



Policy-Based QOS Configuration

Version: 2835

Copyright 2007-2010 ImageStream Internet Solutions, Inc., All rights Reserved.

Table of Contents

Policy-Based QOS Configuration.....	1
QOS_SERVICE.....	1
QOS_POLICY.....	1

Policy-Based QoS Configuration

The latest 4.4.0 and 4.2.12 distributions utilize an improved policy-based configuration for quality of service. The default `rc.qos` script has examples that use a new QoS library of functions, greatly simplifying QoS configuration.

Services define a type of traffic or service. Policies prioritize services, define how the traffic is queued and set minimum and maximum rates as a percentage of the maximum rate. Interface definitions apply a policy to an interface and define the maximum input and output rates as well as the maximum allowed latency for queued traffic. Because policy rates are specified using percentages a policy can be applied to multiple interfaces which may have differing input and output rates.

QOS_SERVICE

The `QOS_SERVICE` command defines parameters that match a type of service such as Web traffic or SMTP traffic. ImageStream pre-defines many services that can be added to or modified.

Currently only `tc` filter matches are supported. See the Linux Advanced Routing and Traffic Control guide for more information on `tc` filter matching.

Usage: `QOS_SERVICE --name <service_name> [--init] --tc_match <tc_filter_command_match>`

Example:

Create a new service called `ims1` that matches traffic to and from `205.159.243.5`

```
QOS_SERVICE --name ims1 --init --tc_match protocol ip u32 match ip src 205.159.243.5
QOS_SERVICE --name ims1 --tc_match protocol ip u32 match ip dst 205.159.243.5
```

Release 4.4.0-77 adds a new `--host` option to simplify the match above:

```
QOS_SERVICE --name ims1 --init --host 205.159.243.5
```

It is also possible to add on to a pre-defined service. Simply omit the `--init` option to add another match to a service.

Example:

In addition to the IP ToS bit matches also match on the SIP server's IP at `205.159.243.5`.

```
QOS_SERVICE --name voip --tc-match protocol ip u32 match ip src 205.159.243.5
QOS_SERVICE --name voip --tc-match protocol ip u32 match ip dst 205.159.243.5
```

QOS_POLICY

The `QOS_POLICY` command prioritizes and rate-limits services. Low latency queueing and class-based queueing are supported.

Low latency queueing provides a strict priority-based first-in-first-out (FIFO) queue which does not rate shape the traffic. This queueing method is preferred for real-time traffic such as VoIP and routing protocols.

Priorities range from 0 to 9. Lower values indicate higher priority.

Minimum and maximum values are expressed as percentages ranging from 1 to 100. The actual rate will be calculated using this percentage and the interface's real bandwidth.

Class-based queueing provides prioritization but also performs rate shaping. This queueing method is preferred for non-realtime traffic such as Web and E-mail.

LLQ Usage: QOS_POLICY --name <policy_name> --type llq --service <service_name> --prio <priority 0-9> --max <max_percent>

Example:

```
QOS_POLICY --name default --type llq --service voip --prio 0 --max 90
```

CBQ Usage: QOS_POLICY --name <policy_name> --type cbq [--default | --service <service_name>] --prio <priority 0-9> --min <min_percent> --max <max_percent>

Example:

Set the default traffic class to priority 5 and prioritize TCP ACKs higher than other traffic.

```
QOS_POLICY --name default --type cbq --default --prio 5 --min 10 --max 75
QOS_POLICY --name default --type cbq --service ack --prio 3 --min 10 --max 40
```

QOS_IFACE

The QOS_IFACE command applies a policy to an interface. The maximum input and output rates are specified in Kbps (Kilobits per second). Latency is specified in milliseconds.

Fair queueing can also be specified on the interface. ImageStream's PUFQ, standard fair-queueing and simple first-in-first-out (FIFO) queueing are options.

FIFO Usage: QOS_IFACE --name <interface_name> --policy <policy_name> --max_in <input_rate_in_Kbps> --max-out <output_rate_in_Kbps> --max-latency <max_latency_in_ms>

Example:

Assign the default policy to Ethernet0 which has a 3.0 Mbps input and 640 Kbps output rate. We want 150 ms max latency.

```
QOS_IFACE --name eth0 --policy default --max_in 3000 --max_out 640 --max_latency 150
```

Fair-Queueing Usage: QOS_IFACE --name <interface_name> --policy <policy_name> --max_in <input_rate_in_Kbps> --max-out <output_rate_in_Kbps> --max-latency <max_latency_in_ms> --fair_queue
sfq advanced options

Example:

Assign the default policy to Ethernet0 using fair queueing which has a 3.0 Mbps input and 640 Kbps output rate. We want 150 ms max latency.

```
QOS_IFACE --name eth0 --policy default --max_in 3000 --max_out 640 --max_latency 150 --fair_queue
```

PUFQ Usage: QOS_IFACE --name <interface_name> --policy <policy_name> --max_in <input_rate_in_Kbps> --max-out <output_rate_in_Kbps> --max-latency <max_latency_in_ms> --pufq --pufq_iface_type <inside | outside | nat_inside | nat_outside>
pufq advanced options

Example:

Assign the default policy to Ethernet0 using PUFQ which has a 3.0 Mbps input and 640 Kbps output rate. We want 150 ms max latency.

```
QOS_IFACE --name eth0 --policy default --max_in 3000 --max_out 640 --max_latency 150 --pufq --puf
```

ATM/DSL Adjustments

Starting in 4.4.0-72 and 4.2.12-46 a new `--calc_atm` parameter was added to the `QOS_IFACE` statement to allow the QOS subsystem to calculate ATM/AAL5 overhead for each packet being transmitted. This allows the system to specify the exact ATM data rate for the link.

