

Configuring Queuing and Scheduling

SYSTEM ADMINISTRATOR GUIDE

Copyright

© Ericsson AB 2009-2011. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

SmartEdge is a registered trademark of Telefonaktiebolaget LM Ericsson.



Contents

1	Overview	1
1.1	Queue Maps	3
1.2	Congestion Management and Avoidance	4
2	Scheduling	7
2.1	Scheduling Algorithms	7
2.2	Queue Rates	10
2.3	MDRR and PWFQ Coexistence	10
3	Traffic Management	13
3.1	Hierarchical Scheduling in Traditional TM	13
3.2	Hierarchical Scheduling in Virtual-port TM	20
3.3	Port Grouping for Traffic Scheduling	22
3.4	Overhead Profiles	24
4	Configuration and Operations Tasks	25
4.1	Configuring a Queue Map	25
4.2	Configuring a Congestion Avoidance Map	25
4.3	Configuring an ATMWFQ Policy	26
4.4	Configuring an MDRR Policy	27
4.5	Configuring a PWFQ Policy	28
4.6	Configuring an Overhead Profile	29
4.7	Configuring a User-Defined Port Group Map and Applying It to a Card	30
4.8	Applying a Predefined or Default Port Group Map to a Card	31
4.9	Configuring VPCGs	31
4.10	Operations Tasks	34
5	Configuration Examples	37
5.1	Queue Maps	37
5.2	Congestion Avoidance Map for Multidrop Profiles	37
5.3	ATMWFQ Policies	38
5.4	MDRR Policy	38
5.5	PWFQ Policies	38
5.6	Overhead Profiles	41



5.7	QoS Port Group Maps	41
5.8	MDRR and PWFQ Coexistence	41
5.9	Traffic Management	42



1 Overview

This document provides an overview of the SmartEdge router quality of service (QoS) queuing and scheduling features and describes the tasks used to configure, monitor, and administer these features. This document also provides examples of QoS scheduling policy configurations.

This document applies to both the Ericsson SmartEdge® and SM family routers. However, the software that applies to the SM family of systems is a subset of the SmartEdge OS; some of the functionality described in this document may not apply to SM family routers.

For information specific to the SM family chassis, including line cards, refer to the SM family chassis documentation.

For specific information about the differences between the SmartEdge and SM family routers, refer to the Technical Product Description *SM Family of Systems* (part number 5/221 02-CRA 119 1170/1) in the **Product Overview** folder of this Customer Product Information library.

For information about other QoS configuration tasks and commands, see the following documents:

- *Configuring Rate-Limiting and Class-Limiting*—Rate- and class-limiting features (metering and policing policies)
- *Configuring Circuits for QoS*—Port, channel, and circuit configuration for all QoS policies and features
- *Configuring IP Multicast*—Configuring IGMP service profiles for QoS rate adjustment on multicast traffic.

The terms *traffic-managed circuit* and *traffic-managed port* refer to a circuit and port on a card that supports Traffic Management (TM). The following cards support TM:

- 12-port Fast Ethernet (FE) MIC card (copper and optical)
- 2-port Gigabit Ethernet (GE) MIC card (copper and optical, with native ports)
- 60-port Fast Ethernet–Gigabit Ethernet (60-port FE, 2-port GE) (fege-60-2-port)
- Advanced Gigabit Ethernet (4-port) (gigaether-4-port)
- Gigabit Ethernet 3 (4-port) (ge3-4-port)
- Gigabit Ethernet 1020 (10-port) (ge-10-port)
- Gigabit Ethernet 1020 (20-port) ge-20-port



- Gigabit Ethernet (5-port) (ge-5-port)
- Gigabit Ethernet (20-port) (ge4-20-port)
- Gigabit Ethernet DDR (10-port) (ge2-10-port)
- 10 Gigabit Ethernet (4-port) (10ge-4-port)
- POS OC-3c/STM-1c (8-port) (oc3e-8-port)
- POS OC-12c/STM-4c (4-port) (oc12e-4-port)
- POS OC-48c/STM-16c (4-port) (oc48e-4-port)
- OC-192c/STM-64c (1-port) (oc192-1-port)

The final stage of QoS enforcement for packets transiting the SmartEdge router before they are transmitted from traffic card interfaces is known as queuing. The operation of this stage is determined by the destination circuit of the packets (for example, a port or PVC, or subscriber session) and the associated queuing policy of that circuit.

Circuits may be subject to a QoS queuing policy that was explicitly assigned to the destination circuit through CLI configuration (for ports and PVCs, for example) or subscriber attributes for subscriber sessions. Whenever a queuing policy is explicitly assigned to a circuit, the system allocates a unique set of First In, First Out (FIFO) queues for use by the egress traffic of the circuit. The number of queues assigned to each circuit is determined by the `num-queues` parameter of the queuing policy, and is always equal to 1, 2, 4, or 8. A circuit with an associated set of queues is also called a queuing point.

Circuits that do not have an explicit queuing policy assignment inherit the queuing policy and share the queues of their nearest parent that does have a queuing policy assignment. If none of the circuits in the egress circuit's parental hierarchy up to the root port or link group have a queuing policy, the traffic of the circuit uses a default queue assigned to the port or link group, which is shared by all circuits on the port or link group that are not subject to a queuing policy through direct configuration or inheritance.

The queuing process can be broken down into three stages:

1. Queue assignment—The set of egress queues to be used by an exiting packet is determined by the egress circuit and its associated queuing policy (as described above). If the queue set includes more than one queue (for example, 2, 4, or 8), the individual queue in the set to be used is selected by applying the individual packet's PD QoS priority value to the queue map associated with the relevant queuing policy. For more information about queue maps, see Queue Maps.
2. Queue admittance—Whether a packet is allowed to enter its target egress queue is determined by the current number of packets currently stored in the queue awaiting final transmission. If the number of enqueued packets is equal to the configured depth of the queue, no more packets can be



admitted, and any additional packets targeted for the queue will be dropped until some packets are transmitted and more space is made available in the queue. Packets discarded in this way are referred to as tail drops. Tail drops are generally an indication of congestion, meaning that the rate of arrival of packets to the queue is greater than the rate of departure. The rate of arrival is primarily determined by the rate at which the packets were received by the SmartEdge router. The rate of departure is determined by the physical bandwidth of the egress interface, any applicable flow control, and the egress scheduler.

An optional mechanism affecting queue admittance called Random Early Discard (RED) can be enabled for a queue. Under RED, some packets may be discarded on a random basis as the occupancy of a queue approaches its maximum depth but before it is completely full. These early drops can act as a signal to some network protocols to begin reducing their bandwidth utilization before the more severe tail drop condition is encountered.

The queue admittance behavior for a particular queue is determined by parameters configured in the egress circuit's associated queuing policy or the congestion-avoidance-map referenced by that policy. See Congestion Management and Avoidance for more information.

3. Queue scheduling— Assignment and admittance apply to how forwarded packets enter egress queues; scheduling determines how packets are removed from queues and transmitted on the network. The scheduler determines the order and frequency in which packets are selected from the heads of all the various queues assigned to circuits on a physical or logical network interface and transmitted on the network. The basic scheduling algorithm and capabilities are determined by the style of queuing policy in use, and the details of scheduling behavior are determined by the configurable parameters of the queuing policy. Typical scheduling parameters include a maximum rate that packets might be allowed to egress for a particular queue, a collective rate for all the queues of a queuing point, or the relative weight to use when performing a round-robin selection between queues or queuing points.

1.1 Queue Maps

By default, the SmartEdge router assigns a PD QoS priority number to an egress queue, according to the number of queues configured on a circuit; see Table 1.

Table 1 Default Mapping of Packets into Queues Using PD QoS Priorities

PD QoS Priority Number	DSCP Value	IP Prec	MPLS EXP	802.1p	8 Queues	4 Queues	2 Queues	1 Queue
0	Network control	7	7	7	Queue 0	Queue 0	Queue 0	Queue 0
1	Reserved	6	6	6	Queue 1	Queue 1	Queue 1	Queue 0
2	Expedited Forwarding (EF)	5	5	5	Queue 2	Queue 1	Queue 1	Queue 0



Table 1 Default Mapping of Packets into Queues Using PD QoS Priorities

PD QoS Priority Number	DSCP Value	IP Prec	MPLS EXP	802.1p	8 Queues	4 Queues	2 Queues	1 Queue
3	Assured Forwarding (AF) level 4	4	4	4	Queue 3	Queue 2	Queue 1	Queue 0
4	AF level 3	3	3	3	Queue 4	Queue 2	Queue 1	Queue 0
5	AF level 2	2	2	2	Queue 5	Queue 2	Queue 1	Queue 0
6	AF level 1	1	1	1	Queue 6	Queue 2	Queue 1	Queue 0
7	Default Forwarding (DF)	0	0	0	Queue 7	Queue 3	Queue 1	Queue 0

You can configure a customized queue map and assign it to any scheduling policy. The map overrides the default mapping of packets into the egress queues of the policy to which it is assigned; see Figure 1. When the scheduling policy is attached to a circuit, it overrides the default queue map. You can configure up to three customized queue maps.

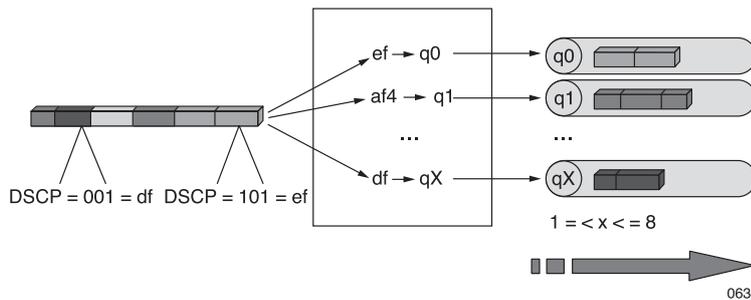


Figure 1 Queue Map (633)

1.2 Congestion Management and Avoidance

The SmartEdge router employs the following congestion avoidance features when processing packets using the different queuing and scheduling policies.

1.2.1 Random Early Detection

With scheduling policies, you can configure random early detection (RED) parameters to manage buffer congestion by signalling to sources of traffic that the network is on the verge of entering a congested state, rather than waiting until the network is actually congested. The technique is to drop packets with a probability that varies as a function of how many packets are waiting in a queue at any particular time, and the minimum and maximum average queue depth.

When a queue is nearly empty, the probability of dropping a packet is small. As the queue's average depth increases, the likelihood of dropping packets becomes greater; see Figure 2.



Note: For second-generation ATM OC traffic cards, and Ethernet traffic cards that support RED, the queue depth value is equal to the value configured for the maximum threshold.

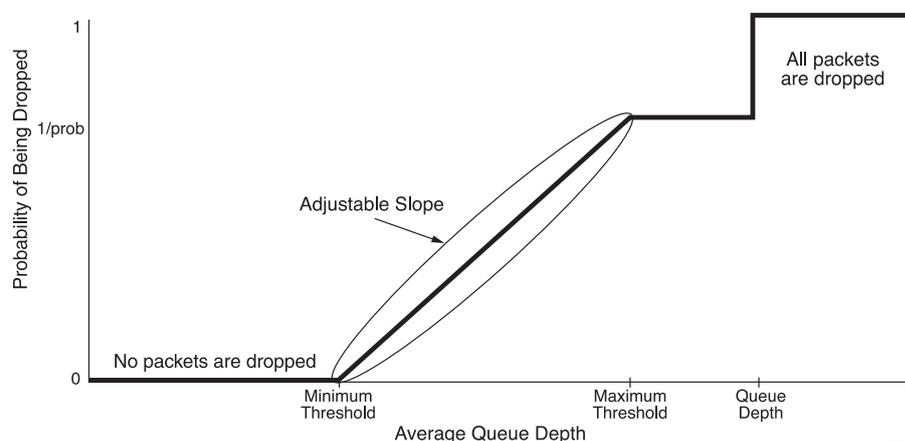


Figure 2 Probability of Being Dropped as a Function of Queue Depth (557)

1.2.2 Early Packet Discard

With ATMWFQ policies, you can also configure early packet discard (EPD), a congestion avoidance mechanism that starts dropping packets after queues reach the EPD threshold. When queue buffers are nearly full (reaching the EPD threshold), the system is signaled that it may become congested. Any packets trying to enter queues, after the EPD threshold has been met, are dropped.

