable
filter acl 804 ace 10 create name "ICMP_PERMIT"
filter acl 804 ace 10 action permit stop-on-match true
filter acl 804 ace 10 ip ip-protocol-type eq icmp
filter acl 804 ace 10 enable
filter acl 804 ace 20 create name "IGMP_PERMIT"
filter acl 804 ace 20 action permit stop-on-match true
filter acl 804 ace 20 ip ip-protocol-type eq 2
filter acl 804 ace 20 enable
filter acl 804 ace 30 create name "VRRP_PERMIT"
filter acl 804 ace 30 action permit stop-on-match true
filter acl 804 ace 30 ip ip-protocol-type eq vrrp
filter acl 804 ace 30 enable
filter acl 804 ace 40 create name "DNS_PERMIT"
filter acl 804 ace 40 action permit stop-on-match true
filter acl 804 ace 40 protocol udp-dst-port eq dns
filter acl 804 ace 40 enable
filter acl 804 ace 45 create name "DC-EXCH-DNS"
```

```
filter acl 804 ace 45 action permit stop-on-match true
filter acl 804 ace 45 ip dst-ip eq 100.20.104.0-100.20.105.255
filter acl 804 ace 45 enable
filter acl 804 ace 50 create name "ESTABLISHED"
filter acl 804 ace 50 action permit stop-on-match true
filter acl 804 ace 50 ip src-ip eq 100.20.174.97-100.20.174.127
filter acl 804 ace 50 ip ip-protocol-type eq tcp
filter acl 804 ace 50 protocol tcp-dst-port ge 1023
filter acl 804 ace 50 protocol tcp-flags match-any rst,ack
filter acl 804 ace 50 enable
filter acl 804 ace 60 create name "E-BANK_ERISIM"
filter acl 804 ace 60 action permit stop-on-match true
filter acl 804 ace 60 ip dst-ip eq 100.20.115.11
filter acl 804 ace 60 ip ip-protocol-type eq tcp
filter acl 804 ace 60 protocol tcp-dst-port eq 80
filter acl 804 ace 60 enable
filter acl 804 ace 70 create name "E-BANK_ERISIM_HTTPS"
filter acl 804 ace 70 action permit stop-on-match true
filter acl 804 ace 70 ip dst-ip eq 100.20.115.11
filter acl 804 ace 70 ip ip-protocol-type eq tcp
filter acl 804 ace 70 protocol tcp-dst-port eq 443
filter acl 804 ace 70 enable
filter acl 804 ace 80 create name "FRED_Erisim"
filter acl 804 ace 80 action permit stop-on-match true
filter acl 804 ace 80 ip dst-ip eq 100.20.100.145
filter acl 804 ace 80 enable
filter acl 804 ace 81 create name "BARNEY_Erisim"
filter acl 804 ace 81 action permit stop-on-match true
filter acl 804 ace 81 ip dst-ip eq 100.20.100.151
filter acl 804 ace 81 enable
filter acl 804 ace 90 create name "BUFFY_ERISIM"
```

```
filter acl 804 ace 90 action permit stop-on-match true
filter acl 804 ace 90 ip dst-ip eq 100.20.100.77
filter acl 804 ace 90 ip ip-protocol-type eq tcp
filter acl 804 ace 90 protocol tcp-dst-port eq 1433
filter acl 804 ace 90 enable
filter acl 804 ace 100 create name "ROMTest_ERISIM"
filter acl 804 ace 100 action permit stop-on-match true
filter acl 804 ace 100 ip dst-ip eq 100.20.24.77
filter acl 804 ace 100 ip ip-protocol-type eq tcp
filter acl 804 ace 100 protocol tcp-dst-port eq 1433
filter acl 804 ace 100 enable
filter acl 804 ace 101 create name "Mrksql-t0_ERISIM"
filter acl 804 ace 101 action permit stop-on-match true
filter acl 804 ace 101 ip dst-ip eq 100.20.20.77
filter acl 804 ace 101 ip ip-protocol-type eq tcp
filter acl 804 ace 101 protocol tcp-dst-port eq 1433
filter acl 804 ace 101 enable filter acl 804 ace 110 create name
"ROSETTA_ERISIM"
filter acl 804 ace 110 action permit stop-on-match true
filter acl 804 ace 110 ip dst-ip eq 172.17.1.100
filter acl 804 ace 110 enable
filter acl 804 ace 120 create name "PLAST_ERISIM"
filter acl 804 ace 120 action permit stop-on-match true
filter acl 804 ace 120 ip dst-ip eq 212.57.7.20
filter acl 804 ace 120 enable
filter acl 804 ace 130 create name "AV-Yama_YONETIM_2967"
filter acl 804 ace 130 action permit stop-on-match true
filter acl 804 ace 130 ip ip-protocol-type eq tcp
filter acl 804 ace 130 protocol tcp-dst-port eq 2967
filter acl 804 ace 130 enable
filter acl 804 ace 140 create name "AV-Yama_YONETIM_9968"
```

