

AlliedWare Plus™ OS

How To Configure QoS to Conform to Standard Marking Schemes

Introduction

This How To Note describes how to deploy a QoS solution across an entire network. It explains how to define per-hop behaviours (PHBs) that each switch in the network should perform, and how to ensure that the switches uniformly interpret the QoS information carried in packets. This means you can achieve a predictable performance for different traffic types across the whole network.

Packets that enter the network's edge may carry no QoS information. If so, the edge switch places such information into the packets before transmitting them to the next node. Thus, QoS information is preserved between nodes within the network and the nodes know what treatment to give each packet.

Contents

Introduction	1
Contents	1
Which products and software versions does this How To Note apply to?	2
Related How To Notes	3
Marking QoS information into packets	4
Layer 2 CoS/802.Ip	4
Layer 3 TOS and DSCP	4
Interpreting the values	6
802.Ip	6
DSCP	6
Simple 8-level prioritisation	6
More sophisticated per-hop behaviours	7
Implementation example I: Mapping CoS, TOS, and DSCP to queues	10
Configuration options	10

Implementation example 2: Implementing per-hop behaviours in a network	12
Example network diagram	12
Configuration common to all the switches	14
Premark table	14
Remarking table	15
Egress queue scheduling	17
RED curves	18
Configuration of access switches	20
Configure class maps for VoIP traffic to and from the SIP server	20
Configure class maps for other traffic to the server farm	20
Configure class maps for other traffic from the server farm	21
Configure a policy map for traffic to the servers	21
Configure a policy map for traffic from the servers	22
Apply the policy maps to the ports	23
Configuration of server farm switch	24
Configure class maps for VoIP traffic to and from the SIP server	24
Configure class maps for other traffic to the server farm	24
Configure class maps for other traffic from the server farm	24
Configure a policy map for traffic to the servers	25
Configure a policy map for traffic from the servers	26
Apply the policy maps to the ports	26
Configuration of core switch	27
Configure class maps	27
Configure a policy map	28
Apply the policy maps to the ports	28

Which products and software versions does this How To Note apply to?

This configuration applies to AlliedWare Plus software version 5.2.1-0.1 and above, for the following Allied Telesis switches:

- SwitchBlade x908
- x900-12XT/S
- x900-24 series

Related How To Notes

You also may find the following AlliedWare Plus How To Notes useful:

- *Overview of Quality of Service Features on x900-12, x900-24, and SwitchBlade x908 Switches*
- *How To Configure QoS on x900-24, x900-12, and SwitchBlade x908 Series Switches*
- *How To Configure Hardware Filters on SwitchBlade x908, x900-12XT/S, and x900-24 Series Switches*

AlliedWare Plus How To Notes are available from www.alliedtelesis.com/resources/literature/howto_plus.aspx.

Also, you can configure standard schemes on all AlliedWare software versions for the AT-8948, AT-9900 series, AT-9900s series, and x900 series switches. For details, see the AlliedWare Note *How To Configure QoS To Conform To Standard Marking Schemes*. This Note is available from www.alliedtelesis.com/resources/literature/howto.aspx.

Also, you may find the *Advanced QoS White Paper* useful, especially for detailed explanations of the policing mechanisms. This paper is available from the White Papers library at www.alliedtelesis.com/resources/literature/literature.aspx?id=3.

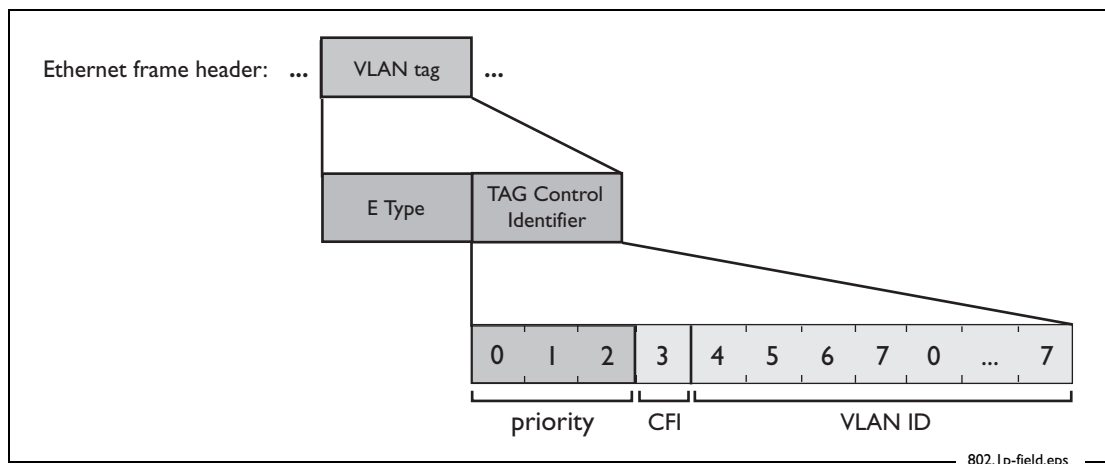
Marking QoS information into packets

There are three options for marking QoS information into packets:

- 802.1p Class of Service (CoS) priority field within the VLAN tag of tagged Ethernet frames (Layer 2)
- IP Precedence from Type of Service (TOS) field (Layer 3)
- Differentiated Services (DiffServ) Code Point (DSCP) (Layer 3)

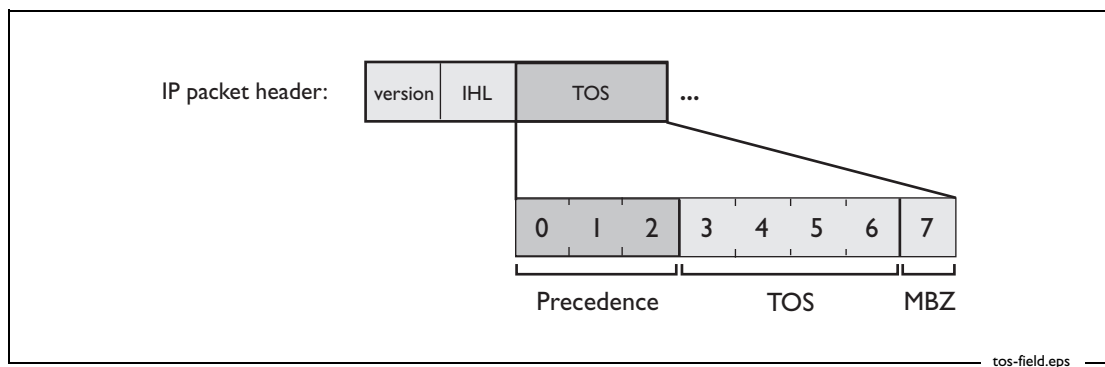
Layer 2 CoS/802.1p

Layer 2 devices can use the 3-bit (binary 000-111; decimal 0-7) CoS field in 802.1p tagged frames to carry priority information. This field is also called the 802.1p field.