1.2.3 Multidrop Precedence

With ATMWFQ and PWFQ policies, you can configure different congestion behaviors that depend on the DSCP values of the packets in a queue; this feature is referred to as multidrop precedence. Multidrop precedence supports up to three profiles for each queue, and each profile defines a different congestion behavior for one or more DSCP values. Each profile is also characterized by its RED parameter values. The DSCP value in the packet is used to select the profile that governs its congestion avoidance behavior.

Figure 3 shows how the three profiles can be defined with different minimum and maximum thresholds. Multidrop profiles are available only for ATMWFQ and PWFQ policies and are configured using congestion avoidance maps.

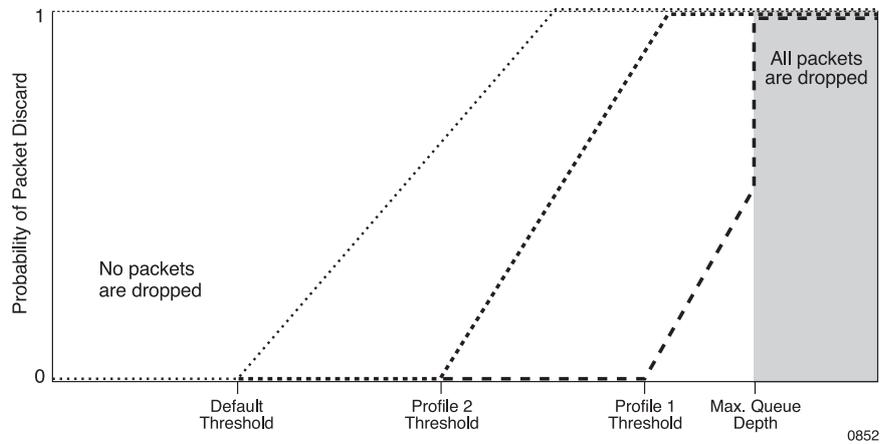


Figure 3 Multidrop Profiles (852)

1.2.4 Congestion Avoidance Maps

A congestion avoidance map specifies how congestion avoidance is managed for a set of queues. Each map supports eight queues.

Note: Congestion avoidance maps are supported only for ATMWFQ, MDRR, and PWFQ policies.

For each queue, you define up to three profiles, each of which describes the congestion behavior for one or more DSCP values. The map specifies RED parameters for every queue. One of the profiles, the default profile, specifies the default congestion behavior for every DSCP value.

When you define either of the other profiles for a queue, the system removes the DSCP values that you specify from the default profile. If a congestion map is not assigned to an ATMWFQ, MDRR, or PWFQ policy, packets are dropped only when the maximum queue depth is exceeded.

1.2.5 Queue Depth

With PWFQ and MDRR policies, you can modify the number of packets allowed per queue on a circuit. Queue depth is configured for PWFQ and MDRR policies with the `queue depth` command in the congestion avoidance map that you assign to the policy.



2 Scheduling

This section includes the following topics:

- Scheduling Algorithms
- Queue Rates
- MDRR and PWFQ Coexistence

2.1 Scheduling Algorithms

The SmartEdge router supports the following scheduling algorithms:

- Modified Deficit Round-Robin Policies
- Asynchronous Transfer Mode Weighted Fair Queuing Policies
- Priority Weighted Fair Queuing Policies

2.1.1 Modified Deficit Round-Robin Policies

Modified deficit round-robin (MDRR) policies support the following features:

- Three scheduling algorithms: WRR (weighted round-robin), strict mode, and priority queuing (PQ).
- Up to 256 congestion-avoidance maps to specify random early detection (RED) parameters
- One, two, four, or eight queues
- Single level of hierarchy
- Circuit rate limit for all scheduling modes

Limitations

When you configure MDRR policies, keep the following limitations in mind:

- Unlike PWFQ policies, MDRR policies do not support rate limits for queues.
- The total number of 802.1Q tunnels, 802.1Q PVCs , and subscribers that can be configured with their own MDRR policy on a traffic card varies depending on the num-queues configuration in the MDRR policies. See Table 2.



- The maximum number of unique MDRR rates that can be applied to circuits on a single slot by configuration in the MDRR policy using the `rate` command and per-circuit customization through the `rate circuit out` command is 200.

Table 2 Total Number of 802.1Q tunnels, 802.1Q PVCs, and Subscribers Configured with Their Own MDRR Policy on Specific Traffic Cards

num-queues Configuration in MDDR policy	Total number of 802.1Q tunnels, 802.1Q PVCs, and Subscribers That Can be Configured with Their Own MDRR Policy	
	On a PPA2 Traffic Card	On a PPA3 Traffic Card
num-queues equal to 8	1,700	475
num-queues equal to 4 or fewer	3,400	950

An MDRR policy can be applied to an access link group including 802.1Q PVCs under the link group. The constituent ports of the link group must support MDRR in order to configure an MDRR policy on a link group.

When you configure MDRR policies on access link groups, keep the following limitations in mind:

- When any type of queuing, including MDRR, is configured on a link group, you should bind the link group to only one port on any slot. Configurations where a link group is bound to two or more ports on the same slot are not supported when queuing is configured on the group. When MDRR is configured on an access link group, up to eight ports can be active (provided that each port is on a separate slot).
- MDRR is not supported on Economical access link groups.
- In access link groups, MDRR is supported on all TM traffic cards including Ethernet traffic cards. If MDRR is configured on an access link group, only these cards will be able to join the group.
- All constituents in a link group must have the same link speed. Additionally, the 1-port 10GE cannot be combined with the 4-port 10GE.
- Link Aggregation Control Protocol (LACP) traffic is not subject to the port queuing policy; however, LACP traffic is given priority.
- MDRR policies explicitly applied to subscriber records for subscribers over a link group are not supported.



2.1.1.1 MDRR Policy Modes

MDRR policies can operate in one of three modes: WRR, strict, or PQ:

- In WRR mode, queue 0 is treated like all other queues on a circuit. Each queue receives its share of the circuit's bandwidth according to the weight assigned to the queue.
- In strict mode, queue 0 always has priority over all other queues configured on a circuit.
- In PQ mode, output queues are serviced in strict priority order; that is, packets waiting in the highest-priority queue (queue 0) are serviced until that queue is empty, then packets waiting in the second-highest priority queue are serviced (queue 1), and so on. Under congestion, a PQ policy allows the highest priority traffic to get through, at the expense of lower-priority traffic.

With a PQ policy, the potential exists for a high volume of high-priority traffic to completely starve low-priority traffic. To prevent such starvation, the SmartEdge router allows a rate limit to be configured on each queue, which limits the amount of bandwidth available to a high priority queue. With careful tuning of the rate limits, you can prevent the lower priority queues from being starved.

2.1.2 Asynchronous Transfer Mode Weighted Fair Queuing Policies

Asynchronous Transfer Mode weighted fair queuing (ATMWFQ) policies ensure that queues do not starve for bandwidth and that traffic obtains predictable service. These policies operate in one of two modes: alternate and strict. In either mode, the ATM segmentation and reassembly (SAR) uses a class-based WFQ algorithm to perform QoS priority packet scheduling. In strict mode, queue 0 is serviced immediately and the other queues are serviced in a round-robin fashion according to their configured weights. In alternate mode, the servicing of queues alternates between queue 0 and the remaining queues, according to their configured weights. Queue 0 is served, then the next queue is served. Queue 0 is served again, and the next queue in turn is served, and so on. For example, if there are four queues configured, the order of servicing will be q0, q1, q0, q2, q0, q3, q0, q1, and so on.

When using ATMFWQ queuing policies, use the *show atm counters queue* command to obtain accurate per-queue statistics for ATM PVCs. Do not use the *show circuit counters detail* or *show circuit counters queue* commands to obtain per-queue statistics for ATM PVCs because the output for those commands always reports that all traffic is transmitted on queue zero.

2.1.3 Priority Weighted Fair Queuing Policies

The PD-QoS priority and the queue-map determine the egress queue. When PWFQ policies are enabled, the queue priority weight assigns an egress queue



to a TM queue priority group that determines the scheduling treatment for packets in the specified queue.

See Hierarchical Scheduling in Traditional TM for information about the PWFQ Scheduling algorithm.

2.2 Queue Rates

With MDRR policies, you can configure a rate limit. The rate is a combined traffic rate for all queues in the policy and is configured through the `rate` command (in MDRR policy configuration mode). A reasonable guideline for burst tolerance is to allow one to two seconds of burst time on the defined queue rate.

2.3 MDRR and PWFQ Coexistence

All Traffic Management (TM) capable cards support the coexistence of configured MDRR and PWFQ policies on circuits within a port. Each circuit can only exist in one schedule cone (either MDRR, or PWFQ) at a time. For a list of TM capable cards, see Overview.

MDRR and PWFQ policy coexistence allows you to selectively divide the traffic across hardware-based (MDRR) queues and software-based (PWFQ) queues. It also allows individual circuits to be scheduled at rates greater than the maximum allowed by PWFQ (1 Gbps). For example, a PVC used for carrying high bandwidth traffic, such as multicast video.

Table 3 lists the guidelines for MDRR and PWFQ policy coexistence along with any exceptions.



Table 3 MDRR and PWFQ Policy Coexistence Guidelines

Guideline	Exceptions
<p>MDRR bindings must always act as leaf nodes in the scheduling hierarchy. A leaf node refers to the last (or lowest) queuing and scheduling node in the context of the hierarchy as shown in the following examples of valid configurations:</p> <ul style="list-style-type: none"> Valid Configuration <pre>port ethernet 1/1 encapsulation dot1q dot1q pvc 1 encapsulation lqtunnel qos rate maximum 1024 qos policy queuing pwfq1 dot1q pvc 1:1 encapsulation pppoe qos policy queuing mdr1 <==== leaf node</pre> Valid Configuration <pre>port ethernet 1/1 encapsulation dot1q qos policy queuing pwfq1 dot1q pvc 1 encapsulation lqtunnel qos rate maximum 1024 qos policy queuing mdr1 <==== leaf node dot1q pvc 1:1 encapsulation pppoe</pre> <p>The following example shows an invalid configuration of a leaf node:</p> <p>Invalid Configuration <pre>port ethernet 1/1 encapsulation dot1q dot1q pvc 1 encapsulation lqtunnel qos rate maximum 1024 qos policy queuing mdr1 dot1q pvc 1:1 encapsulation pppoe qos policy queuing pwfq1 <==== invalid leaf node</pre> </p>	<p>Non leaf-node MDRR bindings are allowed on ports and top-level link group circuits.</p>
<p>PWFQ bindings may not be applied to a circuit that is subordinate in the circuit hierarchy to a circuit with an MDRR binding.</p>	<p>The parent port or link group may have an MDRR policy binding; this will not prevent the application and enforcement of PWFQ policy bindings on subordinate circuits like 802.1q PVCs and subscribers, but the MDRR policy parameters will not apply to the traffic of such subordinate circuits with a PWFQ binding.</p>
<p>A TM-capable port may have either an MDRR or PWFQ policy configured on it. The exception is that PWFQ policies cannot be applied on port or link group circuits within virtual-port TM cards.</p> <p>Currently, the only virtual-port TM card supported is the 10 GE (4-port) card (10ge-4-port).</p>	<p>None</p>

Table 4 provides a comparison of the MDRR and PWFQ policies at a high level.

