Learn About Quality of Service (QoS)

This Learn About introduces you to the fundamentals of Quality of Service (QoS). It explains what QoS is, why it's essential in today's packet-based networks, the difference between QoS and the similarly-named Class of Service (CoS), and, finally, it reviews each of the core functions of a QoS implementation and how they are used.

Exactly how Juniper Networks devices support QoS is detailed at the end of this document.

Why QoS is Essential in Today's Networks

Historically, separate physical networks were used to carry voice and data traffic in telecommunications networks. Each network carried a certain type of traffic— voice, for example—and provided an inherent level of quality and availability that was required by the traffic it was carrying. Because these networks were specifically designed to carry voice or data traffic, network administrators didn't have to worry about managing such traffic; it was inherently designed into the networks.

Today, these same applications run on converged, packet-based networks where all traffic shares a common infrastructure as well as any network resources. These packet-based networks were originally intended to deliver traffic on a best-effort only basis. Packet-based networks have no inherent QoS.

The rapid increase in mobile devices and the associated explosion of traffic associated with them has compounded the issue. Subscribers of voice and video services demand that these services are always available, and that they have an acceptable level of quality. Businesses and banks running mission critical applications over the same network also demand high levels of service.

So how do packet-based networks pass this massive amount of traffic from Point A to Point B, and do it in accordance with the service contracts and performance requirements of all the applications generating the traffic?

The answer is QoS.

QoS or CoS?

Engineers, network managers, network equipment vendors, and their supporting staff use QoS and CoS interchangeably. After all, the terms look and sound similar, and their meanings are both inherently about network traffic. It's easy to see why people confuse the two.

This document defines QoS as *the manipulation of traffic such that the network device forwards it in a fashion consistent with the required behaviors of the applications generating that traffic*. Now that may sound like a politician answering a sensitive question, but it efficiently explains QoS at a basic level: QoS enables a network device to differentiate traffic and then apply different behaviors to that different traffic. To provide QoS solutions, network devices such as routers and switches differentiate the traffic by examining the packets as they enter the device and then *classifying* the traffic into groups, called *classes of service*. So, how do network devices know what QoS to apply to certain traffic? They know this based on the class of service associated with the traffic.

One reason the terms QoS and CoS are often used interchangeably is that QoS uses a class-based scheme for differentiating traffic, appropriately called class of service. Administrators of Juniper Network's devices also use these terms interchangeably because Junos uses a software construct called *Class-of-Service* (CoS) to implement end-to-end QoS solutions. In fact, the second half of this Learn About will introduce you to how Junos CoS configurations can enable QoS principles in your own network.

That's it. The terms QoS and CoS essentially mean the same thing. So, for purposes of this *Learn About*, let's simply use the industry standard term: QoS.

QoS

QoS enables you to differentiate or classify traffic based on various parameters in the received packet header. For example, you can classify traffic based on the type of traffic or the source or destination address in the received packet. Network devices examine parameters in the received packet, and then, based on the value of those parameters, the network device places the packet into different *classes of service*. Each class of service can have different QoS behaviors associated with it. *QoS behaviors tell the device how to treat the traffic as it travels from ingress interface all the way until it is sent out the egress interface of the network device*. The result is that you can treat traffic assigned to any one class of service differently from any other class of service, and in any manner you want in order to provide the desired QoS solution.

Which QoS behaviors a network device applies to the traffic in each class of service depends on the QoS capabilities and how the device is configured. For example, you might want to configure the following QoS behaviors:

- Prioritizing traffic over other traffic based on such things as type of protocol, a source or destination address, or a source or destination port number.
- Filtering traffic upon ingress or egress in any number of simple and complex ways.
- Controlling the allowed bandwidth transmitted or received on the interfaces of the device.
- Reading and writing QoS behavior requirements in the packet header.

- Controlling congestion so that when the device has too much traffic to send it sends the highest priority traffic first based on scheduler queuing priorities.
- Controlling packet loss using random early detection (RED) algorithms, so that when the device has too much traffic to send it knows which packets it can drop and which ones it must process.

You might associate video traffic with one class of service, because it is particularly sensitive to delays and packet loss. If your network connects users to training videos, for example, you don't want their videos to repeatedly freeze, so the QoS behavioral characteristics you want to configure for this class of service would be:

- High priority
- Low packet loss
- No impact whenever congestion occurs on the device

End-to-End QoS Solution Requires Per-Hop Configuration

To implement a complete end-to-end QoS solution it is critical that all traffic assigned to any single class of service receives c*onsistent treatment from each network device along the traffic path*. This doesn't mean every device in the traffic path must be configured identically; on the contrary, the QoS behaviors that each device applies to the packet can differ, but the resulting treatment of the traffic in that CoS must be consistent.

