Allied Telesyn Research Professional Services Group

Overview of the Quality of Service (QoS) features on the AT-8948 switch

Introduction

The QoS featureset on the AT-8948 switch is significantly richer than the featuresets on earlier Allied Telesyn Research (ATR) switch products such as the Rapier and AT-9800 series switches.

As the featureset on the AT-8948 is a superset of the featuresets on the earlier products, it has been possible to retain the existing overall structure of the way the features are presented to the user; the new features are presented as extensions of the existing QoS command structure.

The purpose of this document is to enable a user who already has some familiarity with the QoS features on the Rapier and AT-9800 series products to understand the new features of the AT-8948, and how they have been integrated into the existing command structure.

> ***This document will not discuss the IPv6 QoS features, just the extensions to the existing IPv4 and general Ethernet QoS features. The IPv6 QoS features are the subject of a separate document.***

A quick list of what's new in the AT-8948

To quickly display the scope of the new features, here is a very brief description of the major new features that appear on the AT-8948:

- **Bandwidth metering**
  Packets belonging to any given traffic class can be classified with respect to whether they are inside or outside the bandwidth limits set for that traffic class. The packets are marked with the classification value that was applied to them, and at various points in the QoS process, decisions on the packets' fate can be made on the basis of what classification they have been marked with.

- **Premarking**
  Right at the point of ingress into the QoS process, packets classified to particular flow groups or traffic classes can have values written to one or more of their associated 'markers'. The markers can be externally visible fields (DSCP value, 802.1p value) and/or internally visible fields (bandwidth class, queue number - these are explained further below).
- **Increased control over egress queue parameters**
  Queue lengths, scheduling process, relative weights, etc can be set on all queues on a per-port basis.
- **More configurability for the default traffic class**
  All the parameters that can be set on a normal traffic class can also be set on the default traffic class (the catch-all traffic class that matches all traffic that does not explicitly match any other traffic class).
- **The ability to see the current state of egress queue**
  There are commands that now enable you to see statistics relating to every egress queue on every port.

Each of these new features is discussed in much more detail later on in this document.

The process flow and methodology of the QoS system

Before discussing the details of the various processes that comprise the QoS system, it is desirable to first get a picture of what the processes are, and the order in which they are applied to the packets passing through the system.

What is the QoS system really trying to do to packets, and how does it keep track of what it has decided about any given packet?

In general, the main aim of all the processes in the QoS system is to work out which egress queue a particular packet should be put into.

There are several factors that can affect this choice of egress queue, so packets need to be put through several processes, so that each of the competing factors has its opportunity to exert its influence on the final choice of egress queue.

In some cases, the system can decide to simply discard certain packets at some steps in the process.

Additionally, the QoS system often has an obligation to update certain fields within a packet - to indicate to downstream devices how they should deal with the packet when it gets to them.

So, we have this multi-stage process, and the eventual fate of a packet will depend on the sum total of the various decisions that were made about it at various stages in the process. In order to keep track of the outcomes of those decisions, a packet needs to be marked so that at any point in the process it is possible to know the net effect of the decisions that have been made on it so far.

There are four items that are used to mark packets as they pass through the QoS system.

- Two markers that are carried within fields of the packet itself:

  1. 802.1p: The 802.1p or User Priority field in the VLAN tag of an Ethernet frame. This is a 3-bit number, so it can have a value in the range 0-7.
  2. DSCP: The Differentiated Services Code Point within the TOS field of an IP packet header. This is a 6-bit number, so it can have a value in the range 0-63.

- Two items that are just used within the switch chip. These are not fields within the packets, but are extra parameters that the packets carry with them as they pass through the QoS system:

  1. Bandwidth Class: This parameter can take on the values 1-3. Essentially it is an indicator of whether the packet is deemed to have been within the acceptable bandwidth limit set for any particular traffic flow; or whether the packet's traffic flow had already overflowed its acceptable limit by the time this particular packet arrived.

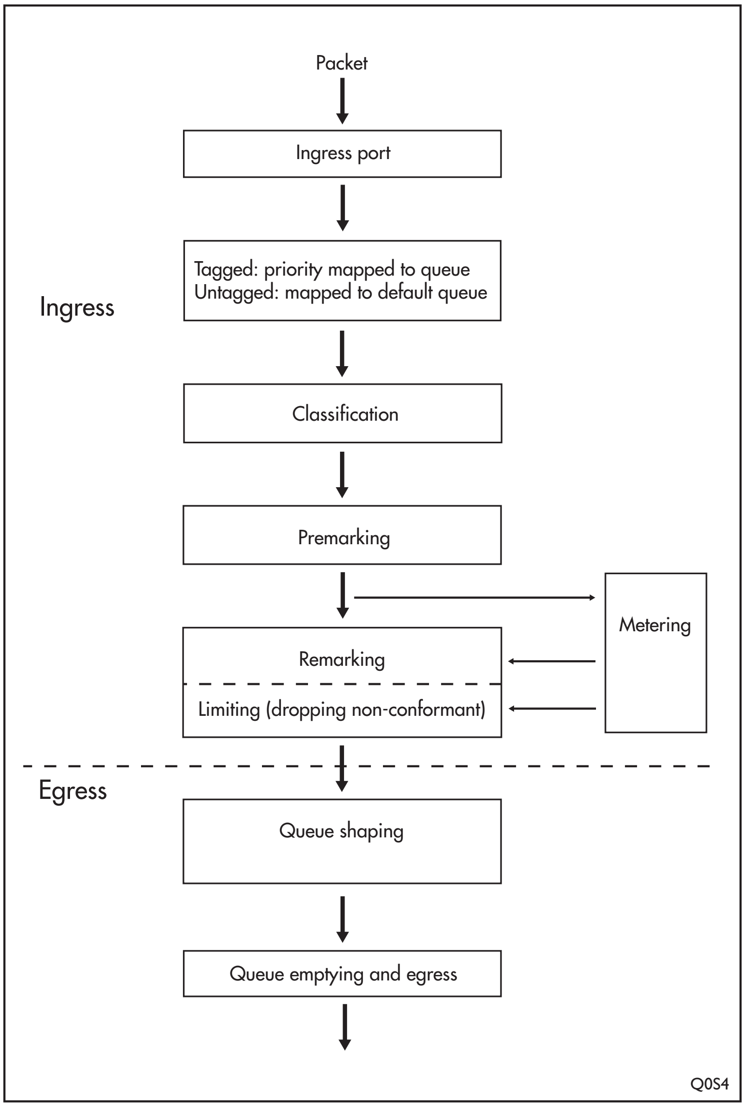
     A value of 1 indicates that the flow was within the acceptable limit when the packet arrived, a value of 2 indicates that the flow was slightly outside its acceptable limit when the packet arrived, and a value of 3 means that the flow was well outside the limit when the packet arrived.

