QoS Policing on Catalyst 6500/6000 Series Switches

Document ID: 12493

Contents

Introduction

QoS policing on a network determines whether network traffic is within a specified profile (contract). This may cause out-of-profile traffic to drop or to be marked down to another differentiated services code point (DSCP) value to enforce a contracted service level. (DSCP is a measure of the QoS level of the frame.)

Do not confuse traffic policing with traffic shaping. Both ensure that traffic stays within the profile (contract). You do not buffer out-of-profile packets when you police traffic. Therefore, you do not affect transmission delay. You either drop the traffic or mark it with a lower QoS level (DSCP markdown). In contrast, with traffic shaping, you buffer out-of-profile traffic and smooth the traffic bursts. This affects the delay and delay variation. You can only apply traffic shaping on an outbound interface. You can apply policing on both inbound and outbound interfaces.

The Catalyst 6500/6000 Policy Feature Card (PFC) and PFC2 only support ingress policing. The PFC3 supports both ingress and egress policing. Traffic shaping is only supported on certain WAN modules for the Catalyst 6500/7600 series, such as the Optical Services Modules (OSMs) and FlexWAN modules. Refer to the Cisco 7600 Series Router Module Configuration Notes for more information

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions.

Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

QoS Policing Parameters

To set up policing, you define the policers and apply them to ports (port-based QoS) or to VLANs (VLAN-based QoS). Each policer defines a name, type, rate, burst, and actions for in-profile and out-of-profile traffic. Policers on Supervisor Engine II also support excess rate parameters. There are two types of policers: microflow and aggregate.

- **Microflow**—police traffic for each applied port/VLAN separately on a per-flow basis.
- **Aggregate**—police traffic across all of the applied ports/VLANs.

Each policer can be applied to several ports or VLANs. The flow is defined using these parameters:

- source IP address
- destination IP address
- Layer 4 protocol (such as User Datagram Protocol [UDP])
- source port number
- destination port number

You can say that packets that match a particular set of defined parameters belong to the same flow. (This is essentially the same flow concept as that which NetFlow switching uses.)

As an example, if you configure a microflow policer to limit the TFTP traffic to 1 Mbps on VLAN 1 and VLAN 3, then 1 Mbps is allowed for each flow in VLAN 1 and 1 Mbps for each flow in VLAN 3. In other words, if there are three flows in VLAN 1 and four flows in VLAN 3, the microflow policer allows each of these flows 1 Mbps. If you configure an aggregate policer, it limits the TFTP traffic for all flows combined on VLAN 1 and VLAN 3 to 1 Mbps.

If you apply both aggregate and microflow policers, QoS always takes the most severe action specified by the policers. For example, if one policer specifies to drop the packet, but another specifies to mark

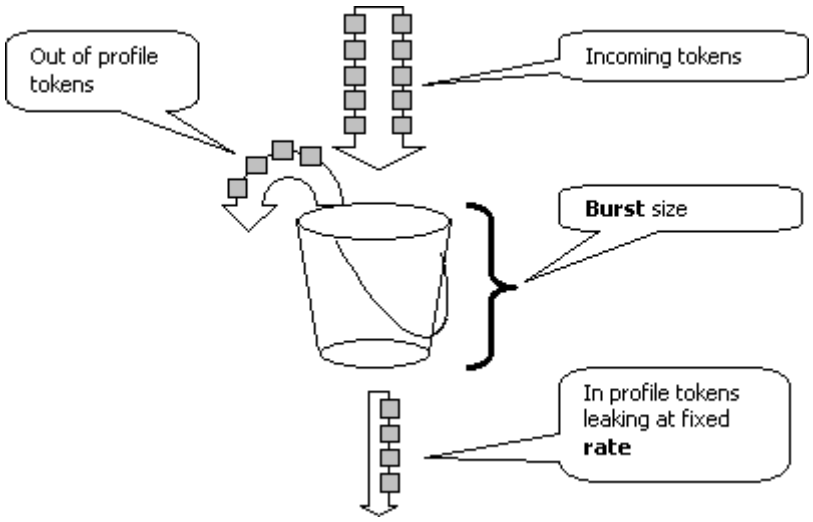
down the packet, the packet is dropped.