Table 4 Comparison of MDRR and PWFQ Policies

PWFQ Policy Highlights	MDRR Policy Highlights
Implemented primarily in software	Implemented primarily in hardware
Large number of queues supported per card:	Smaller number of queues supported per card:
PPA2—24K x 8	PPA2—1.7K x 8 or 3.4K x 4
PPA3—32K x 8	PPA3—450 x 8 or 900 x 4

*Table 4 Comparison of MDRR and PWFQ Policies*

8 levels of scheduling priority all the way up the hierarchical tree	8 levels of scheduling priority within a queuing point and only 2 levels between queuing points
Four or more levels of hierarchical rate enforcement: TM queue priority group, queuing point, hierarchical node, port/link group	Only a single level of rate enforcement: queuing point
Schedules up to 1 Gbps per physical port, link group, or virtual-port	Schedules up to 10 Gbps per port or link group

MDRR and PWFQ schedule independently of each other. The MDRR scheduler does the final aggregation of the PWFQ and MDRR traffic using the MDRR scheduling algorithm. See Table 6 for the fixed mapping that the MDRR scheduler uses between the 8 PWFQ priorities and 2 MDRR priorities.



3 Traffic Management

The TM available in the SmartEdge router provides robust queuing and hierarchical scheduling capabilities and queuing servicing engine, along with the necessary packet buffering capabilities for the management of access networks. TM is used to manage oversubscription and service level agreement (SLA) enforcement, and provide differentiated levels of service for different types and classes of traffic on both layer 3 (for example, IP routed) and layer 2 (cross-connected, bridged, and so on) networks.

The basis of TM scheduling is the PWFQ queuing policy, which when applied to an individual circuit creates a TM queuing point (also called an L2 node). However, TM on the SmartEdge router also allows for the creation of additional intermediate nodes in the scheduling hierarchy for purposes of collective rate and weight scheduling enforcement. The creation and use of such additional scheduling nodes is called hierarchical scheduling.

The SmartEdge router supports two variants of TM:

- Traditional TM, which is supported on TM-capable Gigabit Ethernet (GE) cards and other TM-capable interfaces with a line speed equal to or less than 1 Gbps
- Virtual Port TM, which is only supported on a 10 GE (4-port) card (10ge-4-port)

The sections that follow provide more information about these TM types.

3.1 Hierarchical Scheduling in Traditional TM

This section describes hierarchical scheduling in traditional TM.

3.1.1 Conceptual Scheduling Levels

TM offers four conceptual scheduling levels:

- L1 Node: a TM queue priority group based on the queue assignment of the packet
- L2 Node: the queuing point (also known as a set of queues assigned to one or more circuits)
- L3 Node: an aggregation of subordinate L2 nodes and L3 nodes, or L2 or L3 nodes (also known as hierarchical node)
- L4 Node: the physical port or link group



See Figure 4 for illustration of these scheduling levels.

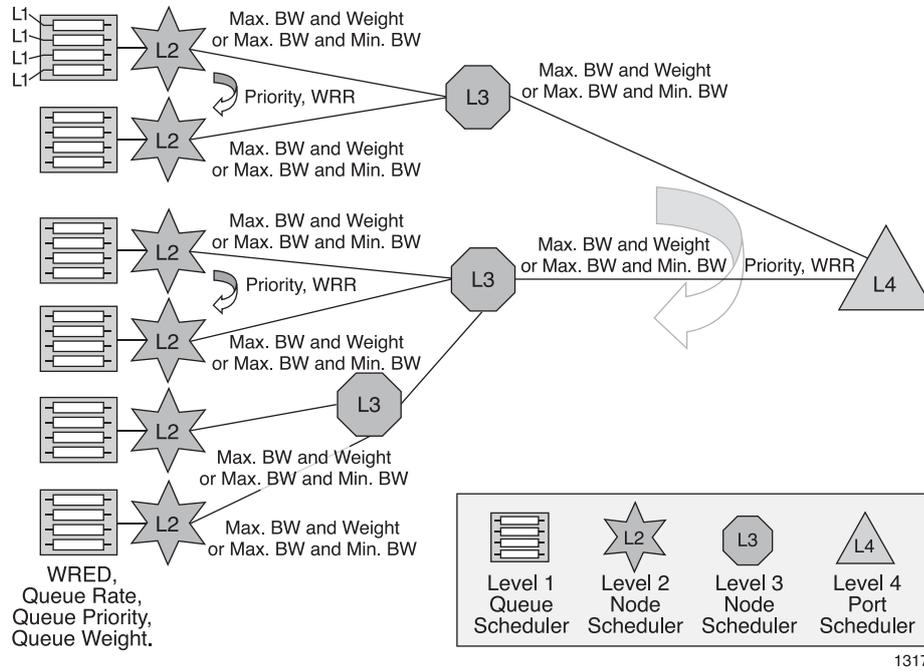


Figure 4 Scheduling Levels for Traditional TM

In the traditional TM hierarchy, an L4 node represents the port or link group. Hierarchical aggregation nodes, or L3 nodes, may be associated with 802.1Q PVCs or circuit-groups. L3 nodes attach to other L3 nodes (for example, 802.1Q PVC to 802.1Q tunnel), or directly to the L4 node. Circuits with queuing policies are represented by L2 nodes. L2 nodes attach to L3 nodes (for example, subscriber into 802.1Q PVC), or directly to the L4 node.

Figure 6 is a representation of a typical access network and one possible way that it might be modeled by the TM hierarchy in the SmartEdge router.

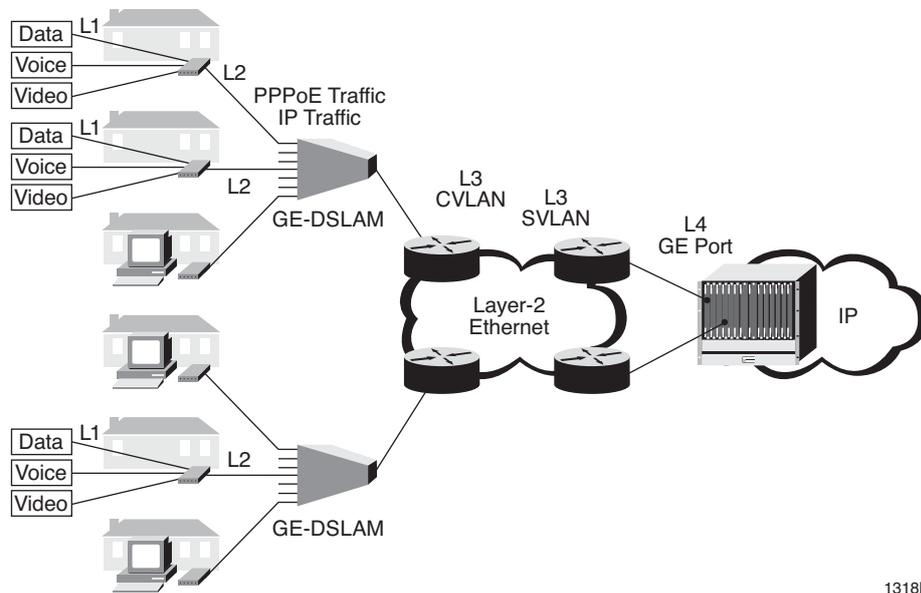


Figure 5 Typical Access Network

Table 5 maps the network elements in Figure 6 to the hierarchical scheduling levels in Figure 5 to show at which point in the network the different levels of scheduling are applied.

Table 5 One Possible Mapping of Access Network Points to Hierarchical Scheduling Levels in Traditional TM

Network Label	Description	Maps to Hierarchical Scheduling Level	Notes
Data Voice Video	Different classes of traffic from subscriber.	L1 Node	A TM queue priority group based on the queue assignment of the packet. Traffic on each queue is assigned to a single priority group (0 - 7). A rate configured on the priority group applies to all traffic carried by the queues assigned to the priority-group. A weight assigned to a queue affects the relative bandwidth that queue receives with respect to the other queues in the priority group.
PPPoE Traffic, IPoE Traffic	Each DSL line is modeled by a PPPoE or CLIPS session.	L2 Node	The queuing point. Each queuing point may offer one, two, four, or eight queues, which will carry the traffic on the circuit where the queuing policy binding is configured (the subscriber session, in this case) and any other circuits which inherit the queuing policy binding (not applicable in this case). Each packet to be transmitted is assigned to a queue based on its internal priority and the queue-map of the queuing policy. The maximum number of packets (queue depth) allowed in a queue is configurable.



CVLAN (inner VLAN of a double tagged VLAN) CVLAN is also known as an 802.1Q PVC.	The Layer 2 network segment that services a particular DSLAM is represented by an inner VLAN. TM parameters can be configured for this segment by configuring the inner VLAN to be a hierarchical scheduling node.	L3 Node	This hierarchical node serves as an aggregation of subordinate L2 nodes (for example, all the subscriber sessions encapsulated by the inner VLAN). A rate or weight configured on this node applies to all traffic carried by the subordinate nodes and their associated circuits.
SVLAN (outer VLAN of a double tagged VLAN) SVLAN is also known as 802.1Q tunnel.	The Layer 2 network segment or path that services a grouping of DSLAMs is represented by an outer VLAN. TM parameters can be configured for this segment by configuring the outer VLAN to be a hierarchical scheduling node.	L3 Node	This hierarchical node serves as an aggregation of subordinate L3 nodes (for example, all the inner VLANs encapsulated by the outer VLAN). A rate or weight configured on this node applies to all traffic carried by the subordinate nodes and their associated circuits.
GE Port	The physical port or link group.	L4 Node	A rate configured on the L4 node applies to all traffic carried by the port or link group.

A circuit may be associated with an L2 node, an L3 node, both, or neither. Rate controls and inter-node weights may be assigned at each node level. Strict priority scheduling is performed at all nodes in the hierarchical tree. Multiple levels of L3 nodes may be provisioned.

3.1.2 Properties of Scheduling Nodes

Each scheduling node has the following properties.

- **Parent:** the node in the scheduling tree that this node is subordinate to. All scheduling nodes other than the root node have a single parent. The root node represents the port or access link group where the traffic of all queues in the scheduling tree is destined for transmission.
- **Children:** any node in the scheduling tree other than L1 nodes may have one or more children that are subordinate to it.
- **Weight:** when making a scheduling selection between child nodes with the same parent, the TM scheduler uses strict priority (for example, the node with the highest priority packet available is selected). However, if the nodes have packets of equal priority, the scheduler uses weighted round-robin for selection. The `weight` or `qos weight` command is used to explicitly configure the node weight to use for this purpose; otherwise, a default rate is selected that is proportional to the configured maximum rate of the node. The weight of a node determines how much bandwidth it receives relative to its peers under periods of congestion.
- **Maximum rate:** this attribute enforces a maximum rate limit for all the traffic scheduled through a node. If the maximum rate of the node is exceeded, the node will not be able to transmit any more traffic until some time has passed and the node has accumulated additional send credits.