  2. Egress Queue: This indicates the egress queue that the packet is currently slated to be placed into, if and when it finally negotiates its way through all the steps in the QoS process and lines up in one of the queues at its eventual egress port.

Outline of the QoS processing flow

Let's look at each QoS process in the order that they are applied to a packet. The diagram in Figure 1 gives a quick view of the QoS features we are about to discuss.

Figure 1: The QoS process at a glance


Packet
Ingress port
Tagged: priority mapped to queue
Untagged: mapped to default queue
Ingress
Classification
Premarking
Metering
Remarking
Limiting (dropping non-conformant)
Egress
Queue shaping
Queue emptying and egress
Q0S4

Initial mapping to an egress queue, based on 802.1p value

Immediately after ingress, a VLAN-tagged Ethernet frame can be assigned to the appropriate egress queue on the basis of the value of its VLAN Tag User Priority. This means that incoming frames that already carry meaningful priority information can be forwarded on the basis of that information. The mapping of the User Priority value to an egress queue is configurable, so the administrator can decide, for example, to send frames with a Priority value of 7 to queue 3 and frames with a Priority of 2 to queue 7.

Untagged frames don't have a VLAN Tag User Priority, so these frames can be assigned to a default queue of the administrator's choice.

The net effect of this process is to set a value on the Egress Queue marker that the packet is carrying.

Classification

Classification is simply a method of dividing the incoming traffic into traffic flows so that packets of one type can be treated differently to packets of another type. To do this, you create Classifiers using the switch's Generic Packet Classifier function. Incoming packets are inspected and may be classified on a very broad range of criteria.

Once packets have been classified, they need to be assigned to a traffic flow. There are two configuration steps involved in doing this. Firstly, the Classifiers that have been created are associated with Flow Groups, which are used to group similar traffic together (each Classifier should be added to only one Flow Group in any QoS Policy). Then the Flow Groups are associated with Traffic Classes, which are used to group similar Flow Groups together. The Traffic Class is the central component of a QoS solution. Any traffic that has not already been assigned to a configured Traffic Class by the classification process is placed in the Default Traffic Class for the QoS Policy

The classification process does not update any of the four marker values on the packet, but does dictate the path that the packet will subsequently take through the QoS system.

Premarking

The 'Pre' part of Premarking means this process happens before any bandwidth metering takes place. The 'marking' part refers to attaching QoS information to packets.

Recall that packets can be marked in four ways:

- the VLAN tag user priority
- the Differentiated Services Code Point (DSCP)
- the Bandwidth Class the packet is assigned to
- the egress queue the packet is assigned to.

A packet can have new values assigned for each of these marking values by the Premarking process. Premarking can be done at both the flow group and traffic class level. If Premarking is specified for a flow group, these settings will take precedence over the Premarking settings for the traffic class that the flow group belongs to.

The new values that are assigned to the four marking methods for a packet from a given flow group or traffic class are specified by one of two criteria:

1. the existing DSCP value of the packet; a value in the range 0-63 (different packets within the flow group or traffic class may well have different DSCP values).
2. a user-defined mark-value (in the range 0-63) that is specified for all of the packets in the flow group or traffic class.

Whichever of these two criteria is used, the value is used to index the Premarking DSCP mapping table. This is a user-defined table which maps particular DSCP values to particular sets of 802.1p, DSCP, bandwidth class, and Egress Queue values.

On the basis of its DSCP, or its user-defined mark-value, a packet can be given a new DSCP and VLAN Tag User Priority, and can be assigned to a new bandwidth class and Egress Queue.

Metering

Metering involves measuring the bandwidth used by a traffic class and comparing the measurement to the bandwidth limits that have been set for the traffic class.

The metering process allocates a temporary bandwidth class value to packets. It is important to note that the metering process does not overwrite the bandwidth class value that the packet is already carrying around with it. Instead, an extra, temporary, bandwidth class marker is attached to the packets.

When traffic first enters the switch, it is all marked with bandwidth class 1, simply because it has not been metered yet. As explained earlier, packets can be assigned a new bandwidth class at the Premarking stage, but this is not done on the basis of actual measurement of bandwidth use. At the metering stage, a traffic class's bandwidth usage is constantly monitored to see how well it conforms to the limits set for it, and the individual packets within the flow are assigned to a temporary bandwidth class depending on the traffic class's conformance to its limits at that time.