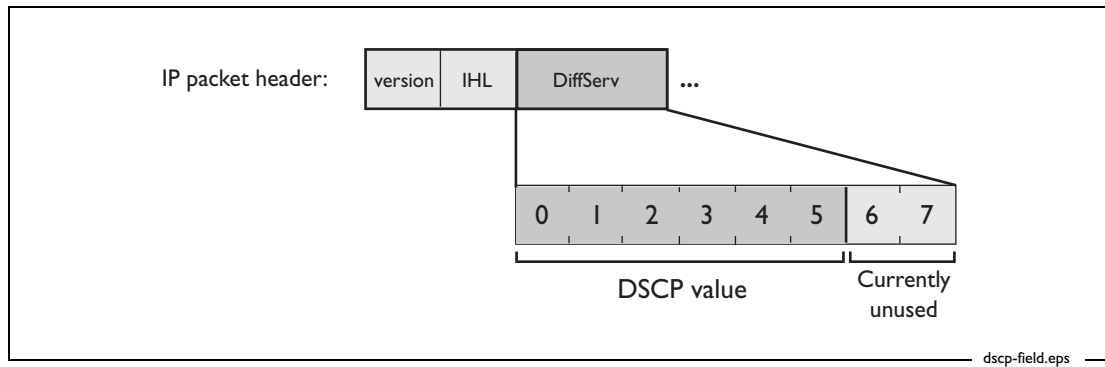


Layer 3 TOS and DSCP

Layer 3 devices can also use the 3-bit (binary: 000-111; decimal: 0-7) IP Precedence field from the TOS byte of the IP header. This is defined in RFC 791 and RFC 1349.



Alternatively, Layer 3 devices can use the Differentiated Services (DiffServ or DS) field from RFC 2474 and RFC 2475. This field is the TOS octet, redefined. The following figure shows the location of the DiffServ field in the IP header.



Interpreting the values

802.Ip

Interpreting the 802.Ip values is quite straightforward: the 8 values define 8 different levels of priority.

Two common mappings of 802.Ip value to priority level are:

		Low priority	—————>	high priority queue				
Map 1	1	0	2	3	4	5	6	7
Map 2	0	1	2	3	4	5	6	7

In the first mapping, an 802.Ip value of 0 goes into the second-lowest priority queue instead of the lowest priority queue. Unclassified default traffic generally has an 802.Ip value of 0, so this mapping reduces the probability of dropping default traffic.

DSCP

Interpreting DSCP values is a little more involved, because there are 64 possible values of DSCP, not just 8.

There are different schemes in common use for interpreting DSCP values:

- ["Simple 8-level prioritisation" on page 6](#)
- ["More sophisticated per-hop behaviours" on page 7](#)

Simple 8-level prioritisation

In this scheme, the DSCP values are grouped into 8 groups, based upon their first 3 bits. All the values in any given group share the same set of first 3 bits. These first 3 bits, of course, correspond to the IP Precedence field in the older TOS definition of the Layer 3 QoS byte.

These 8 groups of values can be mapped to 8 priority levels (i.e. 8 egress queues) as shown in the table below. Note that the last two values in the table (Internetwork and Network control) should be reserved for "Control Plane" traffic such as routing protocols and network management.

IP Precedence	DSCP range	CoS	Queue
Routine (Default)	000 (0) - 000111 (7)	0	0
Priority	001 (1) - 001111 (15)	1	1
Immediate	010 (2) - 010111 (23)	2	2
Flash	011 (3) - 011111 (31)	3	3
Flash Override	100 (4) - 100111 (39)	4	4
Critical	101 (5) - 101111 (47)	5	5
Internetwork control	110 (6) - 110111 (55)	6	6

IP Precedence	DSCP range	CoS	Queue
Network control	111 (7) 111000 (56) - 111111 (63)	7	7

More sophisticated per-hop behaviours

Some RFCs have been written that define mappings of DSCP values to quite specific ways in which the network devices should treat packets.

They define three distinct types of treatment (referred to as *per-hop behaviours*, PHBs):

- Expedited Forwarding (EF) PHB
- Assured Forwarding (AF) PHB
- Default PHB

Relevant RFCs include 2597, 3140, 3246, 3247, 3260, and 4594.

Expedited Forwarding (EF) PHB

This traffic is given highest priority. Packets being given this treatment are prioritized ahead of all other types of data.

This is very suitable for traffic that requires low delay, low loss, and low jitter. Real-time services like voice and video require this kind of treatment.

If too much traffic is put into the top-priority queue, of course, then the packets no longer really get the advantage of being top priority; they end up sitting in the queue, and so suffering from delay and jitter.

Also, if too many packets are being put into the top-priority queue, then the lower-priority queues become starved.

For these reasons, typical networks will limit EF traffic to no more than 30%—and often much less—of the capacity of a link.

Assured Forwarding (AF) PHB

The AF PHB is characterized by a Committed Information Rate (CIR) and Excess Information Rate (EIR).

The network should be able to assure the delivery of all AF traffic up to the CIR. Once the rate of AF traffic exceeds the CIR, then the network will attempt to deliver the extra traffic up to EIR. But, if the rate of AF traffic exceeds the EIR, then the traffic beyond that rate will almost invariably be dropped.

AF traffic is divided into 4 different classes (effectively 4 different priorities, but more about that below).

Within each class, the traffic is divided into 3 categories:

1. Traffic up to the CIR
2. Traffic that exceeded the CIR, but not the EIR
3. Traffic that exceeded the EIR

These three categories are referred to as *drop precedences*. The category that a packet falls into determines how likely it is to get dropped when congestion occurs.

Four classes, each with 3 drop precedences, defines the following 12 AF values:

	Green (low drop)	Yellow (medium drop)	Red (high drop)
AF Class1	AF11	AF12	AF13
AF Class2	AF21	AF22	AF23
AF Class3	AF31	AF32	AF33
AF Class4	AF41	AF42	AF43

If congestion occurs between classes, the traffic in the higher class is given priority. Rather than using strict priority queueing, more balanced queue servicing algorithms such as fair queueing or weighted fair queueing are likely to be used. This prevents high-priority queues from completely starving lower-priority queues—for example, it can ensure that if video traffic is over-subscribed, database traffic still gets some bandwidth.

If congestion occurs **within** a class, the packets with the higher drop precedence are discarded first. To prevent issues associated with tail drop, the random early detection (RED) algorithm is usually employed to decide which packets to drop.

The drop precedence is assigned to packets by a policing process in the access switch where the packets enter the network. Typically, all traffic assigned to a class is initially given a low drop precedence. As the traffic rate exceeds the CIR, and then the EIR, the meter will increase the drop precedence of packets that exceed the threshold.

Default PHB This is just best-effort forwarding of the lowest-priority traffic. Packets being given this treatment get a very limited amount of the available egress bandwidth in times of congestion.

PHB mapping table The DSCP values corresponding to the EF, AF, and Default categories are shown in the table below.

			Queue	
Default PHB			00000000 0	
AF Class1	AF11	AF12	AF13 1	
	001010 (10)	001100 (12)	001110 (14)	
AF Class2	AF21	AF22	AF23 2	
	010010 (18)	010100 (20)	010110 (22)	
AF Class3	AF31	AF32	AF33 3	
	011010 (26)	011100 (28)	011110 (30)	
AF Class4	AF41	AF42	AF43 4	
	100010 (34)	100100 (36)	100110 (38)	
EF			EF 5	
			101110 (46)	
Green (low drop)			Yellow (medium drop)	Red (high drop)

Drop precedence versus priority

It is important to appreciate the difference between drop precedence and priority.