- **Minimum rate:** this attribute suggests a minimum rate target for all the traffic scheduled through a node. Until the minimum rate of the node is exceeded, its traffic will be preferred over any traffic that already received its configured minimum rate or weight allowance.

From a configuration standpoint, weight and minimum rate may be mutually exclusive in some contexts. When configuring the `qos priority-group rate` command, if the `exceed` keyword is specified, it is treated as a minimum rate; otherwise, it is treated as a maximum rate.

3.1.3 Defining a TM Scheduling Tree

By default, all the traffic forwarded through a port or access link group is scheduled through a single default egress queue and receives undifferentiated treatment. The traffic in the port can be scheduled and prioritized through multiple queues by configuring a PWFQ policy and applying it to the port or link group. Additionally or instead, a PWFQ policy can be applied to individual circuits under the port or link group. Each circuit with a PWFQ policy binding is allocated a unique set of queues and constitutes an L2 scheduling node. Each L2 node consists of 1, 2, 4, or 8 queues each assigned to one of 8 TM queue priority groups.

By default, an L2 node created under a port or access link group is created as a child of the port or link group L4 node.

3.1.3.1 Hierarchical TM Scheduling

Additional intermediate scheduling nodes, known as L3 nodes, can be configured to be part of a port or link group TM scheduling tree. Such L3 nodes provide a way to group multiple L2 nodes together for one or both of two possible purposes:

- Configure a strict rate limit (in other words maximum rate) to be enforced by the TM scheduler to be collectively enforced on all the traffic subject to the node (in other words, all the queues of all subordinate L2 nodes, as well as any subordinate L3 nodes and their subordinate L2 nodes).
- Configure the relative weight to be used under congestion conditions when traffic of the same priority is available for transmission under children of a common parent.

You can create an L3 scheduling node by configuring one or more of the following commands in the configuration context of the host circuit:

- `qos weight`
- `qos rate maximum`
- `qos rate minimum`
- `qos hierarchical mode strict`



The following circuit types support the above commands to create and host an L3 scheduling node:

- 802.1Q outer PVC: creates an L3 node that will be parented by the port or link group's L4 node and will be a parent to any L3 or L2 node created on the VLAN's children (for example, encapsulated subscriber sessions or inner PVCs, if this is an 802.1Q tunnel)
- 802.1Q inner PVC: creates an L3 node that will be parented by applicable 802.1Q outer PVC, if it is configured as an L3 node, or port or link group's L4 node otherwise. This L3 node will be a parent to any L2 node created on the PVC's children (for example, encapsulated subscriber sessions).
- Home circuit group: creates an L3 node that will be parented by the port or link group's L4 node and will be a parent to any L3 or L2 node created on the circuit-groups members and their children.

You can build a hierarchical TM scheduling tree in the following ways:

- Follow the natural circuit hierarchy of port > 802.1Q Tunnel > 802.1Q PVC > subscriber (as in the example of figure 6).
- Create an arbitrarily organized hierarchical TM scheduling tree using circuit groups

port > parent circuit group > child circuit group > member

- Create a hierarchy leveraging both circuit groups and the natural hierarchy. For example, port > circuit group > member PVC > subscriber. For example, if you have two 802.1Q PVCs each bound to a PWFQ policy and are members of a circuit group **cg350** with `qos rate max 350` configured, the L2 node of each PVC will be parented by the L3 node of the circuit group, and the total traffic scheduled for both PVCs will be limited to 350 kbps.

Note: L3 node configuration is only applicable and meaningful to TM scheduling and to subordinate L2 nodes. Therefore, if you configure a `qos rate maximum` command on a host circuit that does not have any applicable subordinate circuits with PWFQ bindings, the command will have no impact on traffic forwarding and QoS enforcement.

3.1.4 TM Scheduling Operation

When determining the next packet to be transmitted from a port or link group, the TM scheduler walks the scheduling tree downward from the root looking for queues that meet the following criteria:

- Has at least one packet awaiting transmission
- Is eligible for transmission according to all of its applicable parent scheduling nodes



- Is among the highest priority eligible queues with packets awaiting transmission. Because TM scheduling uses strict priority, only the highest priority eligible queues are considered until their contents have all been emptied and transmitted, or a parent node of the queue loses eligibility, then the next highest priority queues are considered. Under TM scheduling, the scheduling priority of a queue is determined by its priority-group assignment, with the lowest number priority-group (for example, zero) receiving the highest priority.

When more than one eligible highest-priority-available queue is identified, the queue to use to transmit the next packet is determined by a weighted round-robin algorithm that takes into account relative weights of applicable scheduling nodes and which queue recently had an opportunity to transmit packets.

Determining the eligibility of a scheduling node involves answering the following questions:

- Has the L4 node representing the port or link group exceeded its configured maximum rate? If so, none of its subordinate L2 and L3 nodes are eligible for scheduling until more send credits have been accumulated.
- Has an intermediate L3 node exceeded its configured maximum rate? If so, none of its subordinate L2 nodes are eligible for scheduling until more send credits have been accumulated.
- Has an L2 node exceeded its configured maximum rate? If so, none of its subordinate L1 nodes (for example, TM queue priority groups) are eligible for scheduling until more send credits have been accumulated.
- Has an L1 node (for example, priority group) exceeded its configured maximum rate? If so, its subordinate queues are not eligible for scheduling until more send credits have been accumulated.

Deciding between nodes with available packets of equal priority involves the following:

- If there is more than one child L2 or L3 node with eligible highest-available-priority queues awaiting transmission under the same L4 or L3 parent, which node to use is determined by a weighted-round-robin algorithm, with the relative weight of an L2 node determined by the **weight** command configured in the applicable PWFQ policy, and the relative weight of an L3 node determined by the **qos weight** command under the applicable host PVC or circuit group.
- If there is more than one TM queue priority group (in other words, L1 node) under an eligible L2 node with eligible highest-available-priority queues awaiting transmission, which queue to use is determined by a weighted-round-robin algorithm, with the relative weight of an L1 node determined by the **weight** parameter of the **queue priority** command configured in the applicable PWFQ policy.



3.1.5 TM Scheduling Summary

You can specify hierarchical scheduling nodes at various levels (port, 802.1Q tunnel, 802.1Q PVC, subscriber circuit, circuit group) on a traffic-managed port or link group. A level that does not have hierarchical scheduling specified inherits the scheduling specified at the next higher level. For example, a circuit with a PWFQ policy creates an L2 node parented to the closest L3 or L4 node configured in the circuit hierarchy above it. A circuit without its own PWFQ policy inherits the queues of and is subject to the properties of the closest L2 node configured in the circuit hierarchy above it. The circuit hierarchy may be determined by natural inheritance or circuit-group membership, or both.

Different levels in the hierarchical scheduling within traditional TM use different scheduling algorithms:

- Intra-queue (between queues in same priority group)—weighted round robin (WRR)
- Intra-priority group scheduler (between priority groups in L2 node)—strict priority
- Intra-L2 scheduler (between L2 nodes)—strict priority and WRR among L2 at same priority
- Intra-L3 scheduler (between L3 nodes)—strict priority and WRR among L3 at same priority

For more information about the strict priority and WRR scheduling modes, see [Priority Weighted Fair Queuing Policies](#).

3.2 Hierarchical Scheduling in Virtual-port TM

In virtual port TM, high-speed port traffic is partitioned into multiple lower bandwidth scheduling domains using virtual-port circuit-groups. Virtual port TM is currently only supported on a 10 GE (4-port) card. Each physical port is divided into virtual ports with a maximum of 10 virtual ports per port. Each virtual port forms the top of a TM scheduling tree as a virtual-port scheduling node and is capable of scheduling up to 1Gbps of traffic for a total of 10 Gbps of TM scheduled line-rate traffic. Traffic within each virtual port is scheduled independently of each other.

Hierarchical scheduling in virtual-port TM is a hybrid of MDRR and PWFQ scheduling. Multiple virtual-port scheduling nodes are attached to the port (L4 node). Logically, the virtual port is a level of aggregation below the port. You assign L3 and L2 nodes to virtual ports instead of a physical port. Nodes that attach to a virtual port in virtual-port TM are called top-most nodes. Since children follow their parent, the top-most node determines the virtual-port assignment of all the nodes below it. A topmost node may be an L3 or L2 node. See Figure 6. PWFQ is used to schedule the traffic within each virtual port.

The default port queues for 10 GE ports only support MDRR. This ensures that a 10-GE wire speed is achievable using the default port queues. On each physical port, the output from the virtual ports is combined and scheduled by the MDRR scheduler before egressing the port. Table 6 shows the fixed mapping that the SmartEdge router uses between the 8 PWFQ priorities and 2 MDRR priorities.

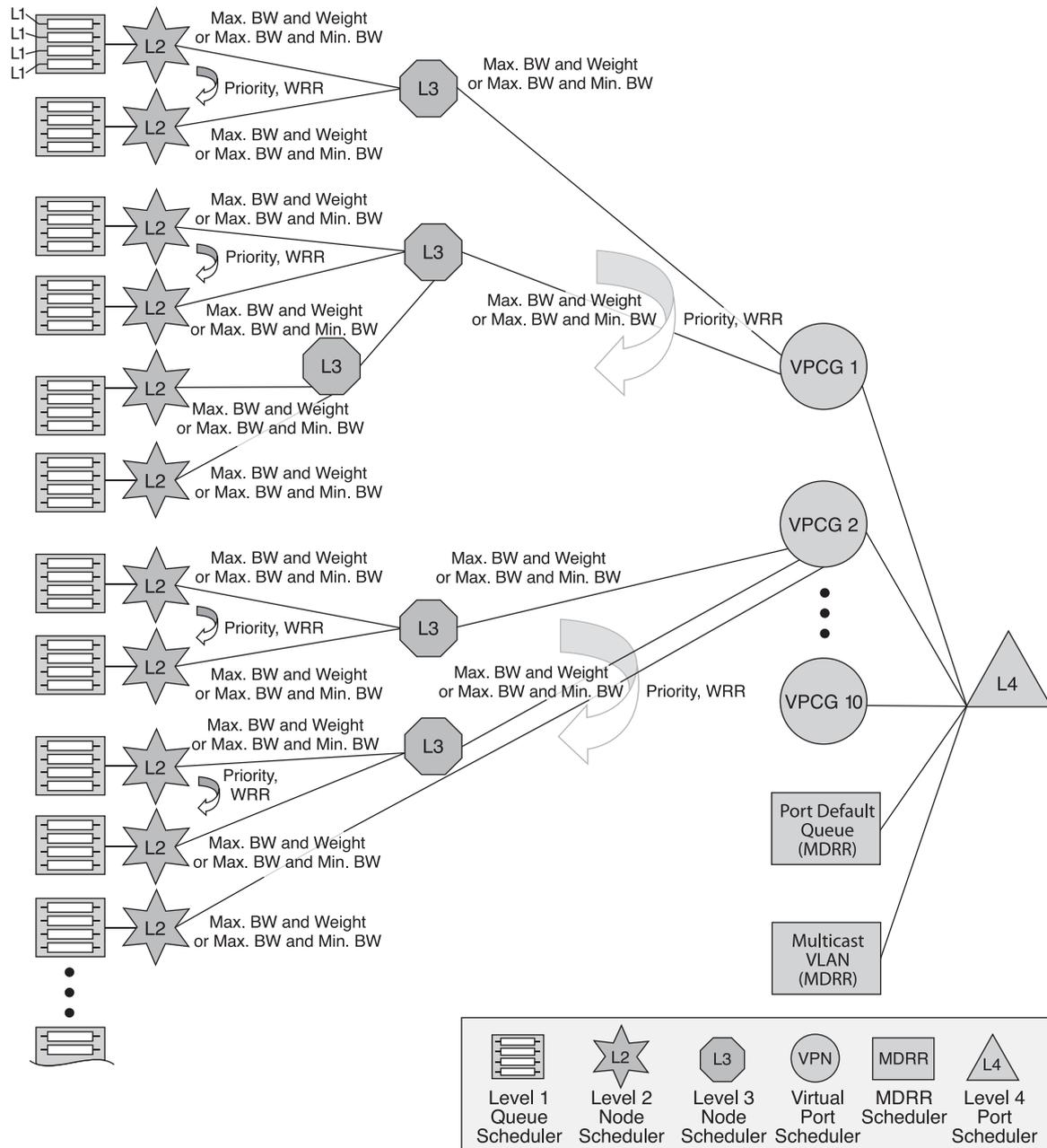


Figure 6 Scheduling Levels for virtual-port TM

1319

**Table 6 Mapping Between 8 PWFQ Priorities to MDRR Priorities**

PWFQ Priority	MDRR Priority
P0 to P1	Real Time (RT). This is high priority.
P2 to P7	non RT. This is low priority.