So, while a traffic class is still within its bandwidth limit, all the packets that have been classified to that traffic class are marked with a temporary bandwidth class=1. If a traffic class starts to exceed its limit, then the packets in that traffic class are given a temporary bandwidth class=2. If it starts seriously exceeding its limits, then the packets in temporary marking is bandwidth class=3.

The actual algorithms used to determine whether a traffic class is slightly exceeding its bandwidth limit or seriously exceeding the limit are described later in this document.

Limiting or remarking (dropping non-conformant packets)

Based on the temporary bandwidth class assigned to a packet at the metering stage, one of two actions can be taken:

1. the packet can be dropped if it is was assigned to bandwidth class 3 by the metering process, or
2. the packet can be remarked with new QoS property values.

The first of these two actions is straightforward; the user can choose to simply drop packets if the traffic class exceeds the bandwidth limits set for it to the extent that packets are assigned to bandwidth class 3.

Remarking is a little more complex as it is not done solely on the basis of the bandwidth blass that the packet has been assigned to; other criteria can be used to determine what the new QoS property values should be.

Remarking can be specified on the basis of a packet's:

- **Temporary Bandwidth Class**
  If this option is chosen, the temporary bandwidth class that was assigned by the metering process becomes the new bandwidth class for the packet. That is, the value of the bandwidth class marker attached to the packet is overwritten with the value that had resulted from the metering process.
- **Priority**
  If this option is chosen, the Egress Queue that the packet is currently assigned to, and the temporary bandwidth class that the packet was assigned to by the metering process are used to determine the new value for the 802.1p value for the packet. The new priority values are taken from the user-configurable **queue2priomap** table.
- **Temporary Bandwidth Class and Priority**
  If this option is chosen, both of the actions detailed above are taken; the temporary bandwidth class becomes the new bandwidth class for the packet, and a new VLAN Tag User Priority is assigned to the packet using the **queue2priomap** Table.
- **DSCP and Temporary Bandwidth Class**
  If this option is chosen, the packet's current DSCP value, and its temporary bandwidth class are used to determine the new values for all four QoS properties for the packet (that is, new values for the DSCP, VLAN Tag User Priority, bandwidth class, and Egress Queue can be specified). The new values are taken from the user-configurable remarking DCSP mapping table.

Queue shaping

Each egress port has eight egress queues, numbered 0-7 with 7 being the highest priority queue. Unfortunately, the queues are of a limited length, so packets cannot be added to them indefinitely; if the switch is congested, the queues may fill up and no more packets can be added. In this case, packets will inevitably be dropped from the end of the queues, even if they are high-priority packets. Queue shaping is a general term to describe how the egress queues can be managed to prevent the indiscriminate dropping of packets from the tails of the egress queues.

Random Early Detection/Discard (RED) is a congestion avoidance mechanism that allows some packets to be dropped before the egress queue exceeds the allocated maximum queue length. Lower priority packets are dropped when severe congestion occurs, with progressively more and higher priority packets dropped until congestion is eased. This is useful for TCP flows, because the sender will slow the rate of transmission when it detects a packet loss. Note that using RED on UDP traffic flows is not recommended because UDP does not reduce the rate of transmission and will simply retransmit the dropped packets, which will add to the congestion.

The Random Early Discarding of packets from egress queues will typically be configured to drop more packets with bandwidth class=3 than those with bandwidth class=2, and to drop even less of the packets with bandwidth class=1.

RED curves are not the only queue shaping mechanism available. You can instead choose to use a relatively simple tail-drop scheme. Using this method, you nominate a queue length at which any further packets will be dropped. This is done for each of the three bandwidth classes. Obviously, the queue-length threshold for bandwidth class 3 is set at a relatively low value, with the other bandwidth classes having progressively higher values.

Scheduling

In addition to managing the way in which packets can be dropped when the egress queues for a given port start to fill up, you can also configure the method that is used to allocate bandwidth to each of the queues to transmit packets onto the line.

There are two ways that the queues can be scheduled for transmission:

1. **Strict Priority Scheduling**
   Higher priority queues are emptied before any packets are transmitted from lower-priority queues. This means that queue 7 must be totally empty before any packets from queue 6 are transmitted.
2. **Weighted Round-Robin Scheduling**
   The queues share bandwidth on the basis of user-defined weightings. Using this method, packets from a lower-priority queue can be transmitted even when packets are waiting in a higher-priority queue. The weightings can be configured to ensure that more packets per second are sent from the higher-priority queues than from the lower-priority queues.

To allow for flexibility in scheduling, it is possible to use different scheduling methods for different queues. For a given port, you can create up to three groups of egress queues, one that uses Strict Priority Scheduling and two separate groups that each use Weighted Round-Robin Scheduling. For example, consider this case:

- queues 7, 6 & 5 are configured to use Strict Priority Scheduling

- queues 4, 3 & 2 are in Weighted Round-Robin group 1

- queues 1 & 0 are in Weighted Round-Robin group 2

Queues 7, 6 & 5 will be emptied using Strict Priority, that is, queue 7 will be emptied before any packets from queue 6 can be transmitted and queue 6 must be completely emptied before any packets from queue 5 are transmitted.

When queues 7, 6 & 5 are all completely empty, queues 4, 3 & 2 will be emptied concurrently based on their respective weightings.

Queues 1 & 0 will be emptied only when there are no packets awaiting transmission in any of the other queues.

Replacing QoS priority information on egress

As discussed earlier, packets can carry two pieces of QoS priority information:

1. the VLAN tag user priority in the VLAN tag field of an Ethernet frame. This is a 3-bit number, so it can have a value in the range 0-7.
2. the Differentiated Services Code Point within the TOS field of an IP packet header. This is a 6-bit number, so it can have a value in the range 0-63.