Note QoS behaviors are unidirectional and network operators must define them in both the ingress and egress directions on each device in the traffic path. This per-hop distinction is important and it is fundamental to understanding QoS. Defining QoS behaviors on a per-hop basis means that if even a single hop along the traffic path has no QoS configuration, or has an incorrect configuration, it can negate the entire end-to-end QoS implementation. The receiving device cannot correct this issue, even if its QoS behaviors are properly configured.

Traffic Characteristics

So you can see that QoS is essential for managing traffic in today's packet-based networks, and you should now know QoS solutions work by differentiating traffic into classes of service and then configuring QoS behaviors on a hop-by-hop basis.

But let's take a closer look at the traffic characteristics you are up against and trying to control:

- Loss: This is the failure of a packet, transmitted into the network at its source, to reach its intended destination. The loss of packets can either have very little impact or be detrimental, depending on the application. For example, if a VoIP application loses a few packets it will have little impact on a conversation between two people because it's possible for the parties to repeat what they said. If a file transfer experiences packet loss, however, it can reduce the transmission rate significantly, even if only a few packets are lost.
- Latency: This is the delay that takes place between the transmission of a packet into the network at its source and its arrival at its intended destination. Again, some packet delays have little impact and some have severe implications, depending on the application. For example, those listening to an Internet radio application can tolerate some packet delay but with voice applications, packet delays can cause intolerable echoes for callers.

- Jitter: This is the variation in latency between consecutive packets in a single flow. Jitter has the most significant impact on some of the most highly-valued services, such as voice and video services. Voice services use a digitization process to convert voice from analog to digital and back again at the receiving end. If the latency of consecutive packets during transmission varies in such a way that it causes the time between the arrival of those consecutive packets to differ too much, however, then the conversion from digital back to analog can fail, because the required packets are not present at the time necessary for them to be converted into a meaningful analog signal. When a video application experiences jitter, frames can get out of sequence or garbled.

These traffic characteristics must be controlled and managed on a hop-by-hop basis in order to achieve the per-hop QoS behaviors needed to provide a complete end-to-end QoS solution.

QoS Standards

To manage the loss, latency, and jitter in today's networks, the Internet Engineering Task Force (IETF) defined two models relevant to QoS in IP packet-based networks. The first model, Integrated Services (IntServ), where hosts signaled their QoS needs to the network, was never commonly deployed due to its complex implementation.

The second model, Differentiated Services (DiffServ), network devices use a class-based classification process, which classifies traffic into classes of service based on the *DS field* of the IP header. The DS field contains a 6-bit Differentiated Services Code Point (DSCP) value. The value of these bits defines the QoS per-hop behavior that the network device applies to the packet as it passes through the device. The QoS per-hop behavior identifies:

- The class of service to which the packet is assigned
- The drop precedence or packet loss preference, which define the loss characteristics allowed for the packet

Using the QoS DiffServ model's 6-bit DS field, a network device can have up to 64 different classes into which it can classify traffic, providing network administrators with great flexibility when defining the various classes of service on their network devices.

The IETF recommends the following commonly defined classes of service:

- *Best Effort* (BE) – This is the default class of service. Traffic that does not meet the requirements of any of the other defined classes is placed in the BE class.
- *Expedited Forwarding* (EF) – This class of service is dedicated to low-delay, low-packet-loss, and low-jitter and is typically used for voice, video, and other packets belonging to real-time applications. Packets in the EF class typically receives *strict priority* queuing, which enables delay-sensitive traffic to be transmitted with minimum delay. In general, packets that are queued in a *strict priority* queue are transmitted before any other packets, including *high-priority* queues.
- Assured Forwarding (AF) – The AF class of service assures packet delivery under defined conditions. Packets are transmitted as long as the traffic does not exceed a subscribed rate. Traffic that exceeds the subscribed rate has a higher probability of being dropped if congestion occurs.

The AF class of service is subdivided into four separate AF classes, each of which has the same priority, but within each class packets are given a *drop precedence* of high, medium, or low. The *higher* the drop precedence, the *more* packets that are dropped during periods of congestion. This combination of classes and drop precedence can enable the encoding of twelve possible AF classes, which enables network operators to configure very granular traffic classification schemes.

- Class Selector – Provides backward compatibility with network devices that do not support the DiffServ model. IPv4 networks use an IP Precedence field to identify the class of service. If a DiffServ-capable device receives a packet from a device that does not support the DiffServ model, the DiffServ-capable device can still determine which class of service the packet should be assigned to based on the value of the Class Selector code point.

The DiffServ model is accepted worldwide as the standard for implementing QoS solutions in IP packet-based networks.

The remainder of this document focuses on QoS using the DiffServ model.

