

Chapter 8

Configuring Strict-Priority Scheduling

This chapter provides information for configuring strict-priority scheduling.

QoS topics are discussed in the following sections:

- Strict-Priority and Relative Strict-Priority Scheduling Overview on page 61
- Comparison of True Strict Priority with Relative Strict Priority Scheduling on page 63
- Configuring Strict-Priority Scheduling on page 68
- Configuring Relative Strict-Priority Scheduling for Aggregate Shaping Rates on page 70

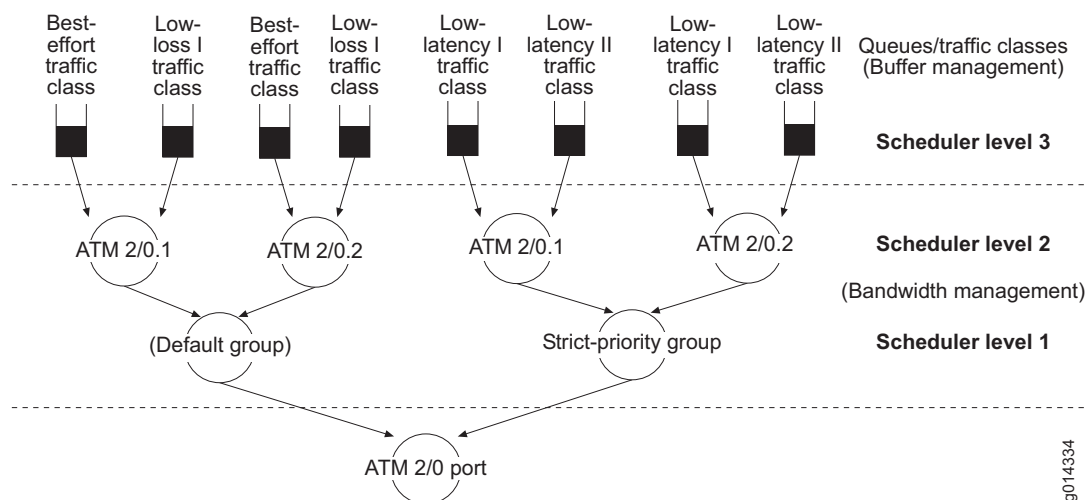
Strict-Priority and Relative Strict-Priority Scheduling Overview

You can configure one or more strict-priority queues per interface. Strict-priority scheduling is implemented with a special strict-priority scheduler node that is stacked directly above the port. Queues stacked on top of the strict-priority scheduler node always get bandwidth before other queues.

You can configure only one node at the first scheduler level as strict priority. If any node or queue above the strict-priority node has packets, it is scheduled next. If multiple queues above the strict-priority node have packets, the HRR algorithm selects which strict-priority queue is scheduled next.

Figure 11 illustrates an example of a QoS scheduler's hierarchy.

Figure 11: Sample Strict-Priority Scheduling Hierarchy



One strict priority traffic-class group is called the auto-strict-priority group. The scheduler nodes and queues in the auto-strict-priority group receive strict-priority scheduling. If multiple queues above the strict-priority node have packets, the HRR algorithm selects which strict-priority queue is scheduled next.



NOTE: If you configured traffic shaping through traffic shape profiles in JUNOS releases before Release 4.0, traffic shaping is replaced with the rate-shaping feature, which is configured when you configure a scheduler profile.

Relative Strict-Priority Scheduling Overview

Relative strict-priority scheduling provides strict-priority scheduling within a shaped aggregate rate. For example, it allows you to provide 1 Mbps of aggregate bandwidth to a subscriber, with up to 500 Kbps of the bandwidth for low-latency traffic. If there is no strict-priority traffic, the low-latency traffic can use up to the full aggregate rate of 1 Mbps.

Relative strict priority differs from true strict priority in that it can implement the aggregate shaping rate for both strict and nonstrict traffic. With true strict priority, you can shape the nonstrict or the strict traffic separately, but you cannot shape the aggregate to a single rate.

The best application of relative strict priority is on Ethernet, where you can shape the aggregate for each VLAN to a specified rate, and provision a strict and nonstrict queue for each VLAN above the shaped VLAN node.

To use relative strict priority, you configure strict-priority queues above the VC or VLAN scheduler node, thereby providing for strict-priority scheduling of the queues within the VC or VLAN. You configure relative strict priority without using QoS traffic-class groups, which causes strict-priority queues to appear in the same scheduler hierarchy as the nonstrict queues.