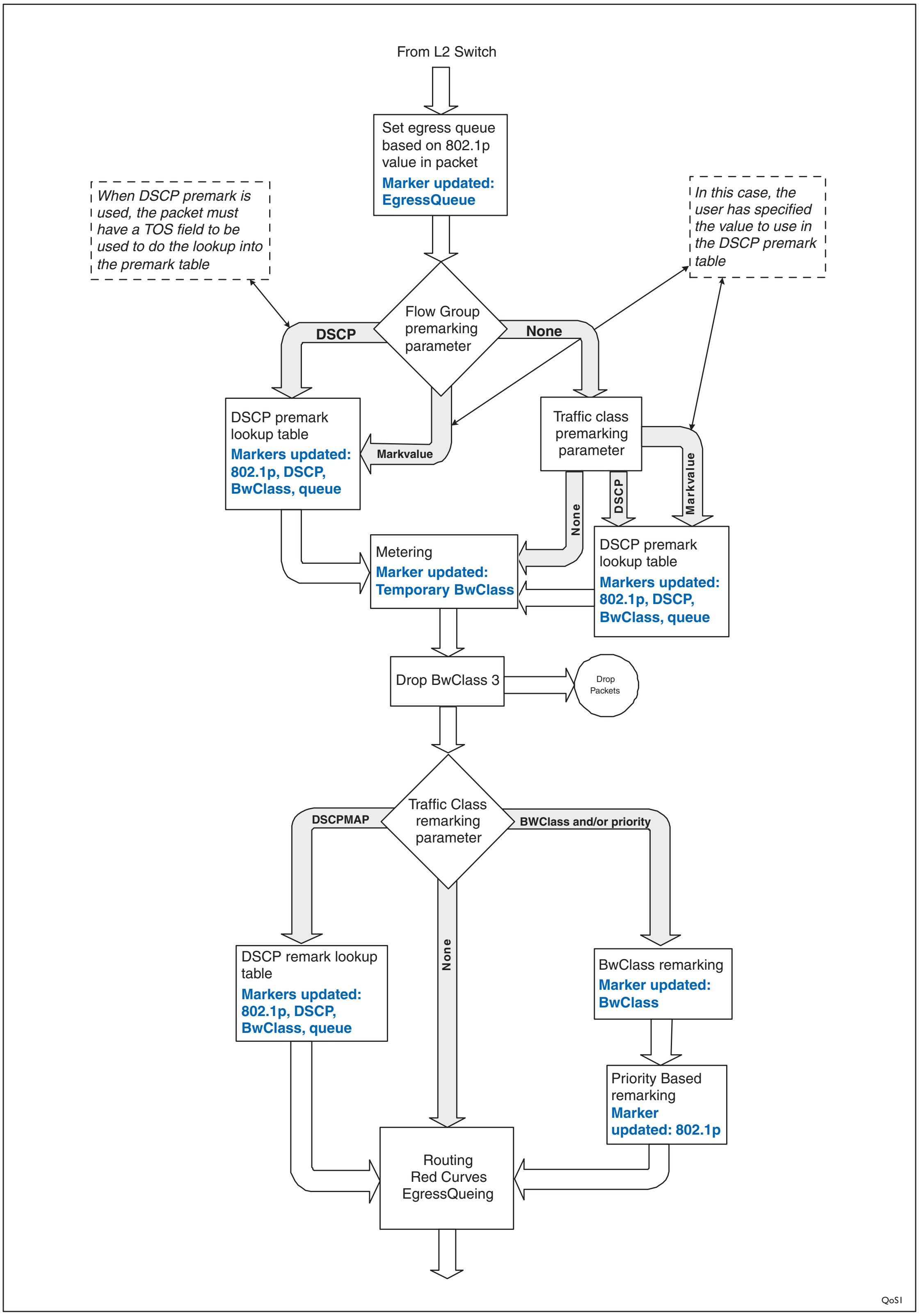
We have already seen that these fields can be modified by the premarking and remarking processes. Downstream devices in a QoS domain will use these values to decide on the appropriate actions to take on incoming packets.

The one case that has not yet been discussed is setting the VLAN Tag User Priority for frames that were untagged at ingress. Recall that untagged frames don't have a VLAN Tag User Priority, so that rather than being assigned to a queue at ingress on the basis of their User Priority, these frames can be assigned to a default queue of the administrator's choice.

If frames that were untagged at ingress, but are being transmitted as tagged packets, have not had a User Priority value assigned to them by any other QoS mechanism, they can have one assigned at egress. The value that is assigned is configured on the basis of the egress queue that the frames are transmitted from.

The diagram in Figure 2 shows the QoS process flow described above.

Figure 2: The QoS process flow


From L2 Switch
Set egress queue based on 802.1p value in packet
Marker updated: EgressQueue
When DSCP premark is used, the packet must have a TOS field to be used to do the lookup into the premark table
In this case, the user has specified the value to use in the DSCP premark table
Flow Group premarking parameter
DSCP
None
DSCP premark lookup table
Markers updated: 802.1p, DSCP, BwClass, queue
Markvalue
Traffic class premarking parameter
None
DSCP
Markvalue
DSCP premark lookup table
Markers updated: 802.1p, DSCP, BwClass, queue
Metering
Marker updated: Temporary BwClass
Drop BwClass 3
Drop Packets
Traffic Class remarking parameter
DSCPMAP
BWClass and/or priority
None
DSCP remark lookup table
Markers updated: 802.1p, DSCP, BwClass, queue
BwClass remarking
Marker updated: BwClass
Priority Based remarking
Marker updated: 802.1p
Routing
Red Curves
EgressQueing
QoS1

Details of the component processes, and how to configure them

Mapping to queue based on tag

There are two choices of the mode in which this mapping can be done. You can either force all packets coming into a given port to ALL be assigned to the same egress queue, or you can let tagged packets be assigned to various different egress queues, depending on the 802.1p values in their tags.