Drop precedences determine which packets the switch first considers dropping if traffic policing indicates that the switch is congested. A policing and colouring process calculates precedences dynamically as packets pass through the switch.

Traffic with the highest drop precedence may be referred to by any of the following terms: *high drop*, *red*, *exceeded EIR*, or *bandwidth class 3 (bwc 3)*.

Some network administrators decide that for some types of traffic in the network, all traffic in the red zone should be dropped, regardless of congestion levels. For example, this may reflect the level of service a customer has paid for.

Priorities define which traffic is more important and therefore will go in a higher-priority or higher-weight egress queue. If a frame has a higher priority, that just means the switch puts it in a higher priority egress queue that is handled first, while any frames in lower priority queues wait. The switch silicon has 8 egress queues available, each of ascending priority.

Implementation example I: Mapping CoS, TOS, and DSCP to queues

In this example, the switch is configured to map layer 2 CoS (802.Ip) values to egress queues, and to map layer 3 DSCP values to egress queues according to the simple 8-level prioritisation scheme described on [page 6](#).

Configuration options

► Map CoS/802.Ip priorities to egress queues for incoming **tagged** frames

Use the following commands to set the mapping of incoming VLAN Tag User Priorities to the egress queues, for incoming packets that include a VLAN tag header.

```
mls qos enable
mls qos map cos-queue 1 to 3
mls qos map cos-queue 3 to 0
```

The above commands send incoming packets with a CoS value of 1 to egress queue 3 and packets with CoS value 3 to egress queue 0. All other incoming CoS values will be mapped to their default egress queue.

You can verify all the CoS to egress queue mappings, by using the command:

```
awplus#show mls qos maps cos-queue
COS-TO-QUEUE-MAP:
COS :    0    1    2    3    4    5    6    7
-----
QUEUE:  2    3    1    0    4    5    6    7
```

By default, the switch uses a commonly deployed cos-to-queue mapping as shown below. As mentioned in "[Interpreting the values](#)" on [page 6](#), unclassified default traffic generally has an 802.Ip value of 0, so this default mapping reduces the probability of dropping default traffic.

The default mapping is:

```
COS-TO-QUEUE-MAP:
COS :    0    1    2    3    4    5    6    7
-----
QUEUE:  2    0    1    3    4    5    6    7
```

► Map 802.Ip priorities to egress queues for incoming **untagged** frames

Untagged frames arriving at an ingress port will get assigned to a default egress queue. Also, if the frame's egress port is a trunk port (and therefore uses tagged frames), then these untagged frames will also be marked with a default CoS value before egress.

You can change the default egress queue and the CoS value assigned at the ingress port, by using the following commands in interface mode for the desired ports:

```
awplus(config-if)#mls qos queue <queue-number>
awplus(config-if)#mls qos cos <cos-value>
```

► Map DSCP values to egress queues

The mapping of DSCP values to egress queues is achieved using the Pre-marking table. For the simple 8-level prioritisation scheme shown in the table on [page 6](#), DSCP values 0-7 map to queue 0; DSCP values 8-15 map to queue 1, on so on.

The following commands do three things:

- map each DSCP range to the correct queue. Note that queue 7 is the highest priority queue
- define a new DSCP value that will be written into the packets (basically, all DSCP values in an 8-value range will be simply replaced by the lowest value of that range)
- define the CoS/802.1p value that will be written into frames if they are sent out tagged at egress.

For clarity, the following commands include all parameters, even though some use the parameters' default values. When you use **show run** to display the configuration, parameters that use the default values will not be displayed.

```
mls qos map mark-dscp 0 to new-dscp 0 new-cos 0 new-queue 0
mls qos map mark-dscp 1 to new-dscp 0 new-cos 0 new-queue 0
mls qos map mark-dscp 2 to new-dscp 0 new-cos 0 new-queue 0
mls qos map mark-dscp 3 to new-dscp 0 new-cos 0 new-queue 0
mls qos map mark-dscp 4 to new-dscp 0 new-cos 0 new-queue 0
mls qos map mark-dscp 5 to new-dscp 0 new-cos 0 new-queue 0
mls qos map mark-dscp 6 to new-dscp 0 new-cos 0 new-queue 0
mls qos map mark-dscp 7 to new-dscp 0 new-cos 0 new-queue 0

mls qos map mark-dscp 8 to new-dscp 8 new-cos 1 new-queue 1
mls qos map mark-dscp 9 to new-dscp 8 new-cos 1 new-queue 1
mls qos map mark-dscp 10 to new-dscp 8 new-cos 1 new-queue 1
mls qos map mark-dscp 11 to new-dscp 8 new-cos 1 new-queue 1
mls qos map mark-dscp 12 to new-dscp 8 new-cos 1 new-queue 1
mls qos map mark-dscp 13 to new-dscp 8 new-cos 1 new-queue 1
mls qos map mark-dscp 14 to new-dscp 8 new-cos 1 new-queue 1
mls qos map mark-dscp 15 to new-dscp 8 new-cos 1 new-queue 1

mls qos map mark-dscp 16 to new-dscp 16 new-cos 2 new-queue 2
mls qos map mark-dscp 17 to new-dscp 16 new-cos 2 new-queue 2
mls qos map mark-dscp 18 to new-dscp 16 new-cos 2 new-queue 2
mls qos map mark-dscp 19 to new-dscp 16 new-cos 2 new-queue 2
mls qos map mark-dscp 20 to new-dscp 16 new-cos 2 new-queue 2
mls qos map mark-dscp 21 to new-dscp 16 new-cos 2 new-queue 2
mls qos map mark-dscp 22 to new-dscp 16 new-cos 2 new-queue 2
mls qos map mark-dscp 23 to new-dscp 16 new-cos 2 new-queue 2
.
.
.
```

```
mls qos map mark-dscp 56 to new-dscp 56 new-cos 7 new-queue 7
mls qos map mark-dscp 57 to new-dscp 56 new-cos 7 new-queue 7
mls qos map mark-dscp 58 to new-dscp 56 new-cos 7 new-queue 7
mls qos map mark-dscp 59 to new-dscp 56 new-cos 7 new-queue 7
mls qos map mark-dscp 60 to new-dscp 56 new-cos 7 new-queue 7
mls qos map mark-dscp 61 to new-dscp 56 new-cos 7 new-queue 7
mls qos map mark-dscp 62 to new-dscp 56 new-cos 7 new-queue 7
mls qos map mark-dscp 63 to new-dscp 56 new-cos 7 new-queue 7
```