```
filter acl 804 ace 140 action permit stop-on-match true
filter acl 804 ace 140 ip ip-protocol-type eq tcp
filter acl 804 ace 140 protocol tcp-dst-port eq 9968
filter acl 804 ace 140 enable
filter acl 804 ace 150 create name "AV-Yama_YONETIM_UDP_2967"
filter acl 804 ace 150 action permit stop-on-match true
filter acl 804 ace 150 ip ip-protocol-type eq udp
filter acl 804 ace 150 protocol udp-dst-port eq 2967
filter acl 804 ace 150 enable
filter acl 804 ace 160 create name "AV-Yama_YONETIM_UDP_9968"
filter acl 804 ace 160 action permit stop-on-match true
filter acl 804 ace 160 ip ip-protocol-type eq udp
filter acl 804 ace 160 protocol udp-dst-port eq 9968
filter acl 804 ace 160 enable
filter acl 804 ace 170 create name "AV-Yama_YONETIM_UDP_Source"
filter acl 804 ace 170 action permit stop-on-match true
filter acl 804 ace 170 ip ip-protocol-type eq udp
filter acl 804 ace 170 protocol udp-src-port eq 9968
filter acl 804 ace 170 enable
filter acl 804 ace 210 create name "PROXY_ERISIM_EK"
filter acl 804 ace 210 action permit stop-on-match true
filter acl 804 ace 210 ip dst-ip eq 100.20.189.0-100.20.189.255
filter acl 804 ace 210 ip ip-protocol-type eq tcp
filter acl 804 ace 210 protocol tcp-dst-port eq 8080
filter acl 804 ace 210 enable
filter acl 804 ace 220 create name "LOGLAMA"
filter acl 804 ace 220 action permit redirect-next-hop 100.20.150.217
stop-on-match true
filter acl 804 ace 220 debug count enable
filter acl 804 ace 220 ip src-ip ge 0.0.0.0
filter acl 804 ace 230 create name "DENY_ANY"
```

```
filter acl 804 ace 230 action deny stop-on-match true
filter acl 804 ace 230 debug count enable
filter acl 804 ace 230 ip src-ip ge 0.0.0.0
filter acl 804 ace 230 ip dst-ip ge 0.0.0.0
filter acl 804 ace 230 enable

filter acl 805 create inVlan act 1 name "SBS_Remote"
filter acl 805 vlan add 805
filter acl 805 ace 5 create name "SBS-to-SBS"
filter acl 805 ace 5 action permit stop-on-match true
filter acl 805 ace 5 ip dst-ip eq 100.20.174.128-100.20.174.135
filter acl 805 ace 5 enable
filter acl 805 ace 10 create name "ICMP_PERMIT"
filter acl 805 ace 10 action permit stop-on-match true
filter acl 805 ace 10 ip ip-protocol-type eq icmp
filter acl 805 ace 10 enable
filter acl 805 ace 20 create name "IGMP_PERMIT"
filter acl 805 ace 20 action permit stop-on-match true
filter acl 805 ace 20 ip ip-protocol-type eq 2
filter acl 805 ace 20 enable
filter acl 805 ace 30 create name "VRRP_PERMIT"
filter acl 805 ace 30 action permit stop-on-match true
filter acl 805 ace 30 ip ip-protocol-type eq vrrp
filter acl 805 ace 30 enable
filter acl 805 ace 40 create name "DNS_PERMIT"
filter acl 805 ace 40 action permit stop-on-match true
filter acl 805 ace 40 protocol udp-dst-port eq 53
filter acl 805 ace 40 enable filter acl 805 ace 50 create name
"ESTABLISHED"
filter acl 805 ace 50 action permit stop-on-match true
filter acl 805 ace 50 ip src-ip eq 100.20.174.128-100.20.174.134
filter acl 805 ace 50 ip ip-protocol-type eq tcp
```