Additionally, virtual-port TM supports a limited number of circuits with MDRR bindings for multicast VLANs.

Enabling virtual-port TM configuration on any circuit under a port of a 10 GE (4-port) card requires that the affected circuit resides in virtual port circuit group (VPCG) by assigning the circuit to the VPCG. Assigning circuits to a VPCG is accomplished through either explicit or automatic VPCG assignment. For more information about VPCGs, see *Virtual Port Circuit Groups*.

3.3 Port Grouping for Traffic Scheduling

You can assign the ports of a traffic card that supports TM into different groups to customize the performance of traffic scheduling. These groups are referred to as scheduling port groups or simply port groups.

The ports within a port group share scheduling capacity within the group. For example, if one port is transmitting large packets and another is transmitting small packets, the port transmitting small packets, which requires more scheduling processing, can borrow capacity that is not needed by the port transmitting larger packets.

Port grouping allows you to manage the balance between scheduling performance and forwarding performance on a card. Each port group in use consumes processing capacity that would otherwise be available for packet forwarding. Defining more port groups results in higher scheduling performance but lower forwarding performance.

Each port map defined must be associated with a particular card type, and can only be referenced by cards of that type. Each port of the card must be assigned to one and only one port group.

The following list shows an example port map with five port groups for a GE 10-port card; each port group maps to two ports:

- Port group 1—ports 1 and 6
- Port group 2—ports 2 and 7
- Port group 3—ports 3 and 8
- Port group 4—ports 4 and 9
- Port group 5—ports 5 and 10



The SmartEdge router supports the following types of port group maps:

- **Predefined**—Each supported card type has a set of predefined port group maps, which reflect the most common deployment scenarios and optimize either forwarding or scheduling performance. Predefined maps assume you use ports in sequential order starting with port 1. For example, if you use five ports, use ports 1 to 5 and not ports 3 to 7 or random ports (for example, 2, 4, 6, 7, 8). If sequential ports are not possible, then a user-defined map should be applied instead of a predefined map.
- **Default**—One of the predefined port group maps for a supported card type is the default port group map, which the SmartEdge router uses when no port group map is configured.
- **User-defined**—Some cards may have multiple common deployment scenarios that require different port grouping. For example, in Protection groups, when two ports are not expected to be passing traffic at the same time, such as an active-and-standby configuration, the ports can be mapped to the same port group for maximum TM performance.

You can configure a user-defined port group map by using the `qos port map` command in global configuration mode and then applying the map to a card.

A port group map that is currently referenced by one or more cards may not be modified. You must remove all card configuration references to a particular port map before modifying it.

A configuration of a card can be modified to reference a different port-map or revert to the default port-map. However, such a port-map change is applied immediately only if that card is unlocked. A card is considered to be locked for port group map purposes if any PWFQ or other TM configuration is currently applied to any of the ports of the card. If the card is locked, then it must be reinitialized by using the `reload card` command for the port map change to take effect. The `show qos port-map bind` command can be used to determine whether a card is currently locked or unlocked for this purpose.

Warning!

Using the `reload card` command results in the temporary loss of all traffic carried by the port.

The SmartEdge router supports a maximum of eight port groups per card and a maximum of 64 ports for each card. The actual number of port groups and ports supported on a given card depends on the card type. For a list of the cards that support TM, see Section 1 on page 1.

Note: The 10 GE (4-port) card does not support port groups.



3.4 Overhead Profiles

Overhead profiles enable the SmartEdge router to take the encapsulation overhead of the access line into consideration so that the rate of traffic does not exceed the permitted traffic rate on the line. This downstream traffic shaping is controlled by QoS overhead profiles.

The overhead profile works in conjunction with a PWFQ policy. The PWFQ defines the rate of traffic flow; the overhead profile defines the encapsulation overhead and the available bandwidth on the access line. The rate can come from one of the following sources:

- Defined in a PWFQ policy
- Defined by the `rate circuit` command
- Received from a Remote Authentication Dial-In User Service (RADIUS) vendor-specific attribute (VSA)
- Received from an Access Node Control Protocol (ANCP) configuration
- Received from the Point-to-Point Protocol over Ethernet (PPPoE) tag, which also contains the line rate of the digital subscriber line-access multiplexer (DSLAM) and the encapsulation of the access line



4 Configuration and Operations Tasks

To configure scheduling policies, perform the tasks described in the following sections.

Note: In this section, the command syntax in the task tables displays only the root command; for the complete command syntax, see *Command List*.

4.1 Configuring a Queue Map

The SmartEdge router assigns a factory preset, or default, mapping of PD QoS priorities to queues, according to the number of queues configured. You can customize this mapping for the circuits to which any QoS scheduling policy is attached. To configure a queue map, perform the tasks in Table 7.

Table 7 Configure a Queue Map

Task	Root Command	Notes
Create or select a queue map and access queue map configuration mode.	<i>qos queue-map</i>	Enter this command in global configuration mode.
Specify the number of queues for the queue map and access num-queues configuration mode. ⁽¹⁾	<i>num-queues</i>	Enter this command in queue map configuration mode.
Customize the mapping of PD QoS priorities to queues.	<i>queue priority</i>	Enter this command in num-queues configuration mode.

(1) For information about the correlation between the number of ATMWFQ queues configured on a particular traffic card type and the corresponding number of PVCs allowed (per port and per traffic card), see *Configuring Circuits*.

4.2 Configuring a Congestion Avoidance Map

By default, the SmartEdge router drops packets at the end of the queue when the number of packets exceeds the configured maximum depth of the queue. A congestion avoidance map, when attached to an ATMWFQ, MDRR, or PWFQ scheduling policy, provides congestion management behavior for each queue defined by the policy.

To configure a congestion avoidance map, perform the tasks described in Table 8; enter all commands in congestion map configuration mode, unless otherwise noted.



Table 8 Configure a Congestion Avoidance Map

Task	Root Command	Notes
Create or select a congestion avoidance map and access congestion map configuration mode.	<i>qos congestion-avoidance-map</i>	Enter this command in global configuration mode.
Set the RED parameters for each queue in the map.	<i>queue red</i>	Perform this task for each queue in the map.
Set the exponential-weight for each queue in the map.	<i>queue exponential-weight</i>	Enter this command for each queue in the map.
Specify the depth of a queue.	<i>queue depth</i>	This command applies only to congestion avoidance maps for PWFQ policies only. Enter this command for each queue in the map.

4.3 Configuring an ATMWFQ Policy

You can configure an ATMWFQ policy with either RED or EPD parameters. To configure an ATMWFQ policy with RED parameters, using a congestion avoidance map, perform the tasks described in Table 9; enter all commands in ATMWFQ policy configuration mode, unless otherwise noted.

Table 9 Configure an ATMWFQ Policy with RED Parameters

Task	Root Command	Notes
Create the policy name and access ATMWFQ policy configuration mode.	<i>qos policy atmwfq</i>	Enter this command in global configuration mode.
Optional. Configure the policy with any or all of the following tasks:		
Assign a queue map to the policy.	<i>queue-map</i>	
Specify the number of queues for the policy. ⁽¹⁾	<i>num-queues</i> <i>Configuring Circuits for QoS</i>	By default, the number of queues is 4.
Assign a congestion avoidance map to the policy.	<i>congestion-map</i>	By default, no congestion map is assigned.



Table 9 Configure an ATMWFQ Policy with RED Parameters

Task	Root Command	Notes
Define the algorithm for queue 0.	<i>queue 0 mode</i>	By default, the queue mode is alternate.
Specify the traffic weight for each queue.	<i>queue weight</i>	By default, the weight is 2.

(1) For information about the correlation between the number of queues and the number of VCs, see *Configuring Circuits*.

To configure an ATMWFQ policy with EPD parameters, perform the tasks described in Table 10; enter all commands in ATMWFQ policy configuration mode, unless otherwise noted.

Table 10 Configure an ATMWFQ Policy with EPD Parameters

Task	Root Command	Notes
Create the policy name and access ATMWFQ policy configuration mode.	<i>qos policy atmwfq</i>	Enter this command in global configuration mode.
Configure the policy with any or all of the following tasks:		
Assign a queue map to the policy.	<i>queue-map</i>	
Specify the number of queues for the policy. ⁽¹⁾	<i>num-queues</i>	By default, the number of queues is 4.
Modify congestion parameters for each queue.	<i>queue congestion epd</i>	
Define the algorithm for queue 0.	<i>queue 0 mode</i>	By default, the queue mode is alternate.
Specify the traffic weight for each queue.	<i>queue weight</i>	By default, the weight is 2.

(1) For information about the correlation between the number of queues and the number of VCs, see *Configuring Circuits*.

4.4 Configuring an MDRR Policy

To configure an MDRR policy, perform the tasks described in Table 11; enter all commands in MDRR policy configuration mode, unless otherwise noted.



Table 11 Configure an MDRR Policy

Task	Root Command	Notes
Create the policy name and access MDRR policy configuration mode.	<i>qos policy mdr</i>	Enter this command in global configuration mode.
Optional. Configure the policy by completing any or all of the following tasks:		
Assign a queue map to the policy.	<i>queue-map</i>	
Specify the number of queues for the policy.	<i>num-queues</i>	By default, the number of queues is 8.
Assign a congestion avoidance map to the policy.	<i>congestion-map</i>	
Specify the scheduling algorithm.	<i>qos mode (MDRR)</i>	By default, the mode is normal.
Specify the traffic weight per queue.	<i>queue weight</i>	By default, the traffic weight is 0.
Set a rate limit for the policy.	<i>rate</i>	By default, there is no rate limit.

4.5 Configuring a PWFQ Policy

To configure a PWFQ policy, perform the tasks described in Table 12; enter all commands in PWFQ policy configuration mode, unless otherwise noted.

Table 12 Configure a PWFQ Policy

Task	Root Command	Notes
Create the policy name and access PWFQ policy configuration mode.	<i>qos policy pwfq</i>	Enter this command in global configuration mode.
Optional. Configure the policy with any or all of the following tasks:		
Assign a queue map to the policy.	<i>queue-map</i>	
Specify the number of queues for the policy.	<i>num-queues</i>	By default, the number of queues is 8.
Assign a congestion avoidance map to the policy.	<i>congestion-map</i>	



Table 12 Configure a PWFQ Policy

Task	Root Command	Notes
Assign a priority and relative weight to each queue.	<i>queue priority</i>	Enter this command for each queue that you specified with the num-queues command.
Set the maximum and minimum rates for the policy.	<i>rate</i>	You must enter this command to specify the maximum rate; the minimum rate is optional. You cannot set a minimum rate if you also assign a relative weight to this policy.
Assign a relative weight to this policy.	<i>weight</i>	You cannot assign a relative weight if you also set a minimum rate for this policy.
Set the rate for each PD QoS priority group.	<i>queue priority-group</i>	Enter this command for each PD QoS priority group.