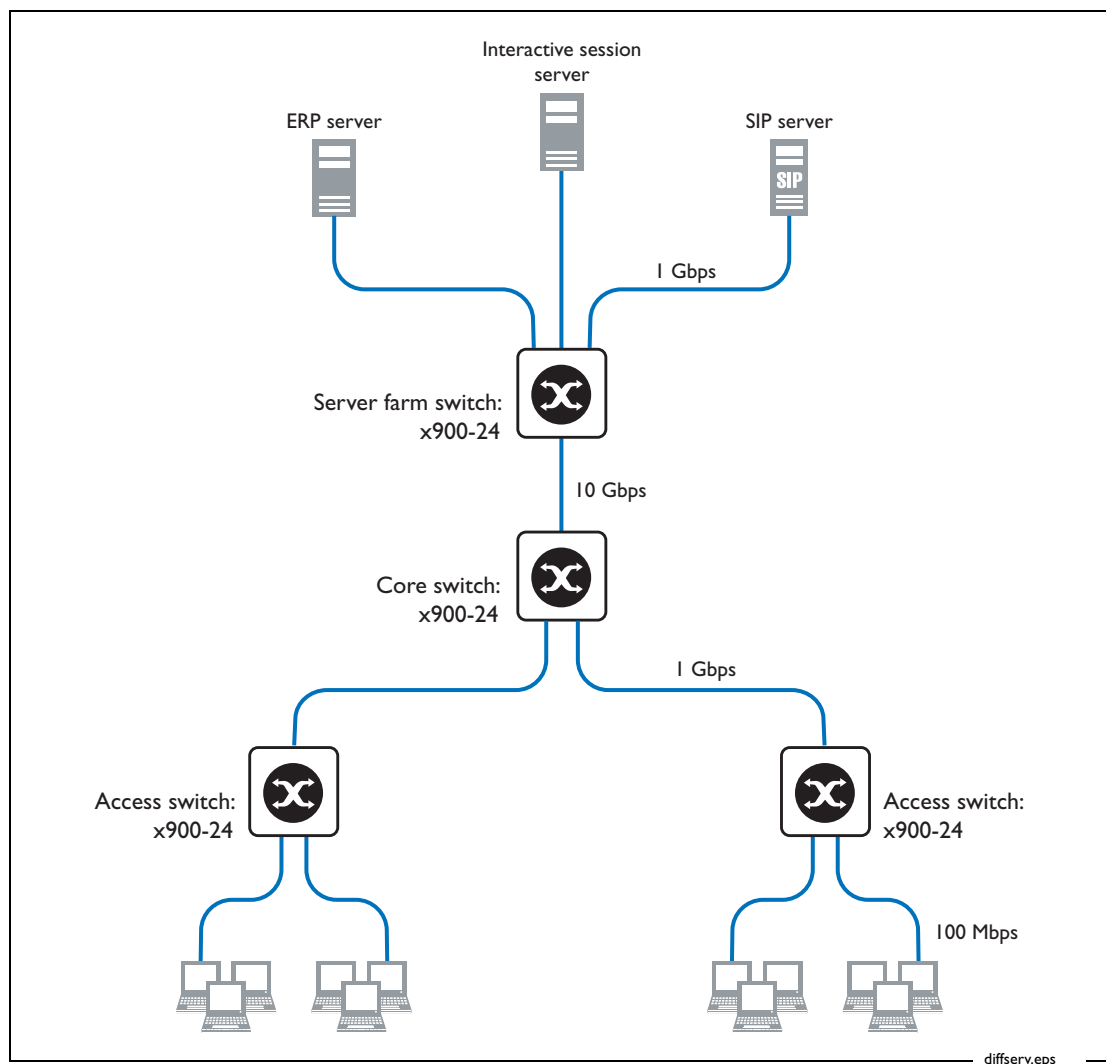
Refer to the prioritisation table ["Simple 8-level prioritisation"](#) on page 6 for help on interpreting these tables.

Implementation example 2: Implementing per-hop behaviours in a network

In this example, we consider a simple network with a set of access switches, and a couple of core/aggregation switches. On this network, we will implement the default/AF/EF PHB scheme (shown in the table on [page 8](#)). The access switches will mark packets and the core/aggregation switches will apply QoS to packets based on those marked values.

Another term for a network like this is a *Diffserv domain*.

Example network diagram



Traffic in the network is classified as shown in the following table. The table shows the DSCP and CoS values for each type of traffic, and the egress queue that each type of traffic will be assigned to.

Application	DSCP	CoS	Queue
Management (OSPF, PIM, STP, BGP, SNMP, etc.)	56	7	7
Voice (stream)	EF(46)	5	5
Interactive (remote sessions on server)	AF41(34)	4	4
Voice (call signalling)	AF31(26)	3	3
Critical database (ERP)	AF21(18)	2	2
Default traffic	0	0	0

Note that this network has no traffic assigned to AF11. The command set in the following sections includes the commands for AF11, for demonstration purposes.

Configuration common to all the switches

The core, server farm, and access switches have the same settings for the following:

- "Premark table" on page 15
- "Remarking table" on page 16
- "Egress queue scheduling" on page 18
- "RED curves" on page 19

As well as these common settings, you need to add switch-specific class-map and policy settings. These are in:

- "Configuration of access switches" on page 21
- "Configuration of server farm switch" on page 25
- "Configuration of core switch" on page 29

Premark table

This premarking table implements the scheme illustrated in the more detailed PHB mapping table introduced on [page 8](#) and repeated below:

				Queue
Default PHB			00000000	0
AF Class1	AF11	AF12	AF13	1
	001010 (10)	001100 (12)	001110 (14)	
AF Class2	AF21	AF22	AF23	2
	010010 (18)	010100 (20)	010110 (22)	
AF Class3	AF31	AF32	AF33	3
	011010 (26)	011100 (28)	011110 (30)	
AF Class4	AF41	AF42	AF43	4
	100010 (34)	100100 (36)	100110 (38)	
EF			EF	5
			101110 (46)	
Green (low drop)			Yellow (medium drop)	Red (high drop)

For example, traffic from AF21 arrives at the switch with a DSCP of 18 and the switch puts it into bandwidth class 1 (green) and queue 2, and puts the value 2 into the 802.1p field of tagged packets.

Comments are marked by an ! at the start of the line.

```
mls qos enable
! Default
mls qos map mark-dscp 0 to new-bandwidth-class red new-queue 0 new-cos 0
```

```

! AF11
mls qos map mark-dscp 10 to new-bandwidth-class green new-queue 1 new-cos 1
! AF12
mls qos map mark-dscp 12 to new-bandwidth-class yellow new-queue 1 new-cos 1
! AF13
mls qos map mark-dscp 14 to new-bandwidth-class red new-queue 1 new-cos 1

! AF21
#mls qos map mark-dscp 18 to new-bandwidth-class green new-queue 2 new-cos 2
! AF22
mls qos map mark-dscp 20 to new-bandwidth-class yellow new-queue 2 new-cos 2
! AF23
mls qos map mark-dscp 22 to new-bandwidth-class red new-queue 2 new-cos 2

! AF31
mls qos map mark-dscp 26 to new-bandwidth-class green new-queue 3 new-cos 3
! AF32
mls qos map mark-dscp 28 to new-bandwidth-class yellow new-queue 3 new-cos 3
! AF33
mls qos map mark-dscp 30 to new-bandwidth-class red new-queue 3 new-cos 3

! AF41
mls qos map mark-dscp 34 to new-bandwidth-class green new-queue 4 new-cos 4
! AF42
mls qos map mark-dscp 36 to new-bandwidth-class yellow new-queue 4 new-cos 4
! AF43
mls qos map mark-dscp 38 to new-bandwidth-class red new-queue 4 new-cos 4

! EF
mls qos map mark-dscp 46 to new-bandwidth-class green new-queue 5 new-cos 5

! Network management and control
mls qos map mark-dscp 56 to new-bandwidth-class green new-queue 7 new-cos 7

```

Remarking table

The remarking table is a little more complex because remarking happens after bandwidth policing. The switch remarks packets' DSCP, queue and priority, on the basis of the DSCP it arrives with and considering its metered bandwidth class. This implementation is called colour-aware mode.