Core QoS Functions

You should have a good understanding of QoS by now so let's look at the core functions performed *within* the network device and the QoS behaviors they can provide. The core QoS functions are:

- Classifiers
- Policers
- Shapers
- Queues
- Schedulers

Classifiers

QoS classifies each traffic type entering a device and then classifies those various traffic types into classes or classes of service. Traffic within each class is treated based on a set of QoS behaviors assigned to that class. Differentiating traffic and assigning it to the proper class is the responsibility of *classifiers*.

Classifiers have a single input, incoming packets, but they have multiple outputs (the various classes of service defined in the QoS configuration) into which to classify the various packets.

Classifiers determine which class a packet belongs to by using a set of IF/THEN rules with values that you configure. The IF rule specifies the match conditions to look for in the packet and the THEN rule specifies the class to which the packet should be classified. When the IF conditions are met, the packet is classified with the corresponding class.

A good QoS solution should allow you to define any number of match conditions for classifiers. Some match conditions can be very simple. For example, a classifier can simply examine the protocol or the receiving interface and assign all traffic from that protocol or interface to a particular class. Alternatively, a classifier can examine various fields in the packet header, and, based on the values of those fields, assign the packet to a specific class. For example, network devices that adhere to the Differentiated Services (DiffServ) specifications for QoS can examine the 6-bit DS field in the IP packet header to determine the level of service that should be applied to the packet.

Classification occurs on the ingress as packets enter each network device, although you can achieve more complex classification by defining classifiers that work in sequence. In this case, the first classifier classifies the packet by assigning the class of service. When the packet reaches the next classifier, it can be accepted, modified, or completely reclassified into a different class of service.

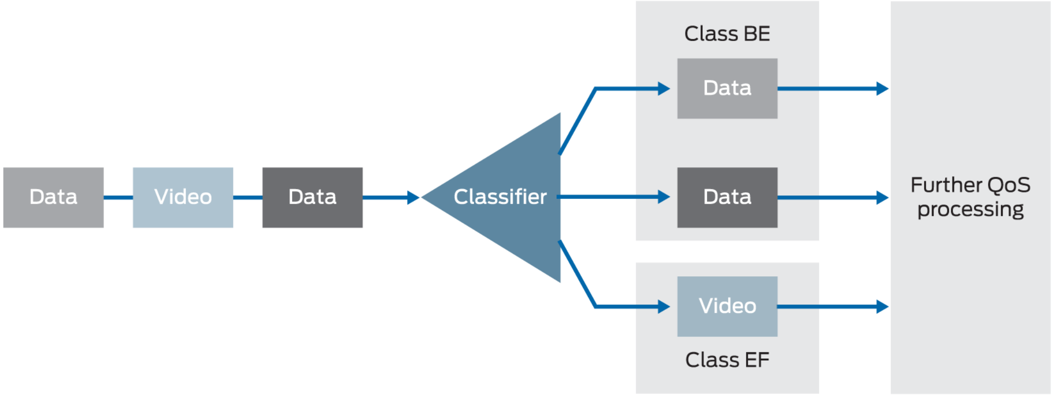


Figure 1 QoS Classifiers

Figure 1 illustrates the classification process. In this example, we have three packets entering the device: two data packets and one video packet. The network administrator has configured the two data packets to be classified into the BE class, and the video packet to be classified into the EF class, which is dedicated to low-delay, low-packet-loss, and low-jitter traffic.

Policers

Policers control traffic bursts by ensuring that incoming or outgoing traffic conforms to a configured rate called the *bandwidth limit*. Policers provide the first line of congestion management by preventing incoming traffic from overloading the network.

Policers rate-limit the traffic flow, either by discarding packets that exceed the configured bandwidth limit (called *hard policing*) or reassigning excess traffic to a different class of service (*soft policing*).

Figure 2 illustrates how policers manage traffic. If the incoming traffic is within the defined thresholds of the policer, it is accepted and sent for further QoS processing. However, if the traffic exceeds the defined thresholds of the policer, the policer can either drop the excess traffic, or it can reassign the excess traffic to a different forwarding class or loss priority and send it for further QoS processing.