Relative strict priority provides low latency only if you undersubscribe the port by shaping all VCs on the port so that the sum of the shaping rates is less than the port rate. The port will not become congested, and the latency caused by the round-robin behavior of both the HRR and cell schedulers is nominal. In these undersubscribed conditions, the latency of a strict-priority queue within each VC is calculated as if the VC were draining onto a wire with bandwidth equal to the shaped rate.

Relative strict priority is carried out in the HRR scheduler on E-series ASIC line modules.

Related Topics

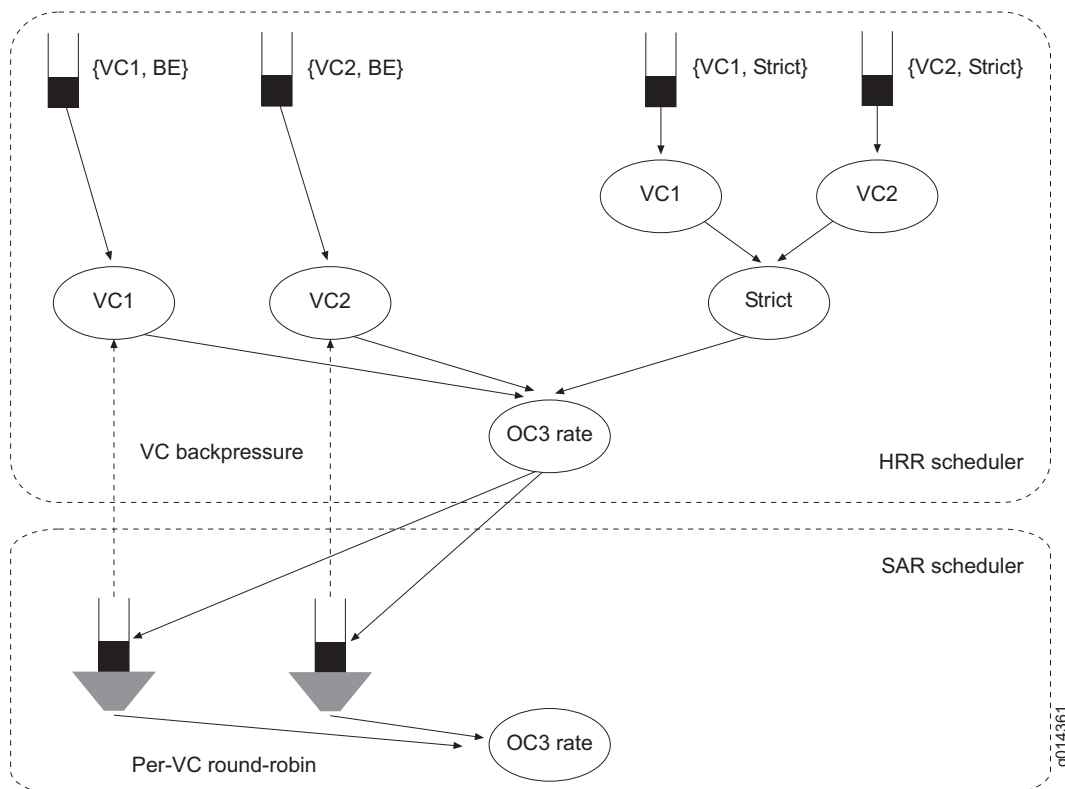
- Comparison of True Strict Priority with Relative Strict Priority Scheduling on page 63
- Configuring Strict-Priority Scheduling on page 68
- Configuring Relative Strict-Priority Scheduling for Aggregate Shaping Rates on page 70

Comparison of True Strict Priority with Relative Strict Priority Scheduling

This section explains how the HRR and SAR schedulers handle true strict-priority and relative strict-priority configurations.

Schedulers and True Strict Priority

In the strict-priority configuration in Figure 12, the queues stacked above the single strict priority scheduler node make up a round-robin separate from the nonstrict queues. All strict queues are drained to completion first, and any residual bandwidth is allocated to the nonstrict round-robin.