The command that chooses the mode is:

```
set qos port={port-list|all} forcedefqueue={yes|no}
```

If the **forcedefqueue** parameter is set to **yes**, then ALL packets arriving into the port are assigned to the queue that is specified by the command:

```
set qos port={port-list|all} defaultqueue=queue-number
```

In this mode, for every packet that arrives into the port, the Egress Queue marker associated with the packet is set to the value of **defaultqueue**.

If the **forcedefqueue** parameter is set to **no**, then untagged packets arriving into the port are still assigned to the **defaultqueue**.

But, for tagged packets, the switch can assign packets to different egress queues, depending on the value of the 802.1p value contained within the packet's VLAN tag.

The command that controls which egress queue is chosen for each 802.1p value is:

```
set qos prio2queuemap=p0,p1,p2,p3,p4,p5,p6,p7
```

The first value, ***p0***, represents the egress queue that will be assigned to an incoming packet whose VLAN tag has an 802.1p value of 0. ***P1*** represents the egress queue that will be assigned to an incoming packet whose VLAN tag has an 802.1p value of 1; and so on.

So, the 802.1p value in the incoming tag is examined, and the Egress Queue marker associated with the packet is set to the corresponding value that has been configured by this command.

Classification

The process of assigning packets to traffic classes requires a two stage configuration.

First, generic classifiers are used to assign packets to flow groups, e.g.:

```
create classifier=1 tcpdport=80

create qos flow=1

add qos flow=1 class=1
```

Then, flow groups are assigned to a traffic class, e.g.:

```
create qos traffic=1

add qos traffic=1 flow=1-3
```

The reason that there is a hierarchy of flow groups and traffic classes is simply that there are some actions you might want to perform on quite specific categories of traffic (flow groups), and some actions you might want to perform on larger grouping of traffic types (traffic classes).

Premarking

The data structure that drives the premarking process is the premarking DSCPmap table. This is a single global table which can be thought of as a table comprising 64 rows and 3 columns. In each cell of the table there are four items; 802.1p, DSCP, bandwidth class and egress queue, that will be applied to packets. The rows are indexed by the 64 possible DSCP values. The columns are indexed by the three bandwidth class values. This is shown in Table 1.

Table 1: DSCPmap table

BWCLASS **DSCP**	**Class 1**	**Class 2**	**Class 3**

0	802.1p = ...
	DSCP = ...
	bwclass = ...
	queue = ...
		802.1p = ...
		DSCP = ...
		bwclass = ...
		queue = ...
			802.1p = ...
			DSCP = ...
			bwclass = ...
			queue = ...

1	802.1p = ...
	DSCP = ...
	bwclass = ...
	queue = ...
		802.1p = ...
		DSCP = ...
		bwclass = ...
		queue = ...
			802.1p = ...
			DSCP = ...
			bwclass = ...
			queue = ...

2	802.1p = ...
	DSCP = ...
	bwclass = ...
	queue = ...
		802.1p = ...
		DSCP = ...
		bwclass = ...
		queue = ...
			802.1p = ...
			DSCP = ...
			bwclass = ...
			queue = ...

...			
...			
...			
...			

63	802.1p = ...
	DSCP = ...
	bwclass = ...
	queue = ...
		802.1p = ...
		DSCP = ...
		bwclass = ...
		queue = ...
			802.1p = ...
			DSCP = ...
			bwclass = ...
			queue = ...

Values are written into the DSCPmap table using the command:

```
set qos dscpmap=premarking dscp=dscp-list [bwclass=bwclass-list]
    [newdscp=dscp-value] [newbwclass=bandwidth-class]
    [newqueue=queuenumber [newpriority=vlan-priority]
```

The premarking process takes values from this table and uses them to mark the four markers (802.1p, DSCP, bandwidth class, Egress Queue) on packets.

The premarking process must do a table lookup in order to obtain the values with which to mark any given packet. Given that the bandwidth class on every packet entering the premarking process is still at the default value of 1, then the lookup is always going to be in the first column of the table. The DSCP value used to complete the lookup can be chosen in one of two ways:

1. the DSCP value that is already present in the packet, or
2. a user-configured DSCP value that is used for all packets.

Premarking can be specified on a per-flowgroup basis or a per-traffic class basis.

If premarking has been specified on a flow group and on the traffic class to which the flow group belongs, then packets will only go through one premarking process, namely that specified on the flow group.

The specification of premarking parameters is carried out using the command:

```
set qos flowgroup=flowgroup-list [markvalue={dscp-value|none}]
    [premarking={usemarkvalue|usedscp|none}]
```

If `premarking=usemarkvalue` is specified, then the lookup into the DSCPmap table always uses the DSCP value specified by the parameter **markvalue**.

If `premarking=usedcsp` is specified, then the lookup into the DSCPmap table uses the DSCP value that is present in the packet. In this case the parameter **markvalue** is irrelevant.

Metering

Metering is performed on a per traffic class basis.

On a traffic class, there are four parameters that can be used to configure the metering process. The parameters are:

- maxbandwidth
- minbandwidth
- maxburstsize
- minburstsize.