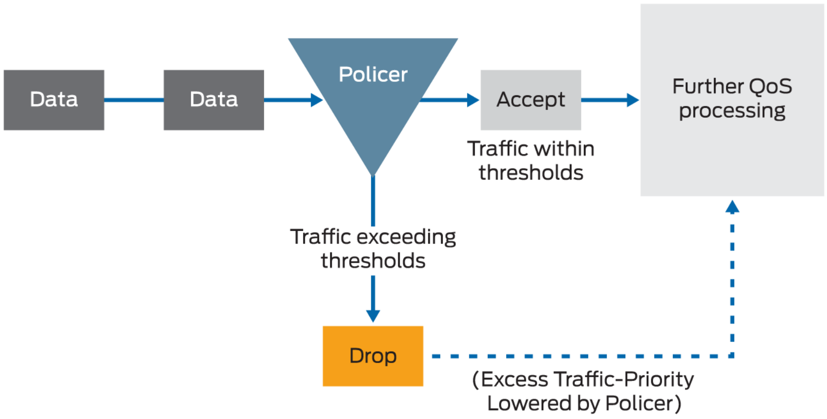

Data
Data
Policer
Accept
Further QoS processing
Traffic within thresholds
Traffic exceeding thresholds
Drop
(Excess Traffic-Priority Lowered by Policer)

Figure 2 QoS Policers

For example, let’s assume you have contracted a 10 Gbps Internet connection with your ISP. To ensure you cannot exceed that rate, your ISP configures a policer on the ingress interface of their device limiting your bandwidth to 10 Gbps. The first packet entering the ISP’s device is within the 10 Gbps rate but the second packet is above the rate. The policer accepts the first packet and can either drop the second packet, or, if configured to do so, can lower the priority of the packet and send it for further QoS processing.

Metering and Color Marking

In many QoS implementations, policers work in conjunction with *metering* and *color marking* tools to increase granularity. Metering measures the traffic arrival rate and assigns different colors to the traffic according to that rate. Metering works by comparing the actual rate of the traffic with the following two configurable values:

- Committed information rate (CIR): This is the guaranteed rate.
- Peak information rate (PIR): This is the maximum allowed traffic rate.

Metering measures the traffic comparing these two values and marks the traffic with colors that identify whether the traffic is *in-contract* or *out-of-contract* as follows:

- Green: Traffic rate is below the CIR and is in-contract.
- Yellow: Traffic rate falls between the CIR and PIR and is out-of-contract.
- Red: Traffic is above the PIR and is out-of-contract.

You can configure whether you want the device to forward the traffic or discard it based on the color.

Metering has one input, which is the traffic arrival rate, and it has three possible outputs: green, yellow, or red.

Shapers

Traffic shapers control congestion by applying a limit to the rate the network device can transmit traffic onto the physical media. Although this sounds similar to what policers do in QoS, there are several differences:

- Shapers act on *egress* traffic.
- Shapers act only on traffic that has already been granted access to a queue and is awaiting access to transmission resources.
- Shapers force egress traffic to conform to a bandwidth rate known as the *shaping rate*. However, shapers do not simply discard traffic above the defined shaping rate; they hold onto the packets until they can be transmitted without exceeding the rate.

Typically, your ISP traffic contract specifies the maximum rate that you are allowed to transmit packets and any traffic above this rate is dropped by the ISP as it enters the core network device. To avoid the traffic being dropped, you can define a shaper to shape the traffic to your contracted rate. The shaper holds any traffic above the rate until it can be transmitted without exceeding the rate.

Note Because shapers add delay by buffering traffic in the event of congestion, shaping can have a negative effect on real-time traffic like voice and video, which is sensitive to delay.

Queues

Queues hold packets while they await transmission resources. Packets may be buffered in a queue while awaiting transmission or they can be discarded. The decision to place a packet in a queue or discard it is made based on the queue's *fill level* as defined on the network device.

When a packet arrives at the queue the queue fill level is examined, and if the queue has reached its maximum fill level the packet is discarded. If the defined fill level for the queue has not been reached, the packet is placed in the queue where it stays until the (next) QoS building block, the scheduler, transmits it out to the network.

Schedulers

Schedulers control the order in which queues are serviced and packets are transmitted out to the next hop in the traffic path. Schedulers provide yet another level of differentiating traffic by deciding *which packets are sent*, *what order they are sent in*, and *how they are treated*. Figure 3 shows an example of how schedulers work.