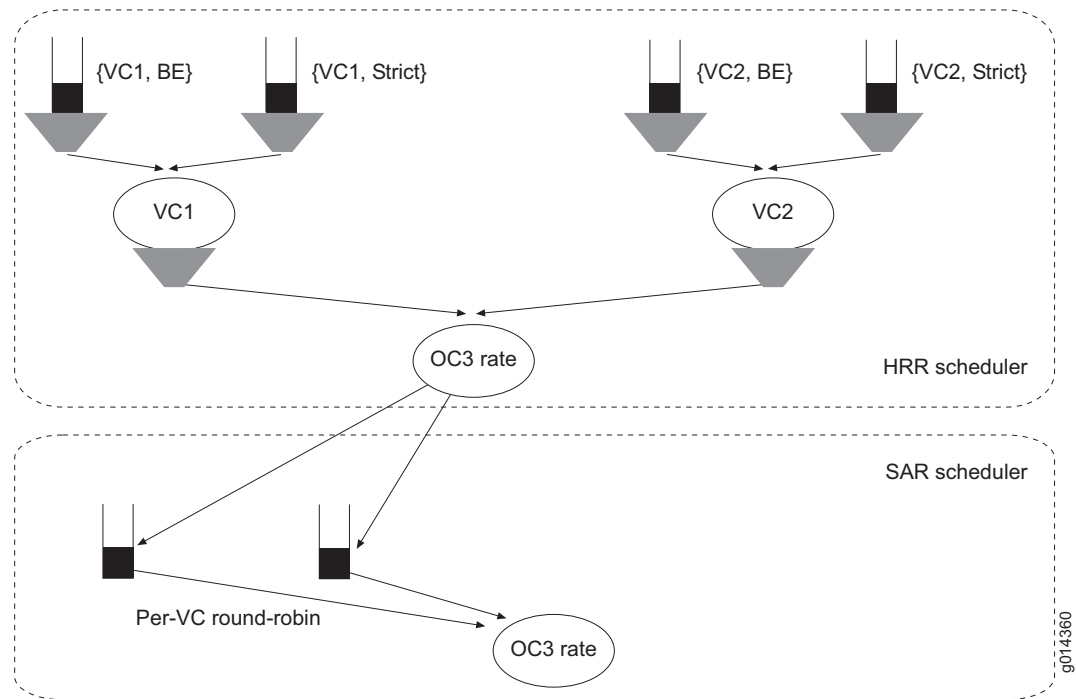
Figure 12: True Strict-Priority Configuration

This configuration provides low latency for the strict-priority queues, irrespective of the state of the nonstrict queues. The worst-case latency for a strict packet caused by a nonstrict packet is the propagation delay of a single large packet at the port rate. For a 1500 byte frame at OC3 rate, that latency is less than 100 microseconds.

Because the strict and nonstrict packets for a VC are scheduled in separate round robins, the scheduler cannot enforce an aggregate rate for both of them.

Schedulers and Relative Strict Priority

In the relative strict-priority configuration in Figure 13, the scheduler provides relative strict-priority scheduling relative to the VC. If the port is not oversubscribed, the VC round robin does not cause significant latency.

Figure 13: Relative Strict-Priority Configuration

This configuration provides a latency bound for the relative strict-priority queues. The worst-case latency caused by a nonstrict packet is the propagation delay of a single large packet at the VC rate. For a 1500 byte frame at a 2 Mbps rate, that delay is about 6 milliseconds.

This configuration provides for shaping the aggregate of nonstrict and relative strict packets to a single rate, and it is consistent with the traditional ATM model. It does not scale as well as true strict priority, because the nonstrict and relative strict traffic together must not oversubscribe the port rate.

Relative Strict Priority on ATM Modules

You can use relative strict priority on any type of E-series line module; however, on ATM line modules you have an alternative. On ATM line modules you can configure true strict-priority queues in the HRR scheduler and shape the aggregate for the VC in the SAR scheduler. VC backpressure affects only the nonstrict traffic for the VC. For this type of configuration, you should shape the relative strict traffic for each VC in the HRR scheduler to a rate that is less than the aggregate VC rate. This shaping prevents the VC queue in the SAR scheduler from being congested with strict-priority traffic.

The major difference between relative and true strict priority on ATM line modules is that relative strict priority shapes the aggregate for the VC to a pre-cell tax rate, whereas true strict priority shapes the aggregate for the VC to a post-cell tax rate. For example, shaping the VC to 1 Mbps in the HRR scheduler allows 1 Mbps of frame data, but cell tax adds anywhere from 100 Kbps to 1 Mbps additional bandwidth, depending on packet size. Shaping the VC to 1 Mbps in the SAR scheduler allows just 1 Mbps of cell bytes regardless of packet size.