These parameters can be used in two possible combinations:

1. **Single-rate metering**

   This is the combination of maxbandwidth, maxburstsize and minburstsize. With this combination, the algorithm used to determine the temporary bandwidth class to assign to a packet is:

   If the data rate for the traffic class is below maxbandwidth, OR is slightly above maxbandwidth, but the accumulation of total bits that have exceeded the maxbandwidth has not yet reached minburstsize, then the bandwidth class is 1.

   If the data rate for the traffic class is above maxbandwidth, and the accumulation of total bits that have exceeded the maxbandwidth has exceeded minburstsize, but not yet reached maxburstsize then the bandwidth class is 2.

If the data rate for the traffic class is above maxbandwidth, and the accumulation of total bits that have exceeded the maxbandwidth has exceeded maxburstsize then the bandwidth class is 3.

An example of configuring a traffic class to do single-rate metering would be:

```
set qos trafficclass=id-list [maxbandwidth={bandwidth|none}]
   [maxburstsize=burstsize] [minburstsize=burstsize]
```

2. Twin-rate metering

This uses the combination of all four parameters. With this combination, the algorithm used to determine the temporary bandwidth class to assign to a packet is:

If the data rate for the traffic class is below minbandwidth, OR is slightly above minbandwidth, but the accumulation of total bits that have exceeded the minbandwidth has not yet reached minburstsize, then the bandwidth class is 1.

If the data rate for the traffic class is above minbandwidth, and the accumulation of total bits that have exceeded the minbandwidth has exceeded minburstsize, OR the data rate is above maxbandwidth and the accumulation of total bits that have exceeded the maxbandwidth has not exceeded maxburstsize, then the bandwidth class is 2.

If the data rate for the traffic class is above maxbandwidth, and the accumulation of total bits that have exceeded the maxbandwidth has exceeded maxburstsize then the bandwidth class is 3.

An example of configuring aa traffic class to do twin-rate metering would be:

```
set qos trafficclass=id-list [maxbandwidth={bandwidth|none}]
   [minbandwidth={bandwidth|none}] [maxburstsize=burstsize]
   [minburstsize=burstsize]
```

There is only one real configuration difference between single-rate and twin-rate metering. If you want to use single-rate metering, just leave the minbandwidth at the default value of NONE.

Remarking

Remarking works a little differently from premarking.

There is a remarking DSCPmap, which has absolutely the same structure as the premarking DSCPmap, and is configured with a command of the same syntax:

```
set qos dscpmap=premarking dscp=dscp-list [bwclass=bwclass-list]
   [newdscp=dscp-value] [newbwclass=bandwidth-class]
   [newqueue=queuenumber][newpriority=vlan-priority]
```

But, there is another table that is specific to the remarking process, and does not have an analogue in the premarking process. This is the queue-to-priority map, frequently referred to as the contracted name **queue2priomap**. The structure of this table is as shown in Table 2.

Table 2: Queue-to-priority map table

Temporary Bandwidth Class Egress Queue	Class 1	Class 2	Class 3
0	802.1p = ...	802.1p = ...	802.1p = ...
1	802.1p = ...	802.1p = ...	802.1p = ...
2	802.1p = ...	802.1p = ...	802.1p = ...
3	802.1p = ...	802.1p = ...	802.1p = ...
4	802.1p = ...	802.1p = ...	802.1p = ...
5	802.1p = ...	802.1p = ...	802.1p = ...
6	802.1p = ...	802.1p = ...	802.1p = ...
7	802.1p = ...	802.1p = ...	802.1p = ...

Values can be entered into this table using the command:

```
set qos queue2priomap queue=queue-list [bwclass=bwclasslist]
   newpriority=vlan-priority
```

You have the choice of doing remarking in one (and only one) of the following ways:

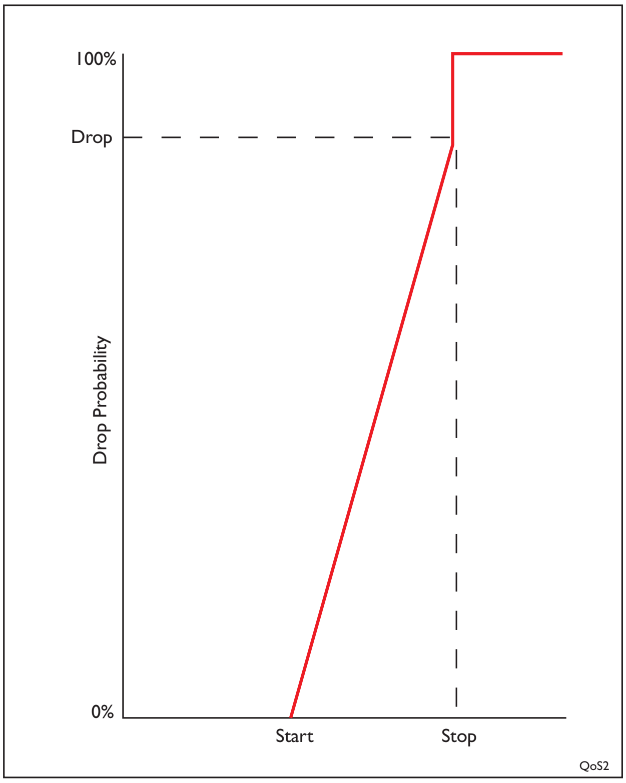
- Doing a lookup into the remarking DSCPmap table, and assigning new values for all four markers (802.1p, DSCP, egress queue, bandwidth class). This is chosen by using the option **set qos trafficclass=<class> remarking=usedscpmap**.
- Doing a lookup into the queue2priomap and assigning a new 802.1p value to the packet. This is chosen by using the option **set qos trafficclass=<class> remarking=priority**.
- Marking the packet's bandwidth class with the temporary bandwidth class that was calculated by the metering process. This is chosen by using the option **set qos trafficclass=<class> remarking=bwclass**.
- Doing a lookup into the queue2priomap and assigning a new 802.1p value to the packet, AND marking the packet's bandwidth class with the temporary bandwidth class that was calculated by the metering process. This is chosen by using the option **set qos trafficclass=<class> remarking=prio+bwclass**.