4.6 Configuring an Overhead Profile

An overhead profile defines the encapsulation overhead and the available bandwidth on the access line. To configure an overhead profile, perform the tasks described in Table 13; enter all commands in overhead profile configuration mode, unless otherwise noted.

Note: For the overhead policy to take effect, you must also configure the *qos policy queuing* command.

Table 13 Configure an Overhead Profile

Task	Root Command	Notes
Create or select a QoS overhead profile.	<i>qos profile overhead (global)</i>	
Create a default rate-factor for the overhead profile.	<i>rate-factor</i>	
Create a default encapsulation access-line type for the overhead profile.	<i>encaps-access-line</i>	
Create a default number of reserved bytes, per packet.	<i>reserved</i>	



Table 13 Configure an Overhead Profile

Task	Root Command	Notes
Configure overhead parameters for the specified DSL data type in the overhead profile.	<i>type (DSL)</i>	
Define the percentage of bandwidth that is unavailable to traffic on the circuit, port, or subscriber record to which the QoS policy is attached to the overhead profile for a specific access-line type in the overhead profile.	<i>rate-factor</i>	Enter this command in overhead type configuration mode.

4.7 Configuring a User-Defined Port Group Map and Applying It to a Card

To configure a user-defined port group map and then apply it to a traffic card that supports port groups, perform the tasks described in Table 14; enter all commands in the specified configuration mode.

Table 14 Configure a User-Defined Port Group Map and Apply It to a Card

Task	Root Command	Notes
Define the name of a port group map for a specified traffic card and enter port group map configuration mode.	<i>qos port-map (global)</i>	Enter this command in global configuration mode.
Define a port group.	<i>group</i>	Enter this command in port group map configuration mode.
Apply the port group map you defined to the card you are configuring.	<i>qos port-map (card)</i>	Enter this command in card configuration mode. Specify the name of the user-defined port group map to apply to the card. The name is displayed as an option. The application of the port group map takes effect after a card reload.



4.8 Applying a Predefined or Default Port Group Map to a Card

To apply a predefined, or default port group map to a traffic card that supports port groups, perform the task described in Table 15; enter the command in the specified configuration mode.

Table 15 Apply a Predefined or Default Port Group Map to a Card

Task	Command	Notes
Apply a port group map to the card you are configuring.	<i>qos port-map (card)</i>	Enter this command in card configuration mode. Specify the name of the predefined or default port group map to apply to the card. The application of the port group map takes effect after a card reload. If no specified port group map is applied, the default port group map is applied.

4.9 Configuring VPCGs

This section describes tasks related to configuring VPCGs.

Note: Configuration examples provided in this section are only supported on a 10 Gigabit Ethernet (4-port) card.

4.9.1 Create Port-Based VPCG and Assign a Circuit Membership to VPCG

To configure a port-based virtual port circuit group (VPCG) and assign a circuit membership to the VPCG, perform the tasks described in Table 16; enter the commands in the specified configuration modes.

Table 16 Configure a Port-Based VPCG and Assign a Circuit Membership to VPCG

Step	Task	Root Command	Notes
1.	Configure a PWFQ policy.	<i>qos policy pwfq</i>	Enter this command in global configuration mode. See Table 12 for details on how to configure a PWFQ policy.
2.	Define a VPCG and specify a port on which all circuits in this circuit group reside.	<i>circuit group</i>	Enter this command in global configuration mode. Use the virtual-port keyword with this command to specify that this circuit group is a virtual port circuit group.



Table 16 Configure a Port-Based VPCG and Assign a Circuit Membership to VPCG

Step	Task	Root Command	Notes
3.	Optional. Attach PWFQ scheduling to VPCG and its members.	<i>qos policy queuing</i>	Enter this command in circuit-group configuration mode. Instead of attaching the PWFQ scheduling policy to the circuit group, you can also attach it to the 802.1Q PVC or subscriber circuit (using the default subscriber profile, a named subscriber profile, or an individual subscriber record).
4.	Select an Ethernet port in which the members of circuit group are to reside, and access port configuration mode.	<i>port ethernet</i>	Enter this command in global configuration mode.
5.	Create an 802.1Q PVC and enter the PVC configuration mode.	<i>dot1q pvc</i>	Enter this command in port configuration mode.
6.	Specify that the circuit is a member of the VPCG.	<i>circuit-group-member</i>	Enter this command in dot1q PVC configuration mode.

4.9.2 Creating a Link-Group Based VPCG and Assign a Circuit Membership to VPCG

To create a link group based VPCG and assign a circuit membership to the VPCG, perform the tasks described in Table 17; enter the commands in the specified configuration modes.

Table 17 Configure a Link Group-Based VPCG and Assign a Circuit Membership to VPCG

Step	Task	Root Command	Notes
1.	Configure a PWFQ policy.	<i>qos policy pwfq</i>	Enter this command in global configuration mode. See Table 12 for details on how to configure a PWFQ policy.
2.	Define a VPCG and specify a link group on which all circuits in this circuit group reside.	<i>circuit group</i>	Enter this command in global configuration mode. Enter this command in global configuration mode. Use the virtual-port keyword with this command to specify that this circuit group is a virtual port circuit group.



Table 17 Configure a Link Group-Based VPCG and Assign a Circuit Membership to VPCG

Step	Task	Root Command	Notes
3.	Optional. Attach PWFQ scheduling to VPCG and its members.	<i>qos policy queuing</i>	Enter this command in circuit-group configuration mode. Instead of attaching the PWFQ scheduling policy to the circuit group, you can attach it to the 802.1Q PVC.
4.	Create an empty link group and access link group configuration mode.	<i>link-group (Global, DS-1, E1, Port Configuration Modes)</i>	Enter this command in global configuration mode. Specify the access keyword for an access link group.
5.	Enable the link group to use PWFQ scheduling on a virtual port.	<i>qos pwfq scheduling</i>	Enter this command in link group configuration mode. Specify virtual-port keyword for virtual port PWFQ scheduling mode.
6.	Create an 802.1Q PVC (in the link group) and enter the PVC configuration mode.	<i>dot1q pvc</i>	Enter this command in link group configuration mode.
7.	Specify that the circuit is a member of the VPCG.	<i>circuit-group-member</i>	Enter this command in link PVC configuration mode.
8.	Apply the link group to a port.	<i>port ethernet</i>	Only applies to a port in a 10 Gigabit Ethernet (4-port) card.

4.9.3 Creating a Port-Based VPCG and Assign Subscriber Membership to VPCG

To configure a port-based VPCG and assign subscribers membership to the VPCG, perform the tasks described in Table 18; enter the commands in the specified configuration modes.

Table 18 Configure a Port-Based VPCG and Assign Circuit-Membership to Subscriber

Step	Task	Root Command	Notes
1.	Configure a PWFQ policy.	<i>qos policy pwfq</i>	Enter this command in global configuration mode. See Table 12 for details on how to configure a PWFQ policy.



Table 18 Configure a Port-Based VPCG and Assign Circuit-Membership to Subscriber

Step	Task	Root Command	Notes
2.	Define a VPCG and specify a port on which all circuits in this circuit group reside.	<i>circuit group</i>	Enter this command in global configuration mode. Use the virtual-port keyword with this command to specify that this circuit group is a virtual port circuit group.
3.	Optional. Attach PWFQ scheduling to VPCG and its members.	<i>qos policy queuing</i>	Enter this command in circuit-group configuration mode. Instead of attaching the PWFQ scheduling policy to the circuit group, you can attach it to the subscriber circuit (using the default subscriber profile, a named subscriber profile, or an individual subscriber record)
4.	Create a default subscriber profile, a named subscriber profile, or an individual subscriber record, and access subscriber configuration mode.	<i>subscriber</i>	Enter this command in context configuration mode. To create a default subscriber profile, use the default keyword with this command. To create a named subscriber profile, use the profile prof-name construct with this command. To create an individual named subscriber record, use the name sub-name construct with this command.
5.	Specify that the subscriber (default subscriber profile, named subscriber profile, or individual subscriber record) is a member of the VPCG.	<i>circuit-group-member</i>	Enter this command in subscriber configuration mode.

4.10 Operations Tasks

To monitor and administer QoS scheduling features, perform the appropriate tasks described in Table 19. Enter the **debug** command in exec mode; enter the **show** commands in any mode.



Table 19 Monitor and Administer QoS Features

Task	Command
Display the queue assignments for a QoS congestion avoidance map.	<i>show qos congestion-map</i>
Display information about one or more QoS ATMWFQ policies.	<i>show qos policy atmwfq</i>
Display information about one or more QoS PWFQ policies.	<i>show qos policy pwfq</i>
Display information about one or more configured QoS queue maps.	<i>show qos queue-map</i>
Display information about a specific QoS port group map or all QoS port group maps, for a specific or all traffic card types that support port groups.	<i>show qos port-map</i>
Display information about the QoS port group map binding for a traffic card in a specific slot or for all configured traffic cards that support port groups.	<i>show qos port-map bind</i>





5 Configuration Examples

The following sections provide examples of QoS scheduling configurations.

5.1 Queue Maps

The following example creates three queue maps and assigns a custom mapping of PD QoS priority groups to queues, based on the number of queues configured:

```
[local]Redback(config)#qos queue-map Custom2
[local]Redback(config-queue-map)#num-queues 2
[local]Redback(config-num-queues)#queue 0 priority 0
[local]Redback(config-num-queues)#queue 1 priority 1 2 3 4 5 6 7
[local]Redback(config-num-queues)#exit

[local]Redback(config)#qos queue-map Custom4
[local]Redback(config-queue-map)#num-queues 4
[local]Redback(config-num-queues)#queue 0 priority 0
[local]Redback(config-num-queues)#queue 1 priority 1 2
[local]Redback(config-num-queues)#queue 2 priority 3 4 5 6
[local]Redback(config-num-queues)#queue 3 priority 7
[local]Redback(config-num-queues)#exit

[local]Redback(config)#qos queue-map Custom8
[local]Redback(config-queue-map)#num-queues 8
[local]Redback(config-num-queues)#queue 0 priority 0
[local]Redback(config-num-queues)#queue 1 priority 1
[local]Redback(config-num-queues)#queue 2 priority 2
[local]Redback(config-num-queues)#queue 3 priority 3
[local]Redback(config-num-queues)#queue 4 priority 4
[local]Redback(config-num-queues)#queue 5 priority 5
[local]Redback(config-num-queues)#queue 6 priority 6
[local]Redback(config-num-queues)#queue 7 priority 7
[local]Redback(config-num-queues)#exit
```

5.2 Congestion Avoidance Map for Multidrop Profiles

The following example shows the configuration of the congestion avoidance map, **map-red4a**, with two profiles for any ATMWFQ policy:

```
[local]Redback(config)#qos congestion-avoidance-map map-red4a atmwfq
[local]Redback(config-congestion-map)#queue 0 exponential-weight 40
[local]Redback(config-congestion-map)#queue 0 red default min-thres 30
max-thres 5200 probability 16
[local]Redback(config-congestion-map)#queue 0 red profile-1 dscp cs7 min-thres 140
max-thres 13000 probability 34
[local]Redback(config-congestion-map)#queue 0 red profile-2 dscp cs3 min-thres 230
max-thres 15600 probability 50
[local]Redback(config-congestion-map)#queue 3 exponential-weight 13
[local]Redback(config-congestion-map)#queue 3 red default max-thresh 5200
[local]Redback(config-congestion-map)#queue 3 red profile-1 dscp af21 min-thres 100
max-thres 14000 prob 450
```