By default, microflow policers work only with routed (Layer 3 [L3]) traffic. To police bridged (Layer 2 [L2]) traffic as well, you need to enable bridged microflow policing. On the Supervisor Engine II, you need to enable bridged microflow policing even for L3 microflow policing.

Policing is protocol-aware. All traffic is divided into three types:

- IP
- Internetwork Packet Exchange (IPX)
- Other

Policing is implemented on the Catalyst 6500/6000 according to a "leaky bucket" concept. Tokens corresponding to inbound traffic packets are placed into a bucket. (Each token represents a bit, so a large packet is represented by more tokens than a small packet.) At regular intervals, a defined number of tokens are removed from the bucket and sent on their way. If there is no place in the bucket to accommodate inbound packets, the packets are considered out-of-profile. They are either dropped or marked down according to the configured policing action.



Note: The traffic is not buffered in the bucket, as it may appear in the image above. The actual traffic does not go through the bucket at all; the bucket is only used to decide whether the packet is in-profile or out-of-profile.

Calculate Parameters

Several parameters control the operation of the token bucket, as shown here:

- **Rate**—defines how many tokens are removed at each interval. This effectively sets the policing rate. All traffic below the rate is considered in-profile.
- **Interval**—defines how often tokens are removed from the bucket. The interval is fixed at 0.00025 seconds, so tokens are removed from the bucket 4,000 times per second. The interval cannot be

changed.

- **Burst**—defines the maximum number of tokens that the bucket can hold at any one time. To sustain the specified traffic rate, burst should be no less than the rate times the interval. Another consideration is that the maximum-size packet must fit into the bucket.

To determine the burst parameter, use this equation:

- Burst = (Rate [bps]) * 0.00025 [sec/interval]) *or* (maximum packet size [bits]), whichever is greater.

For example, if you want to calculate the minimum burst value needed to sustain a rate of 1 Mbps on an Ethernet network, the rate is defined as 1 Mbps and the maximum Ethernet packet size is 1518 bytes. The equation is:

- Burst = (1,000,000 bps * 0.00025) *or* (1518 bytes * 8 bits/byte) = 250 *or* 12144.

The larger result is 12144, which you round to 13 kbps.

Note: In Cisco IOS® Software, the policing rate is defined in bits per second (bps), as opposed to kbps in Catalyst OS (CatOS). Also in Cisco IOS Software, the burst rate is defined in bytes, as opposed to kilobits in CatOS.

Note: Due to hardware policing granularity, the exact rate and burst is rounded to the nearest supported value. Be sure that the burst value is not less than the maximum-size packet. Otherwise, all packets larger than the burst size are dropped.

For example, if you try to set the burst to 1518 in Cisco IOS Software, it is rounded to 1000. This causes all frames larger than 1000 bytes to be dropped. The solution is to configure burst to 2000.

When you configure the burst rate, take into account that some protocols (like TCP) implement a flow-control mechanism that reacts to packet loss. For example, TCP reduces windowing by half for each lost packet. Consequently, when policed to a certain rate, the effective link utilization is lower than the configured rate. You can increase the burst to achieve better utilization. A good start for such traffic is to double the burst size. (In this example, the burst size is increased from 13 kbps to 26 kbps). Then, monitor performance and make further adjustments if needed.

For the same reason, it is not recommended to benchmark the policer operation using connection-oriented traffic. This generally shows lower performance than the policer permits.

Police Actions

As mentioned in the Introduction, the policer can do one of two things to an out-of-profile packet:

- drop the packet (the `drop` parameter in the configuration)
- mark the packet to a lower DSCP (the `policed-dscp` parameter in the configuration)

To mark down the packet, you must modify the policed DSCP map. The policed DSCP is set by default to remark the packet to the same DSCP. (No mark down occurs.)