Oversubscribing ATM Ports

You cannot oversubscribe ATM ports and still achieve low latency with relative strict-priority scheduling. There are several ways to ensure that ports are not oversubscribed. The most common is to use a per-VC scheduler by configuring the HRR scheduler with either ATM VP or VC node shaping (using the **atm-vp node** or **atm-vc node** commands), and setting the sum of the shaping rates less than the port rate. In these scenarios, the cell residency in the SAR scheduler is minimal, and cell scheduling does not interfere with relative strict priority.

Minimizing Latency on the SAR Scheduler

There are two methods you can use to control latency on the SAR scheduler. In the first method, you set the ATM QoS port mode to low-latency mode. In low-latency mode, the HRR scheduler controls scheduling, buffering in the SAR scheduler is limited, and latency caused by the SAR scheduler is minimized.

You can also use the default **no qos-mode-port** mode of SAR operation to minimize the latency induced by the SAR. In this method, you set **qos shaping-mode** cell and shape an OC-3 ATM port to 149 Mbps, or an OC-12 ATM port to 600 Mbps. By throttling the rate at which the HRR scheduler delivers packets to the SAR, you bound SAR buffering and latency. This approach retains the flexibility to configure different ATM QoS in the SAR, including shaped VP tunnels, UBR + PCR, nrtVBR, and CBR services.

To set the SAR mode, use the **qos-mode-port** command. For more information about operational modes on ATM interfaces, see *Chapter 19, Configuring an Integrated Scheduler to Provide QoS for ATM*.



NOTE: Controlling latency is not normally required. If you undersubscribe the port rate in the HRR scheduler, you can obtain latency bounds without modifying the SAR mode of operation.

HRR Scheduler Behavior and Strict-Priority Scheduling

The HRR scheduler does not offer native strict-priority scheduling above the first scheduler level in the hardware; however, you can configure very large weights in the round robin in the HRR scheduler to obtain approximate strict-priority scheduling. Note that under conditions of low VC bandwidth and large packet sizes, latency and jitter increase because of the inherent propagation delay of large packets over a small shaping rate. The following sections describe additional configuration steps that will ensure that no more than a single nonstrict packet can precede a strict-priority packet on the VC.

Zero-Weight Queues

To reduce latency and jitter, you can configure the relative strict-priority queue with a weight of 0 (zero), which gives the queue a weight of 4080. When a packet arrives at a zero-weighted queue, the queue remains in the active WRR until it is exhausted, whereas competing queues must leave the active WRR because their weight credits are exhausted. To completely drain the queue, configure the maximum burst size. The zero-weighted queue is eventually alone in the active round robin and is effectively drained at strict priority.

To configure more than one relative strict queue or node, simply configure a maximum weight, and the two relative strict queues or nodes will share bandwidth fairly. You can shape the nonstrict queue, as described in the next section, to keep latency bounded.

Also, configure only a few nonstrict nodes or queues to prevent additional latency and jitter of the relative strict-priority traffic when the nodes or queues are in the round robin and a packet arrives in the zero-weighted queue. The number of nonstrict frames that precede a relative strict frame equals the number of nonzero weighted queues among the sibling scheduler nodes.

Nonstrict queues must still exhaust their weight credits before they leave the active round robin. The result is that occasionally more than one nonstrict frame may precede a relative strict frame, causing more jitter than may be acceptable. You can eliminate this source of latency by shaping the nonstrict queue to the aggregate rate with a burst size of 1.

Setting the Burst Size in a Shaping Rate

The burst value in a shaping rate determines the number of rate credits that can accrue when the queue or scheduler node is held in the inactive round robin. When the queue is back on the active list, the accrued credits allow the queue or node to catch up to the configured rate, up to the burst value.

Normally, the burst size is several packet lengths to allow a queue deprived of bandwidth because of congestion to catch up to its rate. Larger burst sizes allow more bursting to allow the queue to attain its shaped rate under bursty congestion scenarios.