For example, if traffic from AF21 arrives at the switch with a DSCP of 18 and the policing process finds that it slightly exceeds its bandwidth allocation (bandwidth class 2, yellow), then it gets given a new DSCP of 20 and put into queue 2 with an 802.1p value of 2. This moves it into AF22.

The remark values are based on the incoming DSCP, to stop the switch from over-riding previous switches' marking of packets. If a previous switch found a packet to be non-conformant or semi-conformant, this switch cannot mark that packet as conformant.

Note that there is no need to remark default traffic. In times of congestion, we want the switch to simply drop most default traffic because it has the lowest priority, not to shape it to a CIR.

AFI remarking

```

! AF11
mls qos map policed-dscp 10 bandwidth-class green to new-dscp 10
new-bandwidth-class green new-queue 1 new-cos 1
mls qos map policed-dscp 10 bandwidth-class yellow to new-dscp 12
new-bandwidth-class yellow new-queue 1 new-cos 1
mls qos map policed-dscp 10 bandwidth-class red to new-dscp 14
new-bandwidth-class red new-queue 1 new-cos 1

```

```

! AF12
mls qos map policed-dscp 12 bandwidth-class green to new-dscp 12
  new-bandwidth-class yellow new-queue 1 new-cos 1
mls qos map policed-dscp 12 bandwidth-class yellow to new-dscp 12
  new-bandwidth-class yellow new-queue 1 new-cos 1
mls qos map policed-dscp 12 bandwidth-class red to new-dscp 14
  new-bandwidth-class red new-queue 1 new-cos 1

! AF13
! A previous switch saw this frame as red, so we cannot mark it as conformant.
mls qos map policed-dscp 14 bandwidth-class green to new-dscp 14
  new-bandwidth-class red new-queue 1 new-cos 1
mls qos map policed-dscp 14 bandwidth-class yellow to new-dscp 14
  new-bandwidth-class red new-queue 1 new-cos 1
mls qos map policed-dscp 14 bandwidth-class red to new-dscp 14
  new-bandwidth-class red new-queue 1 new-cos 1

```

**AF2
remarking**

```

! AF21
mls qos map policed-dscp 18 bandwidth-class green to new-dscp 18
  new-bandwidth-class green new-queue 2 new-cos 2
mls qos map policed-dscp 18 bandwidth-class yellow to new-dscp 20
  new-bandwidth-class yellow new-queue 2 new-cos 2
mls qos map policed-dscp 18 bandwidth-class red to new-dscp 22
  new-bandwidth-class red new-queue 2 new-cos 2

! AF22
mls qos map policed-dscp 20 bandwidth-class green to new-dscp 20
  new-bandwidth-class yellow new-queue 2 new-cos 2
mls qos map policed-dscp 20 bandwidth-class yellow to new-dscp 20
  new-bandwidth-class yellow new-queue 2 new-cos 2
mls qos map policed-dscp 20 bandwidth-class red to new-dscp 22
  new-bandwidth-class red new-queue 2 new-cos 2

! AF23
! A previous switch saw this frame as red, so we cannot mark it as conformant.
mls qos map policed-dscp 22 bandwidth-class green to new-dscp 22
  new-bandwidth-class red new-queue 2 new-cos 2
mls qos map policed-dscp 22 bandwidth-class yellow to new-dscp 22
  new-bandwidth-class red new-queue 2 new-cos 2
mls qos map policed-dscp 22 bandwidth-class red to new-dscp 22
  new-bandwidth-class red new-queue 2 new-cos 2

```

**AF3
remarking**

```

! AF31
mls qos map policed-dscp 26 bandwidth-class green to new-dscp 26
  new-bandwidth-class green new-queue 3 new-cos 3
mls qos map policed-dscp 26 bandwidth-class yellow to new-dscp 28
  new-bandwidth-class yellow new-queue 3 new-cos 3
mls qos map policed-dscp 26 bandwidth-class red to new-dscp 30
  new-bandwidth-class red new-queue 3 new-cos 3

! AF32
mls qos map policed-dscp 28 bandwidth-class green to new-dscp 28
  new-bandwidth-class yellow new-queue 3 new-cos 3
mls qos map policed-dscp 28 bandwidth-class yellow to new-dscp 28
  new-bandwidth-class yellow new-queue 3 new-cos 3
mls qos map policed-dscp 28 bandwidth-class red to new-dscp 30
  new-bandwidth-class yellow new-queue 3 new-cos 3

```

```

! AF33
! A previous switch saw this frame as red, so we cannot mark it as conformant.
mls qos map policed-dscp 30 bandwidth-class green to new-dscp 30
  new-bandwidth-class red new-queue 3 new-cos 3
mls qos map policed-dscp 30 bandwidth-class yellow to new-dscp 30
  new-bandwidth-class red new-queue 3 new-cos 3
mls qos map policed-dscp 30 bandwidth-class red to new-dscp 30
  new-bandwidth-class red new-queue 3 new-cos 3

```

AF4 remarking

```

! AF41
mls qos map policed-dscp 34 bandwidth-class green to new-dscp 34
  new-bandwidth-class green new-queue 4 new-cos 4
mls qos map policed-dscp 34 bandwidth-class yellow to new-dscp 36
  new-bandwidth-class yellow new-queue 4 new-cos 4
mls qos map policed-dscp 34 bandwidth-class red to new-dscp 38
  new-bandwidth-class red new-queue 4 new-cos 4

```

```

! AF42
mls qos map policed-dscp 36 bandwidth-class green to new-dscp 36
  new-bandwidth-class yellow new-queue 4 new-cos 4
mls qos map policed-dscp 36 bandwidth-class yellow to new-dscp 36
  new-bandwidth-class yellow new-queue 4 new-cos 4
mls qos map policed-dscp 36 bandwidth-class red to new-dscp 38
  new-bandwidth-class red new-queue 4 new-cos 4

```

```

! AF43
! A previous switch saw this frame as red, so we cannot mark it as conformant.
mls qos map policed-dscp 38 bandwidth-class green to new-dscp 38
  new-bandwidth-class red new-queue 4 new-cos 4
mls qos map policed-dscp 38 bandwidth-class yellow to new-dscp 38
  new-bandwidth-class red new-queue 4 new-cos 4
mls qos map policed-dscp 38 bandwidth-class red to new-dscp 38
  new-bandwidth-class red new-queue 4 new-cos 4

```

EF remarking

```

! EF
mls qos map policed-dscp 46 bandwidth-class green to new-dscp 46
  new-bandwidth-class green new-queue 5 new-cos 5
mls qos map policed-dscp 46 bandwidth-class yellow to new-dscp 46
  new-bandwidth-class yellow new-queue 5 new-cos 5
mls qos map policed-dscp 46 bandwidth-class red to new-dscp 46
  new-bandwidth-class red new-queue 5 new-cos 5

```

Egress queue scheduling

Egress queue scheduling is the stage that most determines the quality of service each traffic type receives.