Note: If "out-of-profile" packets are marked down to a DSCP that is mapped into a different output queue than the original DSCP, some packets may be sent out of order. For this reason, if the order of packets is important, it is recommended to mark down out-of-profile packets to a DSCP that is mapped to the same output queue as in-profile packets.

On the Supervisor Engine II, which supports excess rate, two triggers are possible:

- When traffic exceeds normal rate
- When traffic exceeds excess rate

One example of the application of excess rate is to mark down packets that exceed the normal rate and drop packets that exceed the excess rate.

Policing Features Supported by the Catalyst 6500/6000

As stated in the Introduction, the PFC1 on the Supervisor Engine 1a and the PFC2 on the Supervisor Engine 2 only support ingress (inbound interface) policing. The PFC3 on the Supervisor Engine 720 supports both ingress and egress (outbound interface) policing.

The Catalyst 6500/6000 supports up to 63 microflow policers and up to 1023 aggregate policers.

The Supervisor Engine 1a supports ingress policing, starting with CatOS version 5.3(1) and Cisco IOS Software Release 12.0(7)XE.

Note: A PFC or PFC2 daughter card is required for policing with the Supervisor Engine 1a.

The Supervisor Engine 2 supports ingress policing, starting with CatOS version 6.1(1) and Cisco IOS Software Release 12.1(5c)EX. The Supervisor Engine II supports the excess rate policing parameter.

Configurations with Distributed Forwarding Cards (DFCs) only support port-based policing. Also, the aggregate policer only counts traffic on a per-forwarding-engine basis, not per-system. The DFC and PFC are both forwarding engines; if a module (line card) does not have a DFC, it uses a PFC as a forwarding engine.

Policing Features Update for Supervisor Engine 720

Note: If you are unfamiliar with Catalyst 6500/6000 QoS policing, be sure to read the QoS Policing Parameters and Policing Features Supported by the Catalyst 6500/6000 sections of this document.

The Supervisor Engine 720 introduced these new QoS policing features:

- **Egress policing**. The Supervisor 720 supports ingress policing on a port or VLAN interface. It supports egress policing on a port or L3 routed interface (in the case of Cisco IOS System Software). All ports in the VLAN are policed on egress regardless of the port QoS mode (whether port-based QoS or VLAN-based QoS). Microflow policing is not supported on egress. Sample configurations are provided in the Configure and Monitor Policing in CatOS Software section and Configure and Monitor Policing in Cisco IOS Software section of this document.

- **Per-user microflow policing**. The Supervisor 720 supports an enhancement to microflow policing known as per-user microflow policing. This feature is only supported with Cisco IOS System Software. It allows you to provide a certain bandwidth for each user (per IP address) behind given interfaces. This is achieved by specifying a flow mask inside the service policy. The flow mask defines which information is used to differentiate between the flows. For example, if you specify a source-only flow mask, all traffic from one IP address is considered one flow. Using this technique, you can police traffic per user on some interfaces (where you have configured the corresponding service policy); on other interfaces, you continue to use the default flow mask. It is possible to have up to two different QoS flow masks active in the system at a given time. You can associate only one class with one flow mask. A policy can have up to two different flow masks.

Another important change in policing on the Supervisor Engine 720 is that it can count traffic by the L2 length of the frame. This differs from the Supervisor Engine 2 and Supervisor Engine 1, which count IP and IPX frames by their L3 length. With some applications, L2 and L3 length may not be consistent. One example is a small L3 packet inside a large L2 frame. In this case, the Supervisor Engine 720 may display a slightly different policed traffic rate as compared with the Supervisor Engine 1 and Supervisor Engine 2.

Configure and Monitor Policing in CatOS Software

The policing configuration for CatOS consists of three major steps:

1. Define a policer—the normal traffic rate, excess rate (if applicable), burst, and policing action.
2. Create a QoS ACL to select traffic to police, and attach a policer to this ACL.
3. Apply the QoS ACL to the necessary ports or VLANs.