5.3 ATMWFQ Policies

The following example shows how to configure the ATMWFQ policy, **example2**, with the **map-red4a** congestion avoidance map:

```
[local] Redback (config) #qos policy example2 atmwfq
[local] Redback (config-policy-atmwfq) #num-queues 4
[local] Redback (config-policy-atmwfq) #congestion-map map-red4a
[local] Redback (config-policy-atmwfq) #queue 0 weight 10
[local] Redback (config-policy-atmwfq) #queue 1 weight 20
[local] Redback (config-policy-atmwfq) #queue 2 weight 30
[local] Redback (config-policy-atmwfq) #queue 3 weight 40
[local] Redback (config-policy-atmwfq) #qos 0 mode strict
[local] Redback (config-policy-atmwfq) #exit
```

The following example shows how to configure an ATMWFQ policy, **example3**, with EPD parameters:

```
[local] Redback (config) #qos policy example3 atmwfq
[local] Redback (config-policy-atmwfq) #num-queues 4
[local] Redback (config-policy-atmwfq) #queue 0 cong epd max-thres 5200
[local] Redback (config-policy-atmwfq) #queue 1 cong epd max-thres 5200
[local] Redback (config-policy-atmwfq) #queue 2 cong epd max-thres 5200
[local] Redback (config-policy-atmwfq) #qos 0 mode strict
[local] Redback (config-policy-atmwfq) #exit
```

5.4 MDRR Policy

The following example shows how to configure the MDRR policy, **example4**, using weighted round robin mode with **4** queues and divide the bandwidth between the queues according to an approximate 50:30:10:10 ratio during periods of congestion:

```
[local] Redback (config) #qos policy example4 mdr
[local] Redback (config-policy-mdrr) #qos mode wrr
[local] Redback (config-policy-mdrr) #num-queues 4
[local] Redback (config-policy-mdrr) #queue 0 weight 50
[local] Redback (config-policy-mdrr) #queue 1 weight 30
[local] Redback (config-policy-mdrr) #queue 2 weight 10
[local] Redback (config-policy-mdrr) #queue weight 10
[local] Redback (config-policy-mdrr) #exit
```

5.5 PWFQ Policies

The following examples provide configurations for types of priority scheduling:

In these examples, all policies are configured with four queues, a queue map, **qpm1**, a congestion avoidance map, **map-red4p**, and a maximum bandwidth of 50 Mbits (**50000**) for the policy; each of the four queues in the policy is assigned a priority and a relative weight, which specifies percentage of the available bandwidth within its TM queue priority group.



5.5.1 Strict Priority

The following example shows how to configure the **strict** PWFQ policy for strict priority scheduling. Each queue has a unique priority and the same relative weight:

```
[local]Redback(config)#qos policy strict pwfq
[local]Redback(config-policy-pwfq)#num-queues 4
[local]Redback(config-policy-pwfq)#queue-map qpmap1
[local]Redback(config-policy-pwfq)#congestion-map map-red4p
[local]Redback(config-policy-pwfq)#rate maximum 50000
[local]Redback(config-policy-pwfq)#queue 0 priority 0 weight 100
[local]Redback(config-policy-pwfq)#queue 1 priority 1 weight 100
[local]Redback(config-policy-pwfq)#queue 2 priority 2 weight 100
[local]Redback(config-policy-pwfq)#queue 3 priority 3 weight 100
[local]Redback(config-policy-pwfq)#exit
```

5.5.2 Normal Priority

The following example shows how to configure the **normal** PWFQ policy for normal priority scheduling. All queues have the same priority; scheduling is based on the relative weight assigned to each queue. In this example, queue 0 receives 50% of the available bandwidth (25 Mbits), queue 1 receives 30% (15 Mbits), queue 2 receives 20% (10 Mbits), and queue 3 receives 10% (5 Mbits):

```
[local]Redback(config)#qos policy normal pwfq
[local]Redback(config-policy-pwfq)#num-queues 4
[local]Redback(config-policy-pwfq)#queue-map qpmap1
[local]Redback(config-policy-pwfq)#congestion-map map-red4p
[local]Redback(config-policy-pwfq)#rate maximum 50000
[local]Redback(config-policy-pwfq)#queue 0 priority 0 weight 50
[local]Redback(config-policy-pwfq)#queue 1 priority 0 weight 30
[local]Redback(config-policy-pwfq)#queue 2 priority 0 weight 20
[local]Redback(config-policy-pwfq)#queue 3 priority 0 weight 10
[local]Redback(config-policy-pwfq)#exit
```

5.5.3 Strict + Normal Priority

The following example shows how to configure the PWFQ policy, **pwfq4** with two TM queue priority groups, **0** and **1**.

Queues 0 and 1 have the same priority (group 0) and will be serviced before queues 2 and 3 (assigned to group 1). Within each TM queue priority group the queues are serviced in round-robin order, according to their assigned relative weights. For example, queue 0 receives 70% and queue 1 receives 30% of the bandwidth available for the group. Queues 2 and 3 are serviced only when queues 0 and 1 are empty; queue 2 receives 60% and queue 3 receives 40% of the available bandwidth for the group:



```
[local] Redback (config) #qos policy pwfq4 pwfq
[local] Redback (config-policy-pwfq) #num-queues 4
[local] Redback (config-policy-pwfq) #queue-map qpmap1
[local] Redback (config-policy-pwfq) #congestion-map map-red4p
[local] Redback (config-policy-pwfq) #rate maximum 50000
[local] Redback (config-policy-pwfq) #queue 0 priority 0 weight 70
[local] Redback (config-policy-pwfq) #queue 1 priority 0 weight 30
[local] Redback (config-policy-pwfq) #queue 2 priority 1 weight 60
[local] Redback (config-policy-pwfq) #queue 3 priority 1 weight 40
[local] Redback (config-policy-pwfq) #exit
```

5.5.4 Strict + Normal Priority with Maximum Priority-Group Bandwidth

The following example shows how to configure the **pwfq4** policy as before, but adds a maximum bandwidth limitation for each TM queue priority group. In this case, the combined traffic in group 0 is limited to 10 Mbits (**10000**), even when there is no traffic on the queues in priority group 1. Similarly, combined traffic on queues 2 and 3 is limited to 1 Mbit (**1000**), even when there is no traffic on queues 0 and 1:

```
[local] Redback (config) #qos policy pwfq4 pwfq
[local] Redback (config-policy-pwfq) #num-queues 4
[local] Redback (config-policy-pwfq) #queue-map qpmap1
[local] Redback (config-policy-pwfq) #congestion-map map-red4p
[local] Redback (config-policy-pwfq) #rate maximum 50000
[local] Redback (config-policy-pwfq) #queue 0 priority 0 weight 70
[local] Redback (config-policy-pwfq) #queue 1 priority 0 weight 30
[local] Redback (config-policy-pwfq) #queue priority-group 0 rate 10000
[local] Redback (config-policy-pwfq) #queue 2 priority 1 weight 60
[local] Redback (config-policy-pwfq) #queue 3 priority 1 weight 40
[local] Redback (config-policy-pwfq) #queue priority-group 1 rate 1000
[local] Redback (config-policy-pwfq) #exit
```

5.5.5 Strict + Normal Priority with Maximum and Minimum Bandwidths

The following example shows how to configure the **pwfq4** policy as before, but adds a minimum bandwidth limitation of 10 Mbits (**10000**) for the policy. In this configuration, the minimum bandwidth is guaranteed to the policy only if the next higher level of scheduling (for example, for the scheduling policy applied towards an 802.1Q PVC) is in strict priority mode. If it is not, then the minimum bandwidth is ignored:

```
[local] Redback (config) #qos policy pwfq4 pwfq
[local] Redback (config-policy-pwfq) #num-queues 4
[local] Redback (config-policy-pwfq) #queue-map qpmap1
[local] Redback (config-policy-pwfq) #congestion-map map-red4p
[local] Redback (config-policy-pwfq) #rate maximum 50000
[local] Redback (config-policy-pwfq) #rate minimum 10000
[local] Redback (config-policy-pwfq) #queue 0 priority 0 weight 70
[local] Redback (config-policy-pwfq) #queue 1 priority 0 weight 30
[local] Redback (config-policy-pwfq) #queue priority-group 0 rate 10000
[local] Redback (config-policy-pwfq) #queue 2 priority 1 weight 60
[local] Redback (config-policy-pwfq) #queue 3 priority 1 weight 40
[local] Redback (config-policy-pwfq) #queue priority-group 1 rate 1000
[local] Redback (config-policy-pwfq) #exit
```



5.6 Overhead Profiles

The following example shows how to configure an overhead profile for **example1**, and set the default rate factor to **15**, a reserve value to **8**, and the encapsulation type to **pppoa-llc**. After you set the overhead profile with default values, you configure **ads11** and **vds11** with custom encapsulation and reserve values:

```
[local]Redback(config)#qos profile example1 overhead
[local]Redback(config-profile-overhead)#rate-factor 15
[local]Redback(config-profile-overhead)#encaps-access-line pppoa-llc
[local]Redback(config-profile-overhead)#reserved 8
[local]Redback(config-profile-overhead)#type ads11
[local]Redback(config-type-overhead)#rate-factor 20
[local]Redback(config-type-overhead)#encaps-access-line pppoa-null
[local]Redback(config-type-overhead)#reserved 16
[local]Redback(config-type-overhead)#exit
[local]Redback(config-profile-overhead)#type vds11
[local]Redback(config-type-overhead)#encaps-access-line pppoa-null value 22 data-link ethernet
[local]Redback(config-type-overhead)#reserved 10
```

5.7 QoS Port Group Maps

The following example shows how to specify a user-defined QoS port map named **abc** for a **ge3-4-portcard** and then enter port group map configuration mode. In this mode, the example shows how to define two port groups for the **ge3-4-port** card type with two ports mapped to one of the port groups. After defining the **abc** QoS port map, the example shows how to apply this port group map to the **ge3-4-port** card in **slot2**. Note that when you enter **qos port map ?** the card configuration mode in this case, you have the **abc** QoS port map as an option:

```
[local]Redback(config)#qos port-map abc card-type ge3-4-port
[local]Redback(config-port-group-map)#group 1 ports 1 2
[local]Redback(config-port-group-map)#group 2 ports 3 4
[local]Redback(config-port-group-map)#end
[local]Redback(config)#card ge3-4-port 2
[local]Redback(config-card)#qos port-map ?
abc User-defined
fwd_max_perf Predefined map optimized for forwarding performance
tm_max_perf Default map optimized for TM performance
[local]Redback(config-card)#qos port-map abc
Note: if the card is locked the changes will be applied to the card on its next reload
[local]Redback(config-card)#end
```

5.8 MDRR and PWFQ Coexistence

This example shows a configuration of MDRR and PWFQ coexistence with MDRR and PWFQ policies configured on different circuits within the same port.



```

subscriber default
  ip address pool
  qos policy queuing TMPOLICY1 <----PWFQ policy "TMPOLICY1" is applied to default subscriber.

port eth 6/1
no shut
qos policy queuing MDRRPOLICY1 <----MDRR policy "MDRRPOLICY1" is applied at the port.
encapsulation dot1q
  dot1q pvc 1 encapsulation lqtunnel
  dot1q pvc 1:1 encapsulation multi <----This PVC inherits the "MDRRPOLICY1"
  policy applied at the port level.
  bind interface 2 a
  circuit protocol pppoe <----Subscriber comes up on this child circuit
  bind authen pap chap context CONTEXT max 10 and the "TMPOLICY1" policy is applied to it.
!
!