Queue shaping - RED curves

There are a number of RED curves that can be configured on the switch, so there is quite a structure in the way in which these curves are referenced. It is important to get it clear in your mind how this structure all fits together.

The fundamental entity in this structure is a single set of **Start-Stop-Drop** values. These three values define a "curve" such as the one shown in Figure 3.

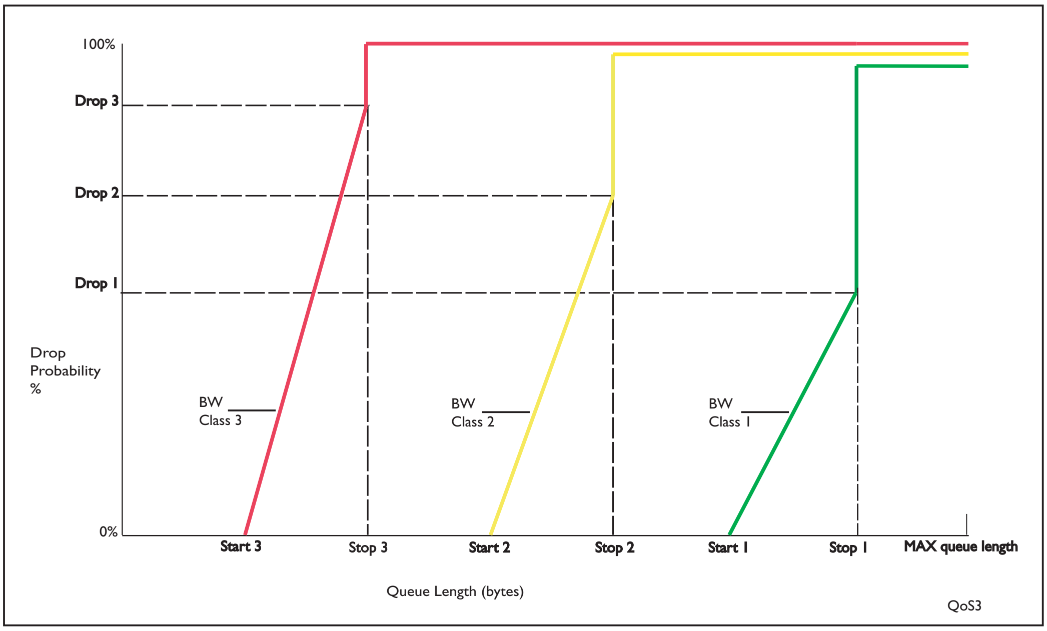
Figure 3: Start-Stop-Drop curve



- **Start** defines the length that the queue must reach before the packets start being dropped.
- **Stop** defines the length that the queue must reach before the shaper stops dropping randomly, and just drops all further packets.
- **Drop** defines the percentage of packets that are being dropped at the point when the length of the queue reaches the Stop value. Effectively, Drop defines how quickly the rate of dropping packets must increase as the queue length grows from the Start value to the Stop value.

These fundamental curves are collected into RED curve groups, as shown in Figure 4. A group is a collection of three curves, one for each of the three possible bandwidth classes.

Figure 4: A RED curve group



Additionally, one other parameter is defined on a RED curve group. This parameter is the Averaging time. The averaging time influences how the queue length is calculated. If the averaging time is 0, then the queue length value that the shaper uses in its calculation of whether a certain packet should be dropped is the exact current length of the queue. But, if you increase the averaging time, then the shaper starts calculating an average length of the queue over a certain time period, and uses this averaged value in its "should-I-drop-the-packet" calculation.

Eight RED curve groups are collected together to make a RED curve set. A RED curve set contains one RED curve group for each of the eight egress queues on a given port.

On the AT-8948 switch, it is possible to define four such RED curve sets, and allocate these sets across the ports of the switch.

The parameters that define a RED curve set are configured as follows:

```
set qos red=red-id [averaging=averaging-factor]
    [description=description] [queue=queue-list] [start1=start]
    [stop1=stop] [drop1=probability] [start2=start] [stop2=stop]
    [drop2=probability] [start3=start] [stop3=stop] [drop3=probability]
```

The parameters Start*x*/Stop*x*/Drop*x* define the curve for bandwidth class *x*.

Scheduling

On any given port, the eight queues can be allocated to the three scheduling groups using the command

```
set qos port egressqueue=queue-list scheduler={strict|wrr1|wrr2}
```

Of course, any given queue on a particular port can only be a member of one of the scheduling groups.

By default, all the queues on a port are initially members of the strict scheduling group.

The relative weights for queues that are members of a weighted round-robin group are set using the command

```
set qos port egressqueue=queue-list wrrweight=queue-weight
```

Other items worthy of honourable mention

Egress bandwidth limiting

The total bandwidth that can be transmitted from a set of egress queues on a port is configurable using the command:

```
set qos port egressqueue=queue-list maxbandwidth=bandwidth
```

This means the maxbandwidth is not necessarily set for the port as a whole, but for a set of the egress queues on the port.

The bandwidth limits can only be specified in multiples of 650 kbps. Whatever value you configure will be rounded up to the nearest multiple of 650 kbps.

It is important to understand the relationship between the **set qos port egressqueue maxbandwidth** command and the **set switch port egresslimit** command. These two commands actually control two different aspects of the egress scheduling process.