This example shows how to police all traffic to UDP port 111 on port 2/8.

Catalyst 6500/6000

```
set qos enable

!--- This enables QoS.

set qos policer aggregate udp_1mbps rate 1000 burst 13 drop

!--- This defines a policer. For the calculation of rate and burst,
!--- refer to Calculate Parameters.

set qos acl ip udp_qos_port dscp 0 aggregate udp_1mbps udp any any eq 111

!--- This creates QoS ACL to select traffic and attaches
!--- the policer to the QoS ACL.

commit qos acl all

!--- This compiles the QoS ACL.

set qos acl map udp_qos_port 2/8

!--- This maps the QoS ACL to the switch port.
```

The next example is the same; however, in this example, you attach the policer to a VLAN. Port 2/8 belongs to VLAN 20.

Note: You need to change the port QoS to `vlan-based` mode. Do this with the **set port qos** command.

This policer evaluates traffic from all ports in that VLAN configured for VLAN-based QoS:

Catalyst 6500/6000

```
set qos enable

!--- This enables QoS.

set qos policer aggregate udp_1mbps rate 1000 burst 13 drop

!--- This defines a policer. For the calculation of rate and burst,
!--- refer to Calculate Parameters.

set qos acl ip udp_qos_vlan dscp 0 aggregate udp_1mbps udp any any eq 111

!--- This creates the QoS ACL to select traffic and attaches
!--- the policer to QoS ACL.

commit qos acl all

!--- This compiles the QoS ACL.

set port qos 2/8 vlan-based

!--- This configures the port for VLAN-based QoS.

set qos acl map udp_qos_vlan 20

!--- This maps QoS ACL to VLAN 20.
```

Next, instead of dropping out-of-profile packets with DSCP 32, mark them down to a DSCP of 0 (best effort).

Catalyst 6500/6000

```
set qos enable

!--- This enables QoS.

set qos policer aggregate udp_1mbps rate 1000 burst 13 policed-dscp

!--- This defines a policer. For the calculation of rate and burst,
!--- refer to Calculate Parameters.

set qos acl ip udp_qos_md trust-ipprec aggregate udp_1mbps udp any any
eq 111 dscp-field 32
```

```
!--- Note: The above command should be on one line.

!--- This creates the QoS ACL to select traffic and attaches
!--- the policer to the QoS ACL.

commit qos acl all

!--- This compiles the QoS ACL.

set qos policed-dscp-map 32:0

!--- This modifies the policed DSCP map to mark down DSCP 32 to DSCP 0.

set port qos 2/8 vlan-based

!--- This configures the port for VLAN-based QoS.

set qos acl map udp_qos_md 20

!--- This maps the QoS ACL to VLAN 20.
```

This example shows the configuration for egress policing for the Supervisor Engine 720 only. It shows how to police all outgoing IP traffic on VLAN 3 to 10 Mbps aggregate.

Catalyst 6500/6000

```
set qos enable

!--- This enables QoS.

set qos policer aggregate egress_10mbps rate 10000 burst 20 drop

!--- This defines a policer. For the calculation of rate and burst,
!--- refer to Calculate Parameters.

set qos acl ip egress_pol trust-ipprec aggregate egress_10mbps ip any any

!--- This creates the QoS ACL to select traffic and attaches
!--- the policer to the QoS ACL.

commit qos acl all

!--- This compiles the QoS ACL.

set qos acl map egress_pol 3 output

!--- This maps the QoS ACL to VLAN 3 in the output direction.
```

Use **show qos maps runtime policed-dscp-map** to see the current policed DSCP map.

Use **show qos policer runtime {*policer_name* | all}** to verify the parameters of the policer. You can also see the QoS ACL to which the policer is attached.