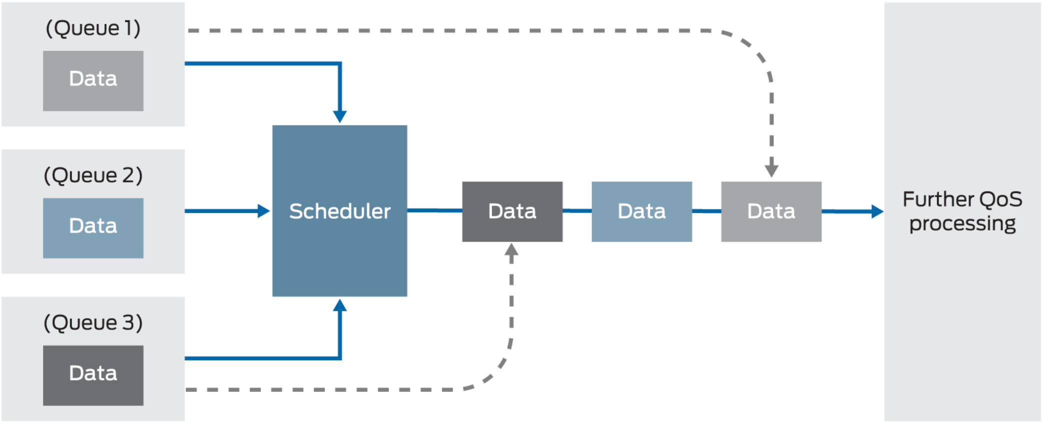


Figure 3 QoS Schedulers

In Figure 3, three queues have packets to transmit. The scheduler configuration is defined to service Queue1 first, Queue2 second, and Queue3 last. This is a very simple example of the capabilities of schedulers. However, scheduling capabilities are the core of QoS solutions because they provide ways to prioritize different types of traffic over others and they provide both congestion management and congestion avoidance techniques. Depending on the capabilities of the network device, schedulers can control such things as:

- The bandwidth assigned to the queue
- The length of time a packet can remain in a queue
- The order in which packets are transmitted out of queues
- Which class of service receives more or less access to the outbound interface (assuming you have a 1:1 mapping between classes of service and queues, which is recommended)
- What type of traffic is dropped during periods of congestion

Remarker

In the DiffServ model there is no end-to-end signaling of QoS requirements. The only way to signal QoS requirements to the next device in the traffic path is to use the QoS markings in the DS field of the packet header. This field identifies the per-hop behavior to apply to the packet and it includes both the class of service and packet loss parameters for the packet. A QoS remarker examines the DS field in the packet header and can modify this field prior to transmitting the packet so that the receiving device knows which QoS behaviors to apply to the packet.

You can use remarking to ensure the next device in the traffic path understands the QoS requirements for the packet. Remarking is performed on egress interfaces prior to packets leaving the device.

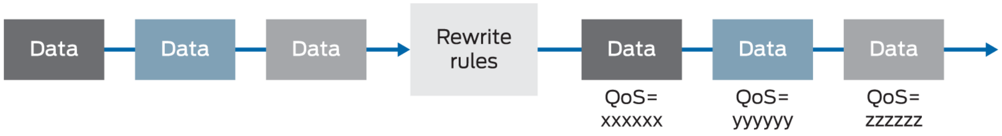


Figure 4 QoS Remarking

For example, in Figure 4 a Service Provider’s network device receives three packets. None of them are marked properly, so the provider’s device remarks the DS field so that subsequent routers will know how to handle the packet.

Pulling It All Together

Figure 5 illustrates all of the core functions of QoS and how they interact within the network device.

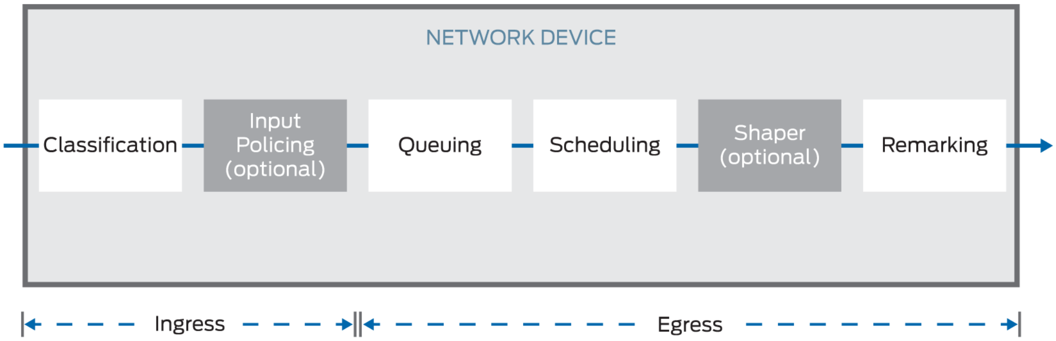


Figure 5 QoS Functions Summarized

The network device receives packets on the ingress interface, classifies the packets into the appropriate class/queue. If there is an optional policer configured, it rate-limits the traffic or assigns the traffic to a different class/queue. The scheduler takes the packets out of the queues and transmits them in the order configured for the scheduler. If there is a shaper configured, it shapes the traffic to the configured shaping-rate. Lastly, if remarking is configured, the device remarks the value of the DS-field so that the next device to receive the packet knows how to classify the it.

Juniper Networks QoS Capabilities

So far this Learn About has focused on helping you understand QoS and the core functions it can provide. Now let's get specific and introduce you to how Juniper Networks devices support all of these QoS requirements.