This example uses:

- **Strict Priority** scheduling for the high priority queues 5-7. EF traffic is in queue 5 and management traffic is in queue 7.
- **Weighted Round Robin (WRR)** scheduling for queues 0-4, with weights of 5, 15, 20, 25, and 35 respectively. Default traffic is in queue 0 and AF traffic is in queues 1-4.

This queue scheduling means that:

- management traffic has top priority (in queue 7)
- voice traffic has next priority (in queue 5)
- interactive session traffic, voice signalling, database traffic and default traffic share the remaining bandwidth in a weighted round robin fashion (in queues 0-4)

If the standard PHB scheme does not completely suit your network, you can tweak the queue scheduling. For example, the queue scheduling in this section places traffic for the Default PHB in the WRR group instead of leaving it strict priority. This avoids starving the default class. Even when the other low-priority classes are busy, the default class gets to forward some traffic.

For another example, if you had streaming video traffic as well as voice traffic, you could place the video stream into queue 4 and make it a strict priority queue. This would make queue 4 act like it was EF instead of AF.

Scheduling uses interface commands so go into interface mode.

```
interface port1.0.1-1.0.24
```

Configure strict priority queuing for queues 5-7:

```
priority-queue 5 6 7
```

Configure the default queue, for unmarked frames:

```
mls qos queue 0
```

Configure Weighted Round Robin scheduling on lower-priority queues:

```
wrr-queue group 1 weight 6 queues 0
wrr-queue group 1 weight 15 queues 1
wrr-queue group 1 weight 20 queues 2
wrr-queue group 1 weight 25 queues 3
wrr-queue group 1 weight 35 queues 4
```

In the next section, we configure the RED queue set 3, and this queue set will also be applied to this interface.

Because we have high priority traffic on queues 5, 6 and 7, we have chosen a strict priority scheduling configuration above. We will configure RED in such a way that it will not do any early discards on these queues.

RED curves

We define RED curves so we can provide a customised "random detect" traffic shaping. We need to create a RED curve set for the different queues and bandwidth classes (green, yellow, red). Of course, we give the red bandwidth class 3 a higher probability of being dropped.

A RED queue-set gets applied to the whole interface, but in this case we really only want to see this shaping occurring on the low priority queues 0-4 used for the default and AF traffic. (These queues also use WRR egress scheduling as described in the previous section). For the other queues 5, 6 and 7, we will configure the queue set with values that stop RED from providing any early discards of packets.

Queue sets are configured in global configuration mode.

Queues 0-4 Configure queue set 3 for queues 0-4.

For the thresholds, values are in bytes and the **syntax** is:

```
mls qos queue-set <1-4> queues {0|1|2|3|4|5|6|7} threshold <min-green
bytes> <max-green bytes> <min-yellow bytes> <max-yellow bytes> <min-red
bytes> <max-red bytes>
```

Here is the CLI command to assign the thresholds to the different bandwidth classes. Give green traffic the highest threshold:

```
mls qos queue-set 3 queues 0 1 2 3 4 threshold 30000 40000 20000 30000
10000 20000
```

For the drop probability, values are integers and the **syntax** is:

```
mls qos queue-set <1-4> queues [0][1][2][3][4][5][6][7] drop-probability
<green> <yellow> <red>
```

Here is the CLI command to assign the thresholds to the different bandwidth classes. Give green traffic a value of 3 = 12.5% drop probability, yellow traffic 2 = 25%, and red traffic 1 = 50% drop probability:

```
mls qos queue-set 3 queues 0 1 2 3 4 drop-probability 3 2 1
```

Queues 5-7 Now we need to configure queues 5, 6 and 7 so that the RED shaper will not do any early discard on these queues. We need to do this because queues 5, 6 and 7 carry the highest priority traffic. To avoid any early discards on these high priority queues, we raise the threshold to a value higher than the queue is ever likely to fill to, and reduce drop probability to 0%.

We give all traffic high thresholds and a drop probability value of 15 0%.

```
mls qos queue-set 3 queues 5 6 7 threshold 100000 200000 100000 200000
100000 200000
mls qos queue-set 3 queues 5 6 7 drop-probability 15 15 15
```

Finally, we apply the queue set to the ports:

```
interface port1.0.1-1.0.24
mls qos queue-set 3 random-detect
```

Configuration of access switches

As well as the "[Configuration common to all the switches](#)" on page 15, add this section's configuration to each access switch.

The access switches use class maps matching on various DSCP values or traffic types to assign traffic to different policy maps. Some class maps need to use an access list to match the required traffic. All packets are policed, and may be remarked and placed in appropriate priority egress queues. Forwarding or dropping decisions will be based on the behaviour of the egress queues, and the level of congestion at that moment.

Start by naming the switch:

```
hostname access-switch
```

Configure class maps for VoIP traffic to and from the SIP server

We assume that all the VoIP equipment (handsets and SIP server) has been configured to mark SIP signaling packets with DSCP=26, 28, or 30 and VoIP RTP packets with DSCP=46. Therefore, the class maps in this section match on VoIP traffic in both directions.

Match on DSCP 46 to capture VoIP RTP traffic:

```
class-map cmap11
match ip-dscp 46
```

Match on DSCP 26, 28, 30 - which correlates to AF Assured Forwarding codes 31, 32, and 33 to capture SIP signalling traffic from all 3 bandwidth classes:

```
class-map cmap12
match ip-dscp 26
class-map cmap13
match ip-dscp 28
class-map cmap14
match ip-dscp 30
```

Configure class maps for other traffic to the server farm

The next set of commands match on other traffic **to** the server farm.

To match traffic to the **interactive** server, we use an IP address /TCP port combination, in an ACL. Then we use the ACL in a class map.

```
access-list 3021 permit tcp any <dest-server-ip>/<mask> eq <dest-port>
class-map cmap21
match access-group 3021
```

Similarly, to match traffic to the **database** server, use the commands:

```
access-list 3031 permit tcp any <dest-server-ip>/<mask> eq <dest-port>
class-map cmap31
match access-group 3031
```

Configure class maps for other traffic from the server farm

The next set of commands match on other traffic **from** the server farm.

We can use DSCP values in this example because the server farm switch will label traffic with relevant DSCP values before it forwards it to the access switches.

To match traffic from the **interactive** server, use the commands:

```
class-map cmap22
match ip-dscp 34
class-map cmap23
match ip-dscp 36
class-map cmap24
match ip-dscp 38
```

Similarly, to match traffic from the **database** server, use the commands:

```
class-map cmap32
match ip-dscp 18
class-map cmap33
match ip-dscp 20
class-map cmap34
match ip-dscp 22
```

Configure a policy map for traffic to the servers

The policy map collects together all traffic destined for the server farm. Policers are applied to class maps in the policy map, to police the traffic.

The policy map is named pmap11. It contains cmaps 11, 12, 13, 14, 21, and 31.

The policing depends on the traffic type:

- VoIP—twin-rate policing with minimum traffic rate 80 Mbps, maximum traffic rate 100 Mbps, minimum burst 800 Kbytes, maximum burst 1000 Kbytes, and red packets dropped.

Red packets are dropped because if too much traffic is put into the top-priority queue, then the packets no longer really get the advantage of being top priority; they end up sitting in the queue and suffering from delay and jitter. Also, if too many packets are being put into the top-priority queue, then the lower-priority queues become starved. For these reasons, typical networks will limit EF traffic to no more than 30%—and often much less—of the capacity of a link.