Note: With Supervisor Engine 1 and 1a, it is not possible to have policing statistics for individual

aggregate policers. To view the per-system policing statistics, use this command:

```
Cat6k> (enable) show qos statistics l3stats
Packets dropped due to policing: 1222086
IP packets with ToS changed: 27424
IP packets with CoS changed: 3220
Non-IP packets with CoS changed: 0
```

To check microflow policing statistics, use this command:

```
Cat6k> (enable) show mls entry qos short
Destination-IP  Source-IP               Port  DstPrt SrcPrt Uptime       Age
--------------- --------------- ----- ------ ------ -------- --------

IP bridged entries:
239.77.77.77    192.168.10.200  UDP   63         63             00:22:02 00:00:
Stat-Pkts : 165360
Stat-Bytes : 7606560
Excd-Pkts : 492240
Stat-Bkts : 1660
239.3.3.3               192.168.11.200  UDP       888    777    00:05:38 00:00:
Stat-Pkts : 42372
Stat-Bytes : 1949112
Excd-Pkts : 126128
Stat-Bkts : 1628

Only out of the profile MLS entries are displayed
Cat6k> (enable)
```

With the Supervisor Engine II, you can view aggregate policing statistics on a per-policer basis with the **show qos statistics aggregate-policer** command.

For this example, a traffic generator is attached to port 2/8. It sends 17 Mbps of UDP traffic with destination port 111. You expect the policer to drop 16/17 of the traffic, so 1 Mbps should go through:

```
Cat6k> (enable) show qos statistics aggregate-policer udp_1mbps
QoS aggregate-policer statistics:
Aggregate policer       Allowed packet Packets exceed Packets exceed
                count                  normal rate           excess rate
------------------------------ ------------- ------------- -------------
udp_1mbps       582439                 9732108                9732108

Cat6k> (enable) show qos statistics aggregate-policer udp_1mbps
QoS aggregate-policer statistics:
Aggregate policer       Allowed packet Packets exceed Packets exceed
                count                  normal rate           excess rate
------------------------------ ------------- ------------- -------------
udp_1mbps       582504                 9733198                9733198
```

Note: Notice that allowed packets have increased by 65 and excess packets have increased by 1090. This means that the policer has dropped 1090 packets and allowed 65 to pass through. You can calculate that 65 / (1090 + 65) = 0.056, or roughly 1/17. Therefore, the policer works correctly.

Configure and Monitor Policing in Cisco IOS Software

The configuration for policing in Cisco IOS Software involves these steps:

1. Define a policer.
2. Create an ACL to select traffic to police.
3. Define a class map to select traffic with ACL and/or DSCP/IP precedence.
4. Define a service policy that uses class, and apply the policer to a specified class.
5. Apply the service policy to a port or VLAN.

Consider the same example as that provided in the section Configure and Monitor Policing in CatOS Software, but now with Cisco IOS Software. For this example, you have a traffic generator attached to port 2/8. It sends 17 Mbps of UDP traffic with destination port 111:

Catalyst 6500/6000

```
mls qos

!--- This enables QoS.

mls qos aggregate-policer udp_1mbps 1000000 2000 conform-action
transmit exceed-action drop

!--- Note: The above command should be on one line.

!--- This defines a policer. For the calculation of rate and burst,
!--- refer to Calculate Parameters.
!--- Note: The burst is 2000 instead of 1518, due to hardware granularity.

access-list 111 permit udp any any eq 111

!--- This defines the ACL to select traffic.

class-map match-all udp_qos
match access-group 111

!--- This defines the traffic class to police.

policy-map udp_policy
class udp_qos
police aggregate udp_1mbps

!--- This defines the QoS policy that attaches the policer to the traffic class.

interface GigabitEthernet2/8
switchport
service-policy input udp_policy

!--- This applies the QoS policy to an interface.
```

There are two types of aggregate policers in Cisco IOS Software: **named** and **per-interface**. The named aggregate policer polices the traffic combined from all interfaces to which it is applied. This is the type

used in the above example. The per-interface policer polices traffic separately on each inbound interface to which it is applied. A per-interface policer is defined within the policy map configuration. Consider this example, which has a per-interface aggregate policer:

Catalyst 6500/6000

```
mls qos

!--- This enables QoS.

access-list 111 permit udp any any eq 111

!--- This defines the ACL to select traffic.

class-map match-all udp_qos
match access-group 111

!--- This defines the traffic class to police.

policy-map udp_policy
class udp_qos

!--- This defines the QoS policy that attaches the policer to the traffic class.

police 1000000 2000 2000 conform-action transmit exceed-action drop

!--- This creates a per-interface aggregate
!--- policer and applies it to the traffic class.

interface GigabitEthernet2/8
switchport
service-policy input udp_policy

!--- This applies the QoS policy to an interface.
```

Microflow policers are defined within the policy map configuration, as are per-interface aggregate policers. In the example below, every flow from host 192.168.2.2 that comes into VLAN 2 is policed to 100 kbps. All traffic from 192.168.2.2 is policed to 500 kbps aggregate. VLAN 2 includes interfaces fa4/11 and fa4/12:

Catalyst 6500/6000

```
mls qos

!--- This enables QoS.

access-list 1 permit 192.168.2.2

!--- This defines the access list to select traffic from host 192.168.2.2.

class-map match-all host_2_2
match access-group 1

!--- This defines the traffic class to police.

policy-map host
```

```
class host_2_2

!--- This defines the QoS policy.

police flow 100000 2000 conform-action transmit exceed-action drop

!--- This defines a microflow policer. For the calculation of rate and
!--- burst, refer to Calculate Parameters.

police 500000 2000 2000 conform-action transmit exceed-action drop

!--- This defines the aggregate policer to limit
!--- traffic from the host to 500 kbps aggregate.

interface fa4/11
mls qos vlan-based
interface fa4/12
mls qos vlan-based

!--- This configures interfaces in VLAN 2 for VLAN-based QoS.

interface vlan 2
service-policy input host

!--- This applies the QoS policy to VLAN 2.
```

The example below shows a configuration for egress policing for the Supervisor Engine 720. It establishes the policing of all outbound traffic on interface Gigabit Ethernet 8/6 to 100 kbps:

Catalyst 6500/6000

```
mls qos

!--- This enables QoS.

access-list 111 permit ip any any

!--- This defines the ACL to select traffic. All IP traffic is subject to policing.

class-map match-all cl_out
 match access-group 111

!--- This defines the traffic class to police.

policy-map pol_out
class cl_out
police 100000 3000 3000 conform-action transmit exceed-action drop

!--- This creates a policer and attaches it to the traffic class.

interface GigabitEthernet8/6
ip address 3.3.3.3 255.255.255.0
service-policy output pol_out

!--- This attaches the policy to an interface.
```

The example below shows a configuration for per-user policing for the Supervisor Engine 720. Traffic that comes in from users behind port 1/1 toward the Internet is policed to 1 Mbps per user. Traffic that comes from the Internet toward the users is policed to 5 Mbps per user:

Catalyst 6500/6000

```
mls qos

!--- This enables QoS.

access-list 111 permit ip any any

!--- This defines the ACL to select user traffic.

class-map match-all cl_out
match access-group 111

!--- This defines the traffic class for policing.

policy-map pol_out
class cl_out
police flow mask src-only 1000000 32000 conform-act transmit exceed-act drop

!--- Only the source IP address is considered for flow creation
!--- on interfaces with this policy attached.

interface gigabit 1/1

!--- 1/1 is the uplink toward the users.

service-policy input pol_out

!--- Traffic comes in from users, so the policy is attached
!--- in the input direction.

class-map match-all cl_in
match access-group 111
policy-map pol_in
class cl_in
police flow mask dest-only 5000000 32000 conform-act transmit exceed-act drop

!--- Only the destination IP address is considered for flow creation
!--- on interfaces with this policy attached.

interface gigabit 1/2

!--- 1/2 is the uplink to the Internet.

service-policy input pol_in
```