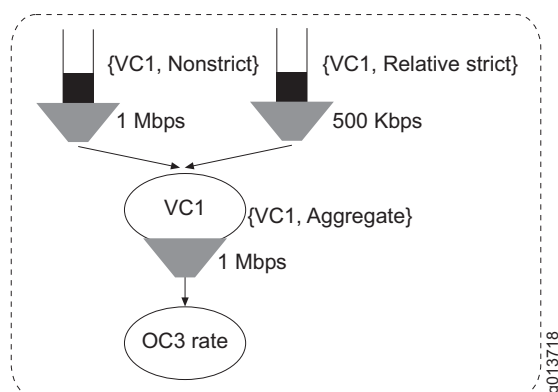
Special Shaping Rate for Nonstrict Queues

To remove additional jitter, you can configure the nonstrict queue with a special shaping rate that causes the hardware to temporarily eject the queue from the active round robin whenever it sends a frame. The result is that at most one nonstrict frame can precede a relative strict-priority frame. The special shaping rate is the same rate as the aggregate rate, but with a configured burst size of 1.

You can still configure a shaping rate for the zero-weighted queue or node. This is useful for limiting starvation of the nonstrict traffic in the aggregate.

In Figure 14, the VC node is shaped in the HRR scheduler to 1 Mbps to limit the aggregate traffic for the subscriber. The relative strict traffic is shaped to 500 Kbps. This shaping limits relative strict traffic to 500 Kbps, and prevents the relative strict-priority traffic from starving out the nonstrict traffic.

The third shaper, on the nonstrict queue, is subtle. The rate is 1 Mbps, which allows the nonstrict traffic to consume up to the full aggregate rate of the VC. But the burst size is 1, which causes the nonstrict queue to always yield to the relative strict-priority queue after sending a packet. This burst size limits the number of nonstrict packets that can precede a relative strict-priority packet to the minimum, one packet.

Figure 14: Tuning Latency on Strict-Priority Queues

Related Topics

- Strict-Priority and Relative Strict-Priority Scheduling Overview on page 61
- Configuring Strict-Priority Scheduling on page 68
- Relative Strict-Priority Scheduling Overview on page 62

Configuring Strict-Priority Scheduling

To configure strict-priority scheduling:

1. Configure the traffic classes.

```

host1(config)#traffic-class Low-loss-1
host1(config-traffic-class)#exit
host1(config)#traffic-class Low-latency-1
host1(config-traffic-class)#exit
host1(config)#traffic-class Low-latency-2
host1(config-traffic-class)#exit

```

2. Configure the auto-strict-priority traffic-class group, and add the traffic classes that must receive strict-priority scheduling to the group.

```

host1(config)#traffic-class-group Strict-priority auto-strict-priority
host1(config-traffic-class-group)#traffic-class Low-latency-1
host1(config-traffic-class-group)#traffic-class Low-latency-2
host1(config-traffic-class-group)#exit

```

3. Create a scheduler profile for strict-priority traffic and configure the shaping rate.

```

host1(config)#scheduler-profile strictPriorityBandwidth
host1(config-scheduler-profile)#shaping-rate 20000000
host1(config-scheduler-profile)#exit

```


4. Configure a QoS profile.

```

host1(config)#qos-profile Example-qos-profile
host1(config-qos-profile)#atm group default
host1(config-qos-profile)#atm group Strict-priority scheduler-profile
strictPriorityBandwidth
host1(config-qos-profile)#atm-vc node group default
host1(config-qos-profile)#atm-vc node group Strict-priority
host1(config-qos-profile)#atm-vc queue traffic-class best-effort
host1(config-qos-profile)#atm-vc queue traffic-class Low-loss-1
host1(config-qos-profile)#atm-vc queue traffic-class Low-latency-1
host1(config-qos-profile)#atm-vc queue traffic-class Low-latency-2
host1(config-qos-profile)#exit

```

5. Attach the QoS profile to an interface.

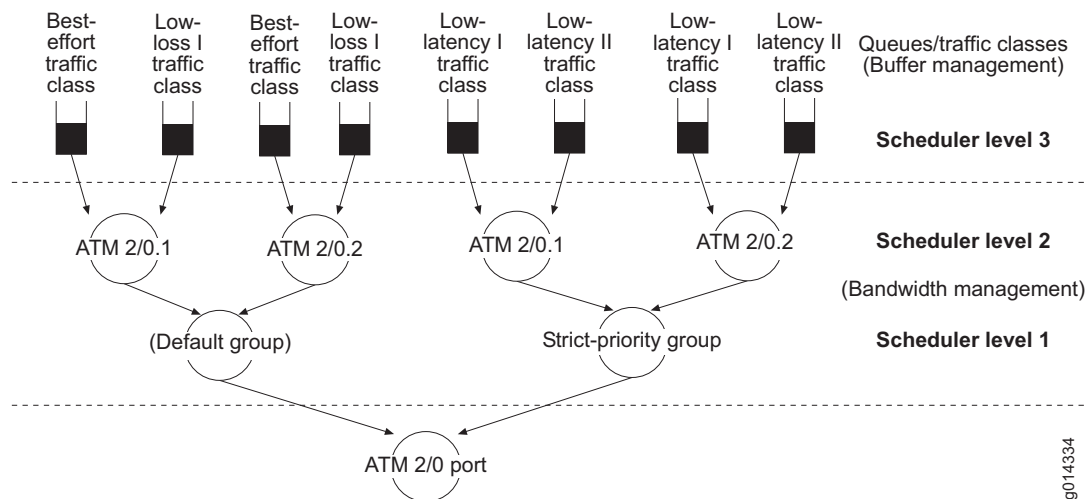
```

host1(config)#interface atm 2/0
host1(config-if)#qos-profile Example-qos-profile
host1(config-if)#exit
host1(config)#