```
filter acl 805 ace 50 protocol tcp-dst-port ge 1023
filter acl 805 ace 50 protocol tcp-flags match-any rst,ack
filter acl 805 ace 50 enable
filter acl 805 ace 80 create name "DC_DNS_EXCH_PERMIT"
filter acl 805 ace 80 action permit stop-on-match true
filter acl 805 ace 80 ip dst-ip eq 100.20.104.0-100.20.105.255
filter acl 805 ace 80 enable
filter acl 805 ace 90 create name "HTTP_PERMIT"
filter acl 805 ace 90 action permit stop-on-match true
filter acl 805 ace 90 ip ip-protocol-type eq tcp
filter acl 805 ace 90 protocol tcp-dst-port eq 80
filter acl 805 ace 90 enable
filter acl 805 ace 100 create name "HTTPS_PERMIT"
filter acl 805 ace 100 action permit stop-on-match true
filter acl 805 ace 100 ip ip-protocol-type eq tcp
filter acl 805 ace 100 protocol tcp-dst-port eq 443
filter acl 805 ace 100 enable
filter acl 805 ace 105 create name "REMDESKTOP_PERMIT"
filter acl 805 ace 105 action permit stop-on-match true
filter acl 805 ace 105 ip ip-protocol-type eq tcp
filter acl 805 ace 105 protocol tcp-dst-port eq 3389
filter acl 805 ace 105 enable
filter acl 805 ace 110 create name "PROXY_8080_PERMIT"
filter acl 805 ace 110 action permit stop-on-match true
filter acl 805 ace 110 ip dst-ip eq 100.20.189.0-100.20.189.255
filter acl 805 ace 110 ip ip-protocol-type eq tcp
filter acl 805 ace 110 protocol tcp-dst-port eq 8080
filter acl 805 ace 110 enable
filter acl 805 ace 120 create name "DAMEWARE_PERMIT"
filter acl 805 ace 120 action permit
filter acl 805 ace 120 ip src-ip eq 100.20.174.128-100.20.174.134
```

```
filter acl 805 ace 120 protocol tcp-dst-port eq 445,6129
filter acl 805 ace 120 enable
filter acl 805 ace 140 create name "DENY_ANY_ANY"
filter acl 805 ace 140 action deny stop-on-match true
filter acl 805 ace 140 ip src-ip ge 0.0.0.0
filter acl 805 ace 140 ip dst-ip ge 0.0.0.0
filter acl 805 ace 140 enable

filter acl 1802 create outVlan act 1 name "NICE-CLS_ACL-out"
filter acl 1802 vlan add 802
filter acl 1802 disable filter acl 1802 ace 10 create name
"ICMP_PERMIT"
filter acl 1802 ace 10 action permit stop-on-match true
filter acl 1802 ace 10 ip ip-protocol-type eq icmp
filter acl 1802 ace 10 enable
filter acl 1802 ace 20 create name "IGMP_PERMIT"
filter acl 1802 ace 20 action permit stop-on-match true
filter acl 1802 ace 20 ip ip-protocol-type eq 2
filter acl 1802 ace 20 enable filter acl 1802 ace 30 create name
"VRRP_PERMIT"
filter acl 1802 ace 30 action permit stop-on-match true
filter acl 1802 ace 30 ip ip-protocol-type eq vrrp
filter acl 1802 ace 30 enable
filter acl 1802 ace 51 create name "UDP_Permit"
filter acl 1802 ace 51 action permit stop-on-match true
filter acl 1802 ace 51 ip ip-protocol-type eq udp
filter acl 1802 ace 51 enable
filter acl 1802 ace 60 create name "NICE_Logging"
filter acl 1802 ace 60 action permit stop-on-match true
filter acl 1802 ace 60 ip src-ip eq 100.20.174.32-100.20.174.63
filter acl 1802 ace 60 protocol tcp-dst-port eq 2011
filter acl 1802 ace 60 enable
```