Juniper Networks devices running the Junos OS provide all of the previously discussed QoS behaviors, and more, through the Junos OS CoS software construct. In fact, the Junos OS CoS feature set provides mechanisms for applying QoS behaviors at a very granular level, making it one of the most flexible QoS solutions available today.

QoS Per-Hop-Behavior on Juniper Devices

You know by now that the QoS per-hop behavior (PHB) of a packet identifies:

- The class to which the packet is assigned.
- Packet loss preferences that define the loss characteristics for the packet by controlling which packets are dropped in the event of congestion. Packets with a higher packet loss preference are dropped first.

In the CoS feature set these two values are referred to as the *forwarding class* and *packet loss priority* (PLP), respectively . As packets travel through the various CoS components of the Juniper device, forwarding class and PLP are examined and can be manipulated to provide the required QoS behaviors.

Depending on the type of classifier you configure, the Junos OS CoS feature set can use the following packet header fields for determining the QoS PHB:

- DSCP, DSCP IPv6, or IP precedence – IP packet classification (Layer 3 headers)
- MPLS EXP – MPLS packet classification (Layer 2 headers)
- IEEE 802.1p – Packet classification (Layer 2 headers)
- IEEE 802.1ad – Packet classification for IEEE 802.1ad formats (including DEI bit)

The rest of this *Learn About* will simply refer to the bits that the CoS feature set examines, and that can be manipulated for the PHB, as *QoS markings*. If you want more information about the packet header fields, see *Links and References* at the end of this *Learn About*.

Hardware Dependency

The next few pages describe the general QoS capabilities of Juniper Networks devices. Realize that QoS capabilities require both hardware and software and that some QoS capabilities may require specific hardware. Always check your specific product documentation at http://www.juniper.net/documentation for the exact QoS capabilities of your device.

Classifiers

The Junos CoS feature set supports three types of classifiers:

- *Fixed Classifier*: Fixed classification simply looks at the ingress interface or VLAN on which the packet arrives and assigns all traffic received on that interface or VLAN to a certain class of service. You could do this on edge routers to classify all traffic from an untrusted source to a certain class.

- *Behavior Aggregate* (BA) *Classifier:* These classifiers classify traffic based on the QoS markings received in the packet header and are used when the QoS markings in the received packet are trusted.

  BA classifying requires less packet analysis and is more efficient in high-volume networks, especially in the network core. Because BA classifiers rely on *trusted* QoS markings in the packet, Juniper recommends they be used on core-router interfaces (and core-facing interfaces on edge routers), where QoS markings are trusted.

- *Multifield Classifiers*: Multifield classifiers differentiate traffic by examining *multiple* header fields in the received packet, including:
  - Source or destination address
  - Source or destination port number
  - IP protocol
  - DSCP value (DiffServ Code Point)

These classifiers use firewall filters and their associated match conditions to identify attributes in the received packet header and take action on *incoming* or *outgoing* packets, depending on whether the firewall filter is applied as an input or output filter.

A firewall filter configuration is specific to a particular protocol family and its associated match conditions. For multifield classifiers you can specify the following protocol families:

- Any protocol-independent traffic
- IPv4 and IPv6
- MPLS
- Virtual private LAN service (VPLS)
- Layer 2 circuit cross-connect (CCC)
- Layer 2 bridging traffic (MX Series only)

You can use BA and multifield classifers in conjunction with each other. The BA classifer is always applied first and the multifield classifer is applied last. Multifield classifers have the ability to override and fine-tune BA classification results.

Use a multifield classifier when the device sending the packet is either *not trusted* (or not able) to mark the QoS values in the packet header and there is a strong requirement for QoS. Typically multifield classifiers are used at the edge of the network because of the general lack of DSCP or IP precedence support in end-user applications.

Policers

To control the traffic rate, Junos OS policers can rate-limit a traffic flow either by discarding packets that exceed the configured threshold or reclassifying excess traffic to a different forwarding class and/or PLP. To control traffic bursts, CoS policers can limit the number of bytes allowed in a traffic burst. The Junos CoS feature set supports the use of policers in one of two ways:

- As part of a filter configuration: When you apply the filter to an interface, the policer is applied to all packets of the filter-specific protocol family that match the conditions specified in the filter configuration. With this method of applying a policer, you can define specific forwarding classes for an interface and apply traffic rate-limiting to each one.

- Apply a policer directly to an interface: This method applies traffic rate-limiting to all traffic on the specified interface, regardless of protocol family or any match conditions.

Policers can be applied on ingress or egress interfaces as:

- Inbound policers: Help conserve resources by dropping traffic that does not need to be routed through a network. This also helps to thwart denial-of-service (DoS) attacks.
- Outbound policers: Control the amount of bandwidth for either the logical or physical interface.