```

This example shows another configuration of MDRR and PWFQ coexistence with MDRR and PWFQ policies configured on the different circuits within the same port.

```

subscriber name jaden
  password pass1
  ip address pool
  qos policy queuing TMPOLICY2 <----PWFQ policy "TMPOLICY2" is applied to subscriber record "jaden"
!
subscriber name jose
  password pass2
  ip address pool
  qos policy queuing MDRRPOLICY2 <----PWFQ policy "MDRRPOLICY2" is applied to subscriber record "jose"
!
subscriber name purvi
  password pass3
  ip address pool
  qos policy queuing TMPOLICY2 <----PWFQ policy "TMPOLICY2" is applied to subscriber record "purvi"
!
subscriber name sunny
  password pass4
  ip address pool
  qos policy queuing MDRRPOLICY2 <----PWFQ policy "MDRRPOLICY2" is applied to subscriber record "sunny"

port eth 6/1
no shut
qos policy queuing MDRRPOLICY2 <----MDRR policy "MDRRPOLICY2" is applied at the port
encapsulation dot1q
  dot1q pvc 1 encapsulation lqtunnel
  qos policy queuing TMPOLICY3 <----PWFQ policy "TMPOLICY3" is applied at the 802.1Q PVC tunnel
  dot1q pvc 1:1 encapsulation multi
  bind interface 2 a <----This PVC inherits the "TMPOLICY3" policy applied at the 802.1Q PVC tunnel
  circuit protocol pppoe
  bind authen pap chap context CONTEXT max 10 <----Subscriber comes up on this child circuit.
  Either "TMPOLICY2" or "MDRRPOLICY2" policy is
  applied here depending on which subscriber comes up.

```

5.9 Traffic Management

This section provides TM-related configurations.

Note: Traffic Management configuration examples provided in this section are only supported on a 10 Gigabit Ethernet (4-port) card.



5.9.1 Homed VPCG

The following example shows how to define a homed VPCG named `vp9` that resides in slot 1, port 1, and specify a dot1q PVC that is a member of `vp9`.

```
[local]Redback(config)#circuit-group vp9 port 1/1 virtual-port
.
.
[local]Redback(config)#port ethernet 1/1
[local]Redback(config-port)#encapsulation dot1q
[local]Redback(config-port)#dot1q pvc 1
[local]Redback(config-dot1q-pvc)#circuit-group-member vp9
[local]Redback(config-policy-pwfq??)#exit
```

5.9.2 Explicit Assignment of Circuit-Group Membership to VPCGs

This section provides an example of an explicit assignment of circuit-group membership of a subscriber circuit to a port-based VPCG. This example highlights the following configurations:

- Create and configure PWFQ policy (see #1 in example)
- Define port-based VPCGs (see #2 in example)
- Attach PWFQ scheduling to VPCG and its members (see #3 in example)
- Create named subscriber record and specify its circuit-group membership to the VPCG (see # 4 in example).
- Specify the circuit-group membership of the subscriber circuit to the VPCG (see # 5 in example). This shows an explicit assignment of circuit group membership.

Note: This configuration is only supported on a 10 Gigabit Ethernet (4-port) card.



```
qos policy tmpolicy1 pwfq <----#1 Create and configure PWFQ policy.
rate maximum 12000
queue 0 priority 0 weight 100
queue 1 priority 1 weight 100
queue 2 priority 2 weight 100
queue 3 priority 3 weight 100
queue 4 priority 4 weight 100
queue 5 priority 5 weight 100
queue 6 priority 6 weight 100
queue 7 priority 7 weight 100
queue priority-group 0 rate 5000
queue priority-group 1 rate 2500
queue priority-group 2 rate 1200
queue priority-group 3 rate 1000
queue priority-group 4 rate 800
queue priority-group 5 rate 600
queue priority-group 6 rate 500
queue priority-group 7 rate 400

qos policy tmpolicy2 pwfq
rate maximum 12000
queue 0 priority 0 weight 100
queue 1 priority 1 weight 100
queue 2 priority 2 weight 100
queue 3 priority 3 weight 100
queue 4 priority 4 weight 100
queue 5 priority 5 weight 100
queue 6 priority 6 weight 100
queue 7 priority 7 weight 100
queue priority-group 0 rate 2500
queue priority-group 1 rate 1250
queue priority-group 2 rate 600
queue priority-group 3 rate 500
queue priority-group 4 rate 400
queue priority-group 5 rate 300
queue priority-group 6 rate 250
queue priority-group 7 rate 200

circuit-group cg1 port 6/1 virtual-port <----#2 Define VPCG and specify a port on which
                                     all circuits in this circuit group resides.
qos policy queuing tmpolicy1 <----#3 Attach PWFQ scheduling to VPCG "cg1". You have the option to
                                     attach the policy at the subscriber record instead of the
                                     circuit group.

circuit-group cg2 port 6/1 virtual-port
qos policy queuing tmpolicy2
!
subscriber name sal <----#4 Create subscriber record.
password pass1
ip address pool
circuit-group-member cg1 <----#5 Specify subscriber session circuit as a circuit group member
                           of VPCG "cg1".

subscriber name sally
password pass2
ip address pool
circuit-group-member cg2
subscriber name santosh
password pass1
ip address pool
circuit-group-member cg3

port eth 6/1
no shut
encap dot1q

!
```



5.9.3 Explicit Assignment of Circuit-Group Membership to Link-Group Based VPCGs

This section provides an example of explicit assignment of circuit-group membership of a circuit to a link group based VPCG. This example highlights the following configurations:

- Create and configure PWFQ policy (see #1 in example)
- Define link group based VPCGs (see #2 in example)
- Define a LAG and enable it to use PWFQ scheduling on a virtual port (see # 3 and 4 in example). The `virtual-port` keyword is only supported on a 10 GE 4-port card.
- Create an 802.1Q PVC and enter the PVC configuration mode (see # 5 in example)
- Specify the circuit-group membership of the 802.1Q PVC (SVLAN) to a link group based VPCG (see # 6 in example). This shows an explicit assignment of circuit group membership.



```
qos policy pwfq-policy1 pwfq <----#1 Create and configure PWFQ policy
rate maximum 12000
queue 0 priority 0 weight 100
queue 1 priority 1 weight 100
queue 2 priority 2 weight 100
queue 3 priority 3 weight 100
queue 4 priority 4 weight 100
queue 5 priority 5 weight 100
queue 6 priority 6 weight 100
queue 7 priority 7 weight 100
queue priority-group 0 rate 5000
queue priority-group 1 rate 2500
queue priority-group 2 rate 1200
queue priority-group 3 rate 1000
queue priority-group 4 rate 800
queue priority-group 5 rate 600
queue priority-group 6 rate 500
queue priority-group 7 rate 400

circuit-group cg1 link-group lg1 virtual-port
!<----#2 Define VPCG and specify an access link group on which all circuits in this circuit group reside.
!
circuit-group cg2 link-group lg1 virtual-port
!
circuit-group cg3 link-group lg1 virtual-port
!

link-group lg1 access <----#3 Define LAG.
encapsulation dot1q
qos pwfq scheduling virtual-port <----#4 Enable an access link group to use PWFQ (or TM) scheduling on a
virtual port. The virtual-port keyword is only supported on a 10 GE 4-port card.
mac-address auto
!
dot1q pvc 1 encapsulation pppoe <----#5 Create an 802.1Q PVC.
qos policy queuing pwfq-policy
circuit-group-member cg1 <----#6 Specify the circuit-group membership of this circuit to VPCG cg1.
bind subscriber joe@abc password pass
dot1q pvc 2 encapsulation pppoe
qos policy queuing pwfq-policy1
circuit-group-member cg2
bind authentication chap pap context abc maximum 10
dot1q pvc 3 encapsulation pppoe
qos policy queuing pwfq-policy1
circuit-group-member cg3
bind authentication chap pap context abc
lcp active

!
```

5.9.4 Auto-Assignment of VPCG LAG Circuits

This section provides an example of an auto-assignment of a link group based VPCG. This example highlights the following configurations:

- Define link group based VPCGs (see #1 in example)
- Define a LAG and enable it to use PWFQ scheduling on a virtual port (see # 2 and 3 in example)
- Create an 802.1Q PVC and enter the PVC configuration mode (see # 4 in example)
- Attach PWFQ scheduling policy to the 802.1Q PVC (see # 5 in example). Here the circuit on the link group requires virtual-port scheduling, and it is provisioned for PWFQ (using the `qos policy queuing` command in



this case). The circuit does not have an explicit or inherited virtual-port membership, and so it is automatically assigned to one of the existing VPCGs that have been configured for that link group.

```
circuit-group cg1 link-group lg1 virtual-port <---#1 Define VPCG and specify an access link group on which
!                                     all circuits in this circuit group reside.
circuit-group cg2 link-group lg1 virtual-port
!
circuit-group cg3 link-group lg1 virtual-port
!
!
link-group lg1 access <----#2 Define LAG.
encapsulation dot1q
qos pwfq scheduling virtual-port <----#3 Enable an access link group to use PWFQ (or TM) scheduling
mac-address auto                                     on a virtual port.
!
dot1q pvc 1 encapsulation pppoe <----#4 Create an 802.1Q PVC.
qos policy queuing pwfq-policy <----#5 Attach PWFQ scheduling policy to the circuit.
bind subscriber joe@abc password pass
dot1q pvc 2 encapsulation pppoe
qos policy queuing pwfq-policy
bind authentication chap pap context abc maximum 10
dot1q pvc 3 encapsulation pppoe
qos policy queuing pwfq-policy
bind authentication chap pap context abc
lACP active
!
!
```

5.9.5 Auto-Assignment of Static PVC

This section provides an example of an auto-assignment of a static PVC. The first configuration example shows the initial configuration that is entered in the CLI. The second example shows the configuration that is saved in the SmartEdge router as a result of the entered configuration. Due to the addition of the `qos weight` TM command, PVC 10 is auto assigned to VPCG vp1-1-1 (or another VP that may have previously been configured for that port). Entered Configuration:

```
circuit-group vp1-1-1 port 1/1 virtual port

port ethernet 1/1
encapsulation dot1q
dot1q pvc 10
qos weight 100
!
```

Resulting saved configuration:

```
circuit-group vp1-1-1 port 1/1 virtual port

port ethernet 1/1
encapsulation dot1q
dot1q pvc 10
circuit-group-member vp1-1-1 <---- This line is automatically added to the
qos weight 100                                     configuration of the PVC.
```



5.9.6 Auto-Assignment of Subscribers (Dynamic Circuit)

The following example shows how to configure auto-assignment of a subscriber to a dynamic circuit. As a result, whenever a PPPoE subscriber comes up on the port, it will be auto assigned to VPCG `vp1-2-1` (or another VPCG that may have previously been configured for that port) because of the PWFQ policy configuration (`qos policy queuing pwfq_gold`) on the default subscriber profile. No changes are applied to the configuration as a result of the VPCG auto-assignment. The circuit-group assignment is specific to this instance of the subscriber session and is not saved permanently. Dynamic circuits do not have their auto-assigned VPCG membership reflected in the router configuration.

```
circuit-group vp1-2-1 port 1/2 virtual port

context local
  subscriber default
  qos policy queuing pwfq_gold

port ethernet 1/2
  encapsulation pppoe
  bind authentication pap chap context local
```