```
filter acl 1802 ace 100 create name "DENY_ANY"
filter acl 1802 ace 100 action deny stop-on-match true
filter acl 1802 ace 100 ip src-ip ge 0.0.0.0
filter acl 1802 ace 100 ip dst-ip ge 0.0.0.0
filter acl 1802 ace 100 enable
filter acl 1804 create outVlan act 1 name "BASIM_LIMITED-out"
filter acl 1804 vlan add 804
filter acl 1804 ace 5 create name "BASIM-to-BASIM"
filter acl 1804 ace 5 action permit stop-on-match true
filter acl 1804 ace 5 ip src-ip eq 100.20.174.96-100.20.174.127
filter acl 1804 ace 5 ip dst-ip eq 100.20.174.96-100.20.174.127
filter acl 1804 ace 5 enable
filter acl 1804 ace 10 create name "ICMP_PERMIT"
filter acl 1804 ace 10 action permit stop-on-match true
filter acl 1804 ace 10 ip ip-protocol-type eq icmp
filter acl 1804 ace 10 enable
filter acl 1804 ace 20 create name "IGMP_PERMIT"
filter acl 1804 ace 20 action permit stop-on-match true
filter acl 1804 ace 20 ip ip-protocol-type eq 2
filter acl 1804 ace 20 enable
filter acl 1804 ace 30 create name "VRRP_PERMIT"
filter acl 1804 ace 30 action permit stop-on-match true
filter acl 1804 ace 30 ip ip-protocol-type eq vrrp
filter acl 1804 ace 30 enable
filter acl 1804 ace 40 create name "DNS_PERMIT"
filter acl 1804 ace 40 action permit stop-on-match true
filter acl 1804 ace 40 protocol udp-src-port eq 53
filter acl 1804 ace 40 enable
filter acl 1804 ace 45 create name "DC-EXCH-DNS"
filter acl 1804 ace 45 action permit stop-on-match true
filter acl 1804 ace 45 ip src-ip eq 100.20.104.0-100.20.105.255
```

```
filter acl 1804 ace 45 enable filter acl 1804 ace 50 create name
"ESTABLISHED"
filter acl 1804 ace 50 action permit stop-on-match true
filter acl 1804 ace 50 ip dst-ip eq 100.20.174.97-100.20.174.127
filter acl 1804 ace 50 ip ip-protocol-type eq tcp
filter acl 1804 ace 50 protocol tcp-dst-port ge 1023
filter acl 1804 ace 50 protocol tcp-flags match-any rst,ack
filter acl 1804 ace 50 enable
filter acl 1804 ace 80 create name "PWC_ERISIM"
filter acl 1804 ace 80 action permit stop-on-match true
filter acl 1804 ace 80 ip src-ip eq 100.20.100.145
filter acl 1804 ace 80 enable
filter acl 1804 ace 110 create name "ROSETTA_ERISIM"
filter acl 1804 ace 110 action permit stop-on-match true
filter acl 1804 ace 110 ip src-ip eq 172.17.1.100
filter acl 1804 ace 110 enable
filter acl 1804 ace 120 create name "PLAST_ERISIM"
filter acl 1804 ace 120 action permit stop-on-match true
filter acl 1804 ace 120 ip src-ip eq 212.57.7.20
filter acl 1804 ace 120 enable
filter acl 1804 ace 130 create name "AV-Yama_YONETIM_9968"
filter acl 1804 ace 130 action permit stop-on-match true
filter acl 1804 ace 130 ip ip-protocol-type eq tcp
filter acl 1804 ace 130 protocol tcp-dst-port eq 9968
filter acl 1804 ace 130 enable
filter acl 1804 ace 140 create name "AV-Yama_YONETIM_2967"
filter acl 1804 ace 140 action permit stop-on-match true
filter acl 1804 ace 140 ip ip-protocol-type eq tcp
filter acl 1804 ace 140 protocol tcp-dst-port eq 2967
filter acl 1804 ace 140 enable
filter acl 1804 ace 150 create name "AV-Yama_YONETIM_UDP_9968"
```