The model to consider is this: the process of putting packets onto the wire "pulls" packets out of egress queues and puts them out the port. But, the queues can actually resist a "pull" and effectively tell the "pulling" process 'sorry, I have hit my bandwidth limit and cannot give you any packets right now'.

The **set switch port egresslimit** command sets the rate at which the port "pulls" packets from the egress queues. The **set qos port egressqueue maxbandwidth** command sets the rate at which a queue will allow packets to be pulled out of it.

The **set switch port egresslimit** command sets the maximum limit at which data can leave the port.

The **set qos port egressqueue maxbandwidth** command can do two things:

1. ensure that data is able to leave some queues more quickly than others.
2. possibly set a maximum limit on the egress rate from the port. If the sum of the egress rates on all the queues is less than the total egresslimit set by the **set switch port egresslimit** command, then in fact the maximum rate at which packets can exit the port will be the sum of the egress rates on the queues.

The equitable treatment of the default traffic class

The default traffic class can be given all the same processing as specifically configured traffic classes. Packets that do not match any configured traffic class can still be subjected to premarking, metering and remarking. The parameters that control these processes for the default traffic class are actually configured on the QoS policy, using the command:

```
set qos policy=id-list [dtcdropbwclass3={yes|no}]
    [dtcignorebwclass={yes|no}][dtcmaxbandwidth={bandwidth|none}]
    [dtcmaxburstsize=burstsize][dtcminbandwidth={bandwidth|none}]
    [dtcminburstsize=burstsize][dtcpremarking={usemarkvalue|usedscp|
    none] [dtcremarking={usedscpmap|bwclass|priority|prio+bwclass|
    none}] [markvalue={dscp-value|none}]
```

Setting the length of Egress Queues

The command **set qos port egress queue length** enables you to set the maximum lengths of individual egress queues on a given port.

But, it is very important to note that the total number of packets that can be queued on a given port has a FIXED maximum of 128 packets. Even if the sum of the individual maximum queue lengths is greater than 128, the total number of packets able to queue on a port is 128.

That is, if you add up the number of packets sitting in each of the eight queues on a given port, the result of that addition will always be less then or equal to 128.

You could set the maximum lengths of some queues to less than 128, so that those queues can never grow greater than a certain length, even if the total number of packets queued on the port has not yet hit the 128 limit.

Note, also, that the length of an egress queue can only be set to values that are multiples of 16. If you set a value that is not a multiple of 16, the value will be rounded UP to the nearest multiple of 16.

If you want to ensure that no queue can grow to a size that is causing other queues to have to drop packets too early, then you must set the lengths of the queues to values where the sum of the queue lengths is 128. In fact, the only sensible way to achieve that is to set all the queues to a length of 16 (certainly, you could set some queues to length 0 and others to 32, but that effectively just removes the 0-length queues from the switch).

Monitoring the state of the egress queues

Using the command **show qos port counter egressqueue**, you can see the number of packets currently sitting in each of the eight egress queues on a particular port.

Figure 5: Example output from the show qos port counter egressqueue command

```
Port 1 Egress Queue Counters:
  Port queue length .........      79 (maximum 128)
  Egress queue length:
    Queue 0 .................      10 (maximum 128)
    Queue 1 .................      0 (maximum 128)
    Queue 2 .................      0 (maximum 128)
    Queue 3 .................      24 (maximum 128)
    Queue 4 .................      0 (maximum 128)
    Queue 5 .................      0 (maximum 128)
    Queue 6 .................      2 (maximum 128)
    Queue 7 .................      43 (maximum 128)
```

The value reported by "Port queue length" will always be the sum of the current individual queue lengths.

Monitoring traffic class statistics on a port

The switch can be put into a mode whereby it will collect and display statistics relating to the operation of the traffic classes on a particular port.

To put the switch into this mode, use the command **set switch enhancedmode=qoscount**. Note, this command does not take effect until the next time the switch is restarted (either a warm restart with 'restart switch' or a full reboot). The command is not stored in the configuration script of the switch, so it is not necessary to perform the 'create config' command before restarting the switch. In fact, the command is automatically stored into a separate file called switch.ini that is used to initialise the switching chip at bootup.

When the switch is in this mode, it is possible to view the traffic class statistics on a port using the command **show qos port count trafficclass**.

Figure 6: Example output from the show qos port count trafficclass command

```
Manager > show qos port=1 count traff

QOS Counter Information

Port 1:
  Policy: 1
  Traffic Class 1:
    Aggregate Bytes .............                     0
    BwConformanceClass1 bytes ....                    0
    BwConformanceClass2 bytes ....                    0
    BwConformanceClass3 bytes ....                    0
    Dropped bytes ...............                     0

  Traffic Class 7:
    Aggregate Bytes .............                     0
    BwConformanceClass1 bytes ....                    0
    BwConformanceClass2 bytes ....                    0
    BwConformanceClass3 bytes ....                    0
    Dropped bytes ...............                     0

  Default Traffic Class:
    Aggregate Bytes .............                     0
    BwConformanceClass1 bytes ....                    0
    BwConformanceClass2 bytes ....                    0
    BwConformanceClass3 bytes ....                    0
    Dropped bytes ...............                     0
```

One word of caution regarding this enhanced mode.

When the switch is in QosCount enhanced mode, some of the switch-chip hardware resources that are normally available to QoS processing of packets are actually diverted to the task of storing collected statistics. The QoS capability of the switch is reduced when it is in this mode. Hence, it should not be used in normal operation. It is a useful mode for specific occasions when you want to monitor the QoS operation on your switch, but is not recommended as the standard mode in which to run the switch.