- interactive server—twin-rate policing with minimum traffic rate 8 Mbps, maximum traffic rate 10 Mbps, minimum burst 80 Kbytes, maximum burst 100 Kbytes, and traffic remarking.
- database server—twin-rate policing with minimum traffic rate 240 Mbps, maximum traffic rate 300 Mbps, minimum burst 2400 Kbytes, maximum burst 3000 Kbytes, and traffic remarking.

Note: The policing rates shown in this document always quote the same nominal figures on all the switches in this topology - for demonstrative purposes only. You will need to calculate policing rates that are appropriate for your needs. You might decide to set the policing rates

based on a calculated percentage of the available uplink bandwidth, or you might decide to set an absolute rate for a certain traffic type, beyond which traffic must be dropped.

Note: Please also note that the AlliedWare Plus OS enables you to police ports and different types of traffic separately (with "ordinary" policers) or in combination (with aggregate policers). Please refer to http://10.32.16.105/thefount/docs/how_to_note_alliedware_plus/overview_aw+_qos_c.pdf

In the example below, we illustrate two alternate ways to do the policing. Alternative 1 is: ordinary policers, Alternative 2 is a "combined traffic types on combined ports" policer that will achieve aggregate policing of all combined upstream VoIP traffic.

Alternative 1: In this alternative are "ordinary" policers that meter traffic types separately and on the ingress ports separately.

```
policy-map pmap11
! VoIP
class cmap11
police twin-rate 80000 100000 800000 1000000 action drop-red
class cmap12
police twin-rate 80000 100000 800000 1000000 action drop-red
class cmap13
police twin-rate 80000 100000 800000 1000000 action drop-red
class cmap14
police twin-rate 80000 100000 800000 1000000 action drop-red
! Interactive server
class cmap21
set dscp 34
police twin-rate 8000 10000 80000 100000 action policed-dscp-transmit
! Database server
class cmap31
set dscp 18
police twin-rate 240000 300000 2400000 3000000 action policed-dscp-transmit
```

Alternative 2: To do this we first must define the aggregate policer:

```
mls qos aggregate-police pol1 twin-rate 80000 100000 800000 1000000 action drop-red
```

The VoIP Class Maps were already defined above - cmap11, cmap12, cmap13, and cmap14.

Now we apply these class maps to the policy map to use the aggregate policer, as follows:

```
policy-map pmap11
! VoIP
class cmap11
police aggregate pol1
class cmap12
police aggregate pol1
class cmap13
police aggregate pol1
class cmap14
police aggregate pol1
```

Configure a policy map for traffic from the servers

The policy maps collect together all traffic that comes from the server farm. Policers are applied to class maps in the policy map, to police the traffic.

The policy map is named pmap13. It contains cmaps 11, 12, 13, 14, 22, 23, 24, 32, 33, and 34.

The same policing applies as for traffic to the server farm.

```
policy-map pmap13
! VoIP
class cmap11
police twin-rate 80000 100000 800000 1000000 action drop-red
class cmap12
police twin-rate 80000 100000 800000 1000000 action drop-red
class cmap13
police twin-rate 80000 100000 800000 1000000 action drop-red
class cmap14
police twin-rate 80000 100000 800000 1000000 action drop-red

! Interactive server
class cmap22
police twin-rate 8000 10000 80000 100000 action policed-dscp-transmit
class cmap23
police twin-rate 8000 10000 80000 100000 action policed-dscp-transmit
class cmap24
police twin-rate 8000 10000 80000 100000 action policed-dscp-transmit

! Database server
class cmap32
police twin-rate 240000 300000 2400000 3000000 action
  policed-dscp-transmit
class cmap33
police twin-rate 240000 300000 2400000 3000000 action
  policed-dscp-transmit
class cmap34
police twin-rate 240000 300000 2400000 3000000 action
  policed-dscp-transmit
```

Apply the policy maps to the ports

Apply pmap11 to the access ports.

```
interface port1.0.1-port1.0.22
switchport mode access
switchport access vlan1
service-policy input pmap11
```

Apply pmap13 to the trunk ports.

```
interface port1.0.23-port1.0.24
switchport mode trunk
switchport trunk allowed vlan all
service-policy input pmap13
```

Configuration of server farm switch

As well as the "Configuration common to all the switches" on page 15, add this section's configuration to the server farm switch.

Start by naming the switch:

```
hostname server-switch
```

Configure class maps for VoIP traffic to and from the SIP server

We assume that all the VoIP equipment (handsets and SIP server) has been configured to mark SIP signaling packets with DSCP=26, 28, or 30. Therefore, the class maps in this section match on VoIP traffic in both directions.

Note: In this case, the server farm switch is not expecting VoIP RTP traffic to connect to the SIP server, so a classmap for DSCP 46 is not defined.

Match on DSCP 26, 28, 30 - which correlates to AF Assured Forwarding codes 31, 32 and 33 to capture SIP signalling traffic from all 3 bandwidth classes:

```
class-map cmap11
match ip-dscp 26
class-map cmap12
match ip-dscp 28
class-map cmap13
match ip-dscp 30
```

Configure class maps for other traffic to the server farm

The next set of commands match on other traffic to the server farm.

We can use DSCP values in this example because the access switch will label traffic with relevant DSCP values before it forwards it to the servers.

To match traffic **to the interactive** server, use the commands:

```
class-map cmap21
match ip-dscp 34
class-map cmap22
match ip-dscp 36
class-map cmap23
match ip-dscp 38
```

Similarly, to match traffic **to the database** server, use the commands:

```
class-map cmap31
match ip-dscp 18
class-map cmap32
match ip-dscp 20
class-map cmap33
match ip-dscp 22
```

Configure class maps for other traffic from the server farm

To match traffic **from the interactive** server, we match on the server's source IP address / TCP port combination, in an ACL. Then we use the ACL in a class map. Use the commands:

```
access-list 3024 permit tcp <server-source-ip>/<mask> eq <source-port> any
class-map cmap24
match access-group 3024
```

Similarly, to match traffic **from the database** server, use the commands:

```
access-list 3034 permit tcp <server-source-ip>/<mask> eq <source-port> any
class-map cmap34
match access-group 3034
```

Configure a policy map for traffic to the servers

The policy map collects together all traffic destined for the server farm. Policers are applied to class maps in the policy map, to police the traffic.

The policy map is named pmap11. It contains cmaps 11, 12, 13, 21, 22, 23, 31, 32, and 33.

The policing depends on the traffic type:

- VoIP—twin-rate policing with minimum traffic rate 80 Mbps, maximum traffic rate 100 Mbps, minimum burst 800 Kbytes, maximum burst 1000 Kbytes, and red packets dropped.

Red packets are dropped because if too much traffic is put into the top-priority queue, then the packets no longer really get the advantage of being top priority; they end up sitting in the queue and suffering from delay and jitter. Also, if too many packets are being put into the top-priority queue, then the lower-priority queues become starved. For these reasons, typical networks will limit EF traffic to no more than 30%—and often much less—of the capacity of a link.