To monitor policing, you can use these commands:

```
bratan# show mls qos
QoS is enabled globally
Microflow policing is enabled globally
QoS global counters:
Total packets: 10779
IP shortcut packets: 0
```

```
Packets dropped by policing: 2110223
IP packets with TOS changed by policing: 0
IP packets with COS changed by policing: 0
Non-IP packets with COS changed by policing: 0

bratan# show mls qos ip gigabitethernet 2/8
[In] Policy map is udp_policy [Out] Default.
QoS Summary [IP]: (* - shared aggregates, Mod - switch module)

Int   Mod Dir Class-map DSCP AgId Trust FlId AgForward-Pk AgPoliced-Pk
--------------------------------------------------------------------------------
Gi2/8 1   In  udp_qos   0    1*   No    0        127451           2129602


bratan# show mls qos ip gigabitethernet 2/8
[In] Policy map is udp_policy [Out] Default.
QoS Summary [IP]: (* - shared aggregates, Mod - switch module)

Int   Mod Dir Class-map DSCP AgId Trust FlId AgForward-Pk AgPoliced-Pk
--------------------------------------------------------------------------------
Gi2/8 1   In  udp_qos   0    1*   No    0        127755           2134670
```

Note: Allowed packets have increased by 304 and excess packets have increased by 5068. This means that the policer has dropped 5068 packets and allowed 304 to pass through. Given the input rate of 17 Mbps, the policer should pass 1/17 of the traffic. If you compare the dropped and forwarded packets, you see that this has been the case: 304 / (304 + 5068) = 0.057, or roughly 1/17. Some minor variation is possible due to hardware policing granularity.

For microflow policing statistics, use the **show mls ip detail** command:

```
Orion# show mls ip detail
IP Destination IP Source        Protocol L4 Ports     Vlan Xtag L3-protocol
--------------+---------------+--------+-------------+----+----+-----------+
192.168.3.33    192.168.2.2     udp     555 / 555       0   1           ip
192.168.3.3     192.168.2.2     udp     63 / 63         0   1           ip

[IN/OUT] Ports Encapsulation RW-Vlan RW-MACSource       RW-MACDestination
--------------+-------------+-------+--------------+-----------------+---------
Fa4/11 - ----   ARPA            3    0030.7137.1000  0000.3333.3333     314548
Fa4/11 - ----   ARPA            3    0030.7137.1000  0000.2222.2222     314824

Packets      Age   Last Seen    QoS      Police Count Threshold Leak
------------+-----+---------+-----------+------------+---------+-----------+
6838         36    18:50:09     0x80      346197        62*2^5   3*2^0
6844         36    18:50:09     0x80      346695        62*2^5   3*2^0

Drop Bucket  Use-Tbl Use-Enable
----+-------+-------+----------+
YES   1968     NO       NO
YES   1937     NO       NO
```

Note: The `Police Count` field shows the number of policed packets per flow.

NetPro Discussion Forums - Featured Conversations

Networking Professionals Connection is a forum for networking professionals to share questions,

suggestions, and information about networking solutions, products, and technologies. The featured links are some of the most recent conversations available in this technology.

NetPro Discussion Forums - Featured Conversations for LAN
Network Infrastructure: LAN Routing and Switching
QOS Configuration on Catalyst 3550 - Jan 2, 2006 Multicast traffic issue - Jan 2, 2006 MSTP / RSTP convergence - Jan 2, 2006 STP problem - Jan 2, 2006 SFP problem with 4507R - Jan 2, 2006
Network Infrastructure: Getting Started with LANs
Where to get those symbols for Visio - Dec 31, 2005 Access-lists - Dec 31, 2005 Switch 2950 catalyst problem - Dec 30, 2005 Howto configure VLAN id > 1005 in Ethernet Switching Network Modules - Dec 29, 2005 VLAN security - Dec 29, 2005

Related Information

- **Configuring QoS**
- **Understanding Quality of Service on Catalyst 6000 Family Switches**
- **LAN Product Support Pages**
- **LAN Switching Support Page**
- **Technical Support & Documentation - Cisco Systems**

Updated: Jan 02, 2006 Document ID: 12493