```
filter acl 1804 ace 150 action permit stop-on-match true
filter acl 1804 ace 150 ip ip-protocol-type eq udp
filter acl 1804 ace 150 protocol udp-dst-port eq 9968
filter acl 1804 ace 150 enable filter acl 1804 ace 160 create name
"AV-Yama_YONETIM_UDP_2967"
filter acl 1804 ace 160 action permit stop-on-match true
filter acl 1804 ace 160 ip ip-protocol-type eq udp
filter acl 1804 ace 160 protocol udp-dst-port eq 2967
filter acl 1804 ace 160 enable filter acl 1804 ace 180 create name
"SUNUCU_YONETIM"
filter acl 1804 ace 180 action permit stop-on-match true
filter acl 1804 ace 180 ip src-ip eq 100.20.150.80-100.20.150.95
filter acl 1804 ace 180 ip ip-protocol-type eq tcp
filter acl 1804 ace 180 protocol tcp-dst-port eq 3389
filter acl 1804 ace 180 enable
filter acl 1804 ace 200 create name "OTOMIZE_DEBIT_CARD_OPS"
filter acl 1804 ace 200 action permit stop-on-match true
filter acl 1804 ace 200 ip src-ip eq 100.20.114.0-100.20.114.255
filter acl 1804 ace 200 ip ip-protocol-type eq tcp
filter acl 1804 ace 200 protocol tcp-dst-port eq 445
filter acl 1804 ace 200 enable
filter acl 1804 ace 210 create name "OTOMIZE_DEBIT_CARD_OPS"
filter acl 1804 ace 210 action permit stop-on-match true
filter acl 1804 ace 210 ip src-ip eq 100.20.24.0-100.20.24.255
filter acl 1804 ace 210 ip ip-protocol-type eq tcp
filter acl 1804 ace 210 protocol tcp-dst-port eq 445
filter acl 1804 ace 210 enable
filter acl 1804 ace 230 create name "DENY_ANY"
filter acl 1804 ace 230 action deny stop-on-match true
filter acl 1804 ace 230 debug count enable
filter acl 1804 ace 230 ip src-ip ge 0.0.0.0
filter acl 1804 ace 230 ip dst-ip ge 0.0.0.0
```

```
filter acl 1804 ace 230 enable
```

Appendix B: Egress queues and pages

The following tables describes the relationship between pages and packets for the Avaya Ethernet Routing Switch 8800/8600 egress queues. In these tables, BP denotes backplane. The first table shows information for data for packets that do not use a PHE. The second table describes pages using packets that use a PHE (that is, packets from R, RS, or 8800 modules).

Table 34: Cell breaks, back breaks, and back page usage without PHE

Start	End	Cells	BP packet bytes		BP usage		BP count	Last page bytes		Break count
1	72	1			0					
73	148	2			0					
149	224	3	1	76	5	80	1	5	80	148
225	300	4	77	152	85	160	1	85	160	0
301	376	5	153	228	165	240	1	165	240	0
377	452	6	229	304	245	360	1	245	360	0
453	528	7	305	380	325	400	1	325	400	0
529	604	8	381	456	405	480	1	405	480	0
605	680	9	457	532	485	560	2	-27	48	632
681	756	10	533	608	565	640	2	53	128	0
757	832	11	609	684	645	720	2	133	208	0
833	908	12	685	760	725	800	2	213	288	0
909	984	13	761	836	805	880	2	293	368	0
985	1060	14	837	912	885	960	2	373	448	0
1061	1136	15	913	988	965	1040	3	-59	16	1120
...	...	...	...	...	...	...	...	...	...	...
11777	11852	156	11629	11704	12245	12320	25	-43	32	11820

Table 35: Cell breaks, back breaks, and back page usage with PHE

Start	End	Cells	BP packet bytes		BP usage		BP count	Last page bytes		Break count
1	68	1			0					

Start	End	Cells	BP packet bytes		BP usage		BP count	Last page bytes		Break count
69	144	2			0					
145	220	3	1	76	5	80	1	5	80	144
221	296	4	77	152	85	160	1	85	160	0
297	372	5	153	228	165	240	1	165	240	0
373	448	6	229	304	245	320	1	245	320	0
449	524	7	305	380	325	400	1	325	400	0
525	600	8	381	456	405	480	1	405	480	0
601	676	9	457	532	485	560	2	-27	48	628
677	752	10	533	608	565	640	2	53	128	0
753	828	11	609	684	645	720	2	133	208	0
829	904	12	685	760	725	800	2	213	288	0
905	980	13	761	836	805	880	2	293	368	0
981	1056	14	837	912	885	960	2	373	448	0
1057	1132	15	913	988	965	1040	3	-59	16	1116
...	...	...	...	...	...	...	...	...	...	...
11773	11848	156	11629	11704	12245	12320	25	-43	32	11816

Comments? infodev@avaya.com

Appendix C: Workaround for inVlan, srcIp ACL

When you create an ACL with the type inVlanthat uses an ACT based on the source IP address, the ACL no longerworks after the ARP aging time elapses. This does not cause a securitybreach.

To ensure the ACL operates correctly, you can add an additional ACL ACE that permits all ARP requests.

The following procedure shows how to create an ACE to solve this issue. Create a VLAN, an inVlan ACT, and an ACL. Then, create two ACEs; the key step is to create the ARP request ACE, which solves the ACL operation issue.

Procedure steps

1. Create the VLAN:

   ```
   ERS8610:5# vlan 3000 create byport 1 color 5
   ERS8610:5# vlan 3000 ports add 2/1-2/48
   ERS8610:5# vlan 3000 ip create 172.30.0.252/24
   ERS8610:5# vlan 3000 ip vrrp 5 address 172.30.0.254
   ERS8610:5# vlan 3000 ip vrrp 5 backup-master enable
   ERS8610:5# vlan 3000 ip vrrp 5 enable
   ```

2. Create the ACT and ACL:

   ```
   ERS8610:5# filter act 1 create name "test-ACT-1"
   ERS8610:5# filter act 1 ip srcIp
   ERS8610:5# filter act 1 arp operation
   ERS8610:5# filter act 1 apply
   ERS8610:5# filter acl 1 create inVlan act 1 name "test-ACL-1"
   ERS8610:5# filter acl 1 set default-action deny
   ERS8610:5# filter acl 1 vlan add 3000
   ```

3. Create the ACEs:

   These ACEs filter based on the source IP addresses of 172.30.0.100, 172.30.0.252, and 172.30.0.254 and permit ARP requests. The key part of this workaround is to

configure the ACE to permit ARP requests. Ensure that the ACE you add to permit ARP requests uses a unique ACE ID.