```

This configuration creates the hierarchy shown in Figure 15.

Figure 15: Sample Strict-Priority Scheduling Hierarchy



Related Topics

- Strict-Priority and Relative Strict-Priority Scheduling Overview on page 61
- For more information about specifying an expression that you can reference within a scheduler profile, see *Using Expressions for Bandwidth and Burst Values in a Scheduler Profile* on page 51
- **group** command
- **node** command

- **qos-profile** command
- **queue** command
- **scheduler-profile** command
- **shaping-rate** command
- **strict-priority** command
- **traffic-class** command
- **traffic-class-group** command

Configuring Relative Strict-Priority Scheduling for Aggregate Shaping Rates

To configure relative strict priority scheduling for aggregate shaping rates:

1. Create a scheduler profile for the strict-priority queue.

```
host1(config)#scheduler-profile relativeStrict
host1(config-scheduler-profile)#shaping-rate 500000
host1(config-scheduler-profile)#weight 0
host1(config-scheduler-profile)#exit
```

Configuring the weight of 0 reduces latency and jitter.

2. Create a scheduler profile for the nonstrict best-effort queue.

```
host1(config)#scheduler-profile be
host1(config-scheduler-profile)#shaping-rate 1000000 burst 1
host1(config-scheduler-profile)#weight 8
host1(config-scheduler-profile)#exit
```



TIP: If you need to impose a shaping rate on the nonstrict queues to meet a functional requirement, you can specify a rate less than the aggregate rate. The key is that the burst size must be one, or small. The burst size determines the maximum-sized packet that can squeeze in front of a relative strict-priority packet in the round robin.

3. Create a scheduler profile for the aggregate bandwidth.

```
host1(config)#scheduler-profile vcAggregate
host1(config-scheduler-profile)#shaping-rate 1000000
host1(config-scheduler-profile)#exit
```

4. Create a QoS profile, configure node shaping for each queue, and add each of the queues to the QoS profile.

```
host1(config)#qos-profile relative-strict-aggregate
host1(config-qos-profile)#atm-vc node scheduler-profile vcAggregate
host1(config-qos-profile)#atm-vc queue traffic-class best-effort
scheduler-profile be
```

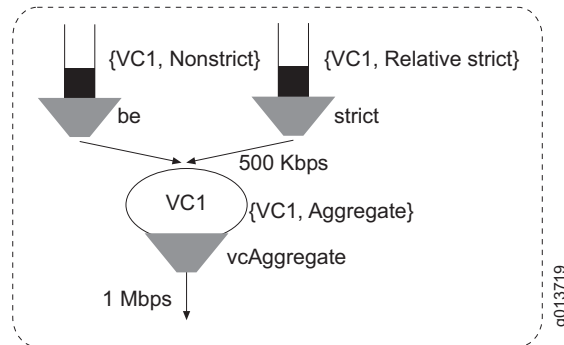
```

host1(config-qos-profile)#atm-vc queue traffic-class voice scheduler-profile
relativeStrict
host1(config-qos-profile)#exit
host1(config)#

```

This configuration creates the hierarchy shown in Figure 16.

Figure 16: Sample Relative Strict-Priority Scheduler Hierarchy



Related Topics

- Strict-Priority and Relative Strict-Priority Scheduling Overview on page 61
- For more information about specifying an expression that you can reference within a scheduler profile, see *Using Expressions for Bandwidth and Burst Values in a Scheduler Profile* on page 51
- **node** command
- **qos-profile** command
- **scheduler-profile** command
- **shaping-rate** command
- **weight** command