Junos OS policers also fully support the metering and color marking described earlier in this *Learn About*.

Shapers

Junos OS traffic shapers smooth traffic bursts by storing excess traffic in a buffer and delivering it only when bandwidth becomes available. Junos OS supports both dynamic and static traffic shaping. Dynamic shaping uses traffic-control profiles to dynamically configure shaping and enables you to configure both the PIR and the CIR. Static shaping uses only the PIR.

Junos OS supports traffic shaping on either the physical interface or the individual logical interfaces, which reside on the physical interface, but not both. Shaping capabilities are hardware dependent, but in general:

- For physical interfaces you can configure traffic shaping based on the rate-limited bandwidth of the total interface bandwidth.
- For logical interfaces you can configure traffic shaping by specifying the amount of bandwidth to be allocated to the logical interface (this is typically some portion of the physical interface bandwidth).

Queues and Schedulers

Junos OS applies a scheduler configuration to a forwarding class/queue combination by means of a *scheduler map*. You can associate each scheduler map with an interface, thereby configuring the hardware queues, packet schedulers, and congestion management characteristics that operate according to this mapping.

Schedulers are at the very core of QoS and in Junos OS you can configure the following parameters for schedulers:

- Transmit rate – Specifies the bandwidth assigned to a queue. Each queue is allocated some portion of the bandwidth on the outgoing interface. The transmit rate can be a fixed rate, a percentage of the total bandwidth for the interface, or the remainder of the available bandwidth. By default, when you exceed the transmit rate you can borrow extra bandwidth if it’s available on the interface, or you can limit the transmit rate to an exact value, which applies a hard limit with no buffering.
- Buffer size delay – This controls congestion by providing storage space to absorb traffic bursts up to the specified duration of delay. After the delay buffer becomes

full, packets with 100 percent drop probability are dropped from the buffer. You can assign a buffer-size delay as a percentage of total available buffer size or as a time or a remainder of an available buffer. When the buffer space is exceeded, you can allow more space if there is extra buffer space available for the interface, or you can set a hard limit. If you set a hard limit and you've reached the maximum bandwidth for that queue, you will start to drop packets if no extra storage is provided.

Note Remember that available buffer size always varies by hardware type.

Queue priority – Determines which queue(s) get to send traffic out on the interface first, second, and beyond. Higher priority queues are serviced before lower priority queues.

RED drop profiles – Manage congestion and work in conjunction with delay buffers. These profiles define when and how to drop packets when delay buffers fill up. As the delay buffers fill up, more and more packets are dropped. RED drop profiles take action on outgoing packets. You define drop profiles by specifying the buffer fill level and percentage of drop probability. You then map the drop profiles to a scheduler using a *drop-profile map*. The map sets the drop profile for a specific PLP and protocol type. You can associate multiple drop-profile maps with a single queue. For each scheduler, you can configure separate drop-profile maps for each loss priority. You can configure a maximum of 32 different drop profiles.

Remarker

Junos OS remarking enables you to alter the QoS markings in the packet header before transmitting the packet. To determine which QoS markings to place in the packet header, each rewrite rule examines the PHB (forwarding class and loss priority) in the received packet header, locates the chosen DSCP code-point alias from a table, and writes the respective DSCP bits into the packet header before it transmits the packet. The DSCP codes inform the next device in the traffic path of the required PHB to apply to the packet.

You configure remarking by using rewrite rules on the outbound interfaces of the device to meet the policies of the receiving peer device. This allows the receiving device to classify each packet into the appropriate class of service.

For every incoming packet, the ingress classifier decodes the ingress QoS markings into a forwarding class and PLP. The egress CoS information depends on which type of remarker is active, as follows:

- For MPLS EXP and IEEE 802.1 remarkers, values are derived from the forwarding class and loss priority values defined in the rewrite rules. MPLS EXP and IEEE 802.1 markers in the received packet are not preserved because they are part of the Layer 2 encapsulation.
- For IP precedence and DSCP remarkers, the marker alters the first three bits on the type of service (ToS) byte while leaving the last three bits unchanged.

Summary

You can see that the Junos OS CoS feature set provides a structured and very granular way to define and configure QoS. This granularity gives you flexibility in configuring a wide range of scheduling, queuing, and prioritizing techniques used for numerous end-to-end QoS solutions. But this Learn About has barely scratched the surface of Junos OS specifics and details of QoS configuration. Some of the many resources available to you for further study are listed below.

Links and References

Books

http://amzn.to/1OIAvhg

QoS Enabled Networks: Tools and Foundations, 2nd Edition, (February 2016) by Miguel Barreiros and Peter Lundqvist. This book provides an in-depth treatment of the subject ranging from a more theoretical level all the way through to an understanding of the tools available to influence the behaviors, and the application of those tools.

http://www.juniper.net/dayone

The Day One book series is available for free download in PDF format. Look for these two books to cover QoS fundamentals.