Example:

A 3.0 Mb / 512 K DSL link has synced up at 3520000 bps down and 768000 bps up.

```
QOS_IFACE --name eth0 --policy default --max_in 3520 --max_out 768 --max_latency 150 --fair_queue
```

Frame Relay Parameters

frame relay options

Example:

A 256K CIR burstable to 384K Frame Relay circuit

```
QOS_IFACE --name Serial0.1 --policy default --max_latency 150 --fair_queue --frame_relay_mincir 2
```

Advanced Example

```
#####
#
# Advanced example
# Override the voip service definition. We define voip as traffic to/from
# 192.168.0.35
# Policy:
# Low Latency Queues:
#   voip: Low Latency Queue with highest priority and maximum of 90% of the
#         interface's bandwidth.
# Class-Based Queues:
#   telnet: Highest priority with min of 10% and max 40%.
#   ssh: Lower priority than telnet with min 10% and max 40%.
#   tcp acks: Lower priority than ssh with min of 10% and max of 40%.
#   smtp: Lowest priority with min of 10% and max 60%.
#   default: Priority 5 (just above smtp) with min 10% max 95%
#
# Interface:
# Apply the policy "default" to interface eth0 with the maximum input rate of
# 3000 Kbps (3.0 Mbps) and maximum output rate of 640 Kbps with a maximum
# queuing latency of 150 ms. Configure PUFQ with nat_outside as the interface type
# since this interface performs SNAT on user traffic.
#
QOS_SERVICE --name voip --init --tc_match protocol ip u32 match ip src 192.168.0.35/32
QOS_SERVICE --name voip --tc_match protocol ip u32 match ip dst 192.168.0.35/32
QOS_POLICY --name default --type llq --service voip --prio 0 --max 90
QOS_POLICY --name default --type cbq --init --default --prio 5 --min 10 --max 95
QOS_POLICY --name default --type cbq --service telnet --prio 1 --min 10 --max 40
QOS_POLICY --name default --type cbq --service ssh --prio 2 --min 10 --max 40
```

```

QOS_POLICY --name default --type cbq --service ack          --prio 3 --min 10 --max 40
QOS_POLICY --name default --type cbq --service smtp        --prio 6 --min 10 --max 60

QOS_IFACE --name eth0 --policy default --max_in 3000 --max_out 640 --max_latency 150 --pufq --pufq_iface

```

Statistics

Release 4.4.0-83 introduces a new command line utility *qos_stats* to retrieve QoS statistics. You must enter the bash shell (available from the advanced menu) to use this utility. Without any parameters the utility will display statistics on all known services for all policies on all interfaces. You can filter the results by specifying the policy, type, service, or interface.

qos_stats usage: `qos_stats --policy <policy_name> --type <LLQ | CBQ> --service <service_name> --iface <interface_name> [--machine]`

Specify any number of the policy, type, service or iface filters to narrow the results. Use the `--machine` flag to produce machine parseable output with one line per value.

Examples:

```

Router:/usr/local/sand# qos_stats
Interface brSerial0.1, Policy default, Service voip, Type LLQ, Priority 0
  Rx limits: 3420 Kbps min, 3420 Kbps max
  Rx 895678 packets 231548469 bytes (dropped 0 overlimits 0)
  Rx backlog 0 packets 0 bytes 0 ms
  Tx limits: 728 Kbps min, 728 Kbps max
  Tx 897262 packets 245654314 bytes (dropped 0 overlimits 0)
  Tx backlog 0 packets 0 bytes 0 ms
Interface brSerial0.1, Policy default, Service default, Type CBQ, Priority 5
  Rx limits: 290 Kbps min, 2616 Kbps max
  Rx 2542388 packets 3422291056 bytes (dropped 39545 overlimits 0)
  Rx backlog 0 packets 0 bytes 0 ms
  Tx limits: 58 Kbps min, 523 Kbps max
  Tx 512702 packets 108693924 bytes (dropped 673 overlimits 0)
  Tx backlog 0 packets 0 bytes 0 ms

Router:/usr/local/sand# qos_stats --service voip --iface brSerial0.1
Interface brSerial0.1, Policy default, Service voip, Type LLQ, Priority 0
  Rx limits: 3420 Kbps min, 3420 Kbps max
  Rx 895768 packets 231573574 bytes (dropped 0 overlimits 0)
  Rx backlog 0 packets 0 bytes 0 ms
  Tx limits: 728 Kbps min, 728 Kbps max
  Tx 897340 packets 245680162 bytes (dropped 0 overlimits 0)
  Tx backlog 0 packets 0 bytes 0 ms

```

Release 4.4.0-83 also allows querying these values via SNMP using the IMAGESTREAM-QOS-MIB.