```
ERS8610:5# filter acl 1 ace 1 create name "arp"
ERS8610:5# filter acl 1 ace 1 action permit
ERS8610:5# filter acl 1 ace 1 arp operation eq arprequest
ERS8610:5# filter acl 1 ace 1 enable
ERS8610:5# filter acl 1 ace 2 create name ip
ERS8610:5# filter acl 1 ace 2 action permit
ERS8610:5# filter acl 1 ace 2 ip src-ip eq 172.30.0.100
ERS8610:5# filter acl 1 ace 2 enable
ERS8610:5# filter acl 1 ace 3 create name ip2
ERS8610:5# filter acl 1 ace 3 action permit
ERS8610:5# filter acl 1 ace 3 ip src-ip eq 172.30.0.252
ERS8610:5# filter acl 1 ace 3 enable
ERS8610:5# filter acl 1 ace 4 create name ip3
ERS8610:5# filter acl 1 ace 4 action permit
ERS8610:5# filter acl 1 ace 4 ip src-ip eq 172.30.0.254
ERS8610:5# filter acl 1 ace 4 enable
```

Glossary

access control entry (ACE)	One of the filter rules that comprise an access control list (ACL). An ACE statement defines pattern match criteria for a packet and the desired behavior for packets that carry the pattern. When the packets match an ACE rule, the specified action executes.
access control list (ACL)	An ordered list of filter rules referred to as access control entries. The ACEs provide specific actions, such as dropping packets within a specified IP range, or a specific Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) port or port range. When an ingress or egress packet meets the match criteria specified in one or more ACEs within an ACL, the corresponding action executes.
class of service (CoS)	A method used to manage traffic congestion based on the CoS level assigned to the packet.
Layer 2	The Data Link Layer of the OSI model. Examples of Layer 2 protocols are Ethernet and Frame Relay.
Layer 3	The Network Layer of the OSI model. Example of a Layer 3 protocol is Internet Protocol (IP).
Local Area Network (LAN)	A data communications system that lies within a limited spatial area, uses a specific user group and topology, and can connect to a public switched telecommunications network (but is not one).
per-hop behavior (PHB)	A traffic class forwarding treatment based on criteria defined in the DiffServ field.
quality of service (QoS)	Use QoS features to reserve resources in a congested network. For example, you can configure a higher priority to IP deskphones, which need a fixed bit rate, and, split the remaining bandwidth between data connections if calls in the network are important than the file transfers.
User Datagram Protocol (UDP)	In TCP/IP, a packet-level protocol built directly on the Internet Protocol layer. TCP/IP host systems use UDP for application-to-application programs.
Voice over IP (VOIP)	The technology that delivers voice information in digital form in discrete packets using the Internet Protocol (IP) rather than the traditional circuit-committed protocols of the public switched telephone network (PSTN).