- *Day One: Deploying Basic QoS*
- *Day One: Junos QoS for IOS Engineers*

Juniper Networks Training

http://www.juniper.net/training/fasttrack

Try this *Learning Byte* from the award-winning training series written and validated by Juniper Networks.

- *Class of Service Basics Learning Byte*

Junos OS CoS Feature Set Documentation

Juniper TechLibrary includes everything you need to understand and configure all aspects of the Junos OS CoS feature set. The documentation set is both comprehensive and thoroughly reviewed by Juniper engineering.

See the following links for more information on specific CoS components::

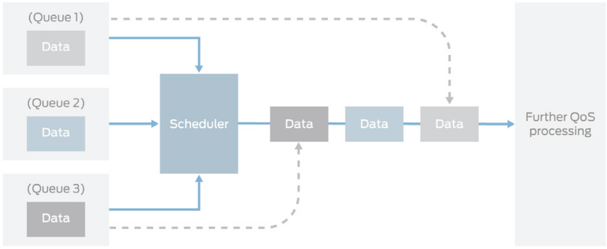
- For all CoS configuration, see *Class of Service Feature Guide for Routing Devices*
- For BA Classifiers, see *Understanding How Behavior Aggregate Classifiers Prioritize Trusted Traffic*

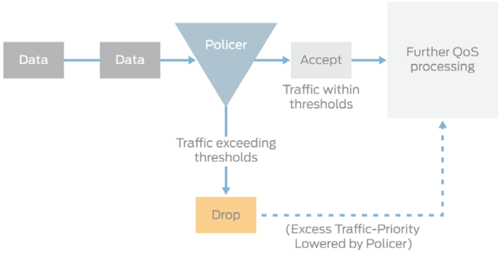
- For Multifield classifiers, see *Overview of Assigning Service Levels to Packets Based on Multiple Packet Header Fields*
- Any protocol-independent traffic. See *Firewall Filter Match Conditions for Protocol-Independent Traffic*
- IPv4 traffic. See *Match Conditions for IPv4 Traffic*
- IPv6 traffic. See *Match Conditions for IPv6 Traffic*
- MPLS traffic. See *Match Conditions for MPLS Traffic*
- Virtual private LAN service VPLS traffic. See *Match Conditions for VPLS Traffic*
- Layer 2 circuit cross-connection (CCC) traffic. See *Match Conditions for Layer 2 CCC Traffic*
- Layer 2 bridging traffic (MX Series only) See *Match Conditions for Layer 2 Bridging Traffic*
- For Policers, see *Traffic Policing Overview*
- For a Scheduler overview see: *Configuring Scheduler Transmission Rate*
- For Schedulers, see:
  - *Scheduler Overview*
  - *Configuring Scheduler Transmission Rate*
  - *Managing Congestion on Egress Interface by Configuring the Scheduler Buffer Size*
  - *Configuring Schedulers for Priority Scheduling*
- For Remarking, see *Configuring Rewrite Rules*

Learn About Quality of Service (QoS)

by Colleen Feerick

Learn about the fundamentals of Quality of Service (QoS) and all of its unique concepts and principles, and you'll soon understand why it's so essential in today's modern packet-based networks. In this volume you'll learn the difference between QoS and the (similarly-named) Class of Service (CoS), and you'll review the key functions of a QoS implementation and how Juniper Networks devices support it all. This is fascinating reading that's a must for anyone involved in modern networking.




Data
Data
Policer
Accept
Further QoS processing
Traffic within thresholds
Traffic exceeding thresholds
Drop
(Excess Traffic-Priority Lowered by Policer)

About the Author: Colleen Feerick is a Staff Information Development Engineer at Juniper Networks, with over 30 years of experience in technical support and information development in telecommunications networking.

For more information see:
juniper.net/documentation

Author's Acknowledgments: Many thanks to Miguel Barreiros for his review of this document and for the book *QoS Enabled Networks: Tools and Foundations, 2nd. Edition* by Miguel Barreiros and Peter Lundqvist, (John Wiley & Sons, 2011, ISBN 978-0-470-68697-3), which was invaluable to me when researching this project. Thank you to Kieran Milne of Juniper Networks for his excellent CoS Learning Bytes series. Thank you to Josef Jahde of Juniper Networks for his constant patience and guidance. Thank you to Ulf Bremer for his guidance on this review project. Finally, a huge thank you to Patrick Ames, who guided me through this process and whose help was invaluable to me throughout the entire project.

ISBN: 978-1-941441-27-5 Version History: First Edition, December 2015 2 3 4 5 6 7 8 9