- interactive server—twin-rate policing with minimum traffic rate 8 Mbps, maximum traffic rate 10 Mbps, minimum burst 80 Kbytes, maximum burst 100 Kbytes, and traffic remarking.
- database server—twin-rate policing with minimum traffic rate 240 Mbps, maximum traffic rate 300 Mbps, minimum burst 2400 Kbytes, maximum burst 3000 Kbytes, and traffic remarking.

Note: The policing rates shown in this document always quote the same nominal figures on all the switches in this topology - for demonstrative purposes only. You will need to calculate policing rates that are appropriate for your needs. You might decide to set the policing rates based on a calculated percentage of the available uplink bandwidth, or you might decide to set an absolute rate for a certain traffic type, beyond which traffic must be dropped.

Note: Please also note that the AlliedWare Plus OS enables you to police ports and different types of traffic separately (with "ordinary" policers) or in combination (with aggregate policers). Please refer to http://10.32.16.105/thefount/docs/how_to_note_alliedware_plus/overview_aw+_qos_c.pdf. An example of an aggregate policer has been shown in the Access Switch section for the upstream VoIP traffic.

```

policy-map pmap11
! SIP server
class cmap11
police twin-rate 80000 100000 800000 1000000 action drop-red
class cmap12
police twin-rate 80000 100000 800000 1000000 action drop-red
class cmap13
police twin-rate 80000 100000 800000 1000000 action drop-red

! Interactive server
class cmap21
police twin-rate 8000 10000 80000 100000 action policed-dscp-transmit
class cmap22
police twin-rate 8000 10000 80000 100000 action policed-dscp-transmit
class cmap23
police twin-rate 8000 10000 80000 100000 action policed-dscp-transmit

! Database server
class cmap31
police twin-rate 240000 300000 2400000 3000000 action
  policed-dscp-transmit
class cmap32
police twin-rate 240000 300000 2400000 3000000 action
  policed-dscp-transmit
class cmap33
police twin-rate 240000 300000 2400000 3000000 action
  policed-dscp-transmit

```

Configure a policy map for traffic from the servers

The policy map collects together all traffic that comes from the server farm. Policers are applied to class maps in the policy map, to police the traffic.

The policy map is named pmap13. It contains cmaps 11, 12, 13, 24, and 34.

The same policing applies as for traffic to the server farm.

Note: The policing rates shown in this document always quote the same nominal figures on all the switches in this topology - for demonstrative purposes only. You will need to calculate policing rates that are appropriate for your needs. You might decide to set the policing rates based on a calculated percentage of the available uplink bandwidth, or you might decide to set an absolute rate for a certain traffic type, beyond which traffic must be dropped.

Note: Please also note that the AlliedWare Plus OS enables you to police ports and different types of traffic separately (with "ordinary" policers) or in combination (with aggregate policers). Please refer to http://10.32.16.105/thefount/docs/how_to_note_alliedware_plus/overview_aw+_qos_c.pdf. An example of an aggregate policer has been shown in the Access Switch section for the upstream VoIP traffic.

```
policy-map pmap13
! SIP server
class cmap11
police twin-rate 80000 100000 800000 1000000 action drop
class cmap12
police twin-rate 80000 100000 800000 1000000 action drop
class cmap13
police twin-rate 80000 100000 800000 1000000 action drop
! Interactive server
class cmap24
set dscp 34
police twin-rate 8000 10000 80000 100000 action policed-dscp-transmit
! Database server
class cmap34
set dscp 18
police twin-rate 240000 300000 2400000 3000000 action
  policed-dscp-transmit
```

Apply the policy maps to the ports

Apply pmap11 to the ports on the core-facing side of the switch. These ports are tagged trunk ports.

```
interface port1.0.23-port1.0.24
switchport mode trunk
switchport trunk allowed vlan all
service-policy input pmap11
```

Apply pmap13 to the server-facing ports. These ports are standard access ports.

```
interface port1.0.1-port1.0.22
switchport mode access
switchport access vlan1
service-policy input pmap13
```

Configuration of core switch

As well as the "[Configuration common to all the switches](#)" on page 15, add this section's configuration to the core switch.

This switch's QoS handling is bi-directional—it treats traffic in the same way whether it comes from the server farm or access side of the network.

When packets arrive at the core switch, they will already have DSCP values because the server farm or access switches will have marked them with DSCPs.

Start by naming the switch:

```
hostname core-switch
```

Configure class maps

We assume that all the VoIP equipment (handsets and SIP server) has been configured to mark SIP signaling packets with DSCP=26, 28, or 30.

Match on DSCP 46 to capture RTP voice traffic. Also match on DSCP 26, 28, and 30 which correlates to AF Assured Forwarding codes 31, 32 and 33 - to capture SIP signalling traffic from all 3 bandwidth classes:

```
class-map cmap11
match ip-dscp 46
class-map cmap12
match ip-dscp 26
class-map cmap13
match ip-dscp 28
class-map cmap14
match ip-dscp 30
```

Match on DSCP 34, 36, and 38 for interactive server traffic:

```
class-map cmap21
match ip-dscp 34
class-map cmap22
match ip-dscp 36
class-map cmap23
match ip-dscp 38
```

Match on DSCP 18, 20, and 22 for database server traffic:

```
class-map cmap31
match ip-dscp 18
class-map cmap32
match ip-dscp 20
class-map cmap33
match ip-dscp 22
```

Configure a policy map

The policy map collects all traffic that comes to the core switch from either direction. Policers are applied to class maps in the policy map, to police the traffic.

The policy map is named pmap11. It contains all the class maps.

Note: The policing rates shown in this document always quote the same nominal figures on all the switches in this topology - for demonstrative purposes only. You will need to calculate policing rates that are appropriate for your needs. You might decide to set the policing rates based on a calculated percentage of the available uplink bandwidth, or you might decide to set an absolute rate for a certain traffic type, beyond which traffic must be dropped.

Note: Please also note that the AlliedWare Plus OS enables you to police ports and different types of traffic separately (with "ordinary" policers) or in combination (with aggregate policers). Please refer to http://10.32.16.105/thefount/docs/how_to_note__alliedware_plus/overview_aw+_qos_c.pdf. An example of an aggregate policer has been shown in the Access Switch section for the upstream VoIP traffic.

```

policy-map pmap11
! VoIP
class cmap11
police twin-rate 80000 100000 800000 1000000 action drop
class cmap12
police twin-rate 80000 100000 800000 1000000 action drop
class cmap13
police twin-rate 80000 100000 800000 1000000 action drop
class cmap14
police twin-rate 80000 100000 800000 1000000 action drop

! Interactive server
class cmap21
police twin-rate 8000 10000 80000 100000 action policed-dscp-transmit
class cmap22
police twin-rate 8000 10000 80000 100000 action policed-dscp-transmit
class cmap23
police twin-rate 8000 10000 80000 100000 action policed-dscp-transmi

! Database server
class cmap31
police twin-rate 240000 300000 2400000 3000000 action policed-dscp-transmit
class cmap32
police twin-rate 240000 300000 2400000 3000000 action policed-dscp-transmit
class cmap33
police twin-rate 240000 300000 2400000 3000000 action policed-dscp-transmit

```

Apply the policy maps to the ports

Apply pmap11 to all ports on the switch.

```

interface port1.0.1-port1.0.24
switchport mode trunk
switchport trunk allowed vlan all
service-policy input pmap11

```