How To | How to configure software QoS for some specific customer scenarios

Introduction

The purpose of this document is to give some real-world configuration examples for software QoS (SQoS). In each case, the customer requirements are described, and then the configuration is given which satisfies those requirements. In some cases, there is a discussion of the reasoning behind the configuration choices.

What information will you find in this document?

This document provides information on:

- Policy routing and VoIP prioritisation
- Voice over IP (VoIP) over a Frame Relay connection
- Offering different bandwidth to different users
- Providing different levels of service to different types of traffic
- Marking packets
- Bandwidth limiting and VoIP prioritisation

Which product and software version does this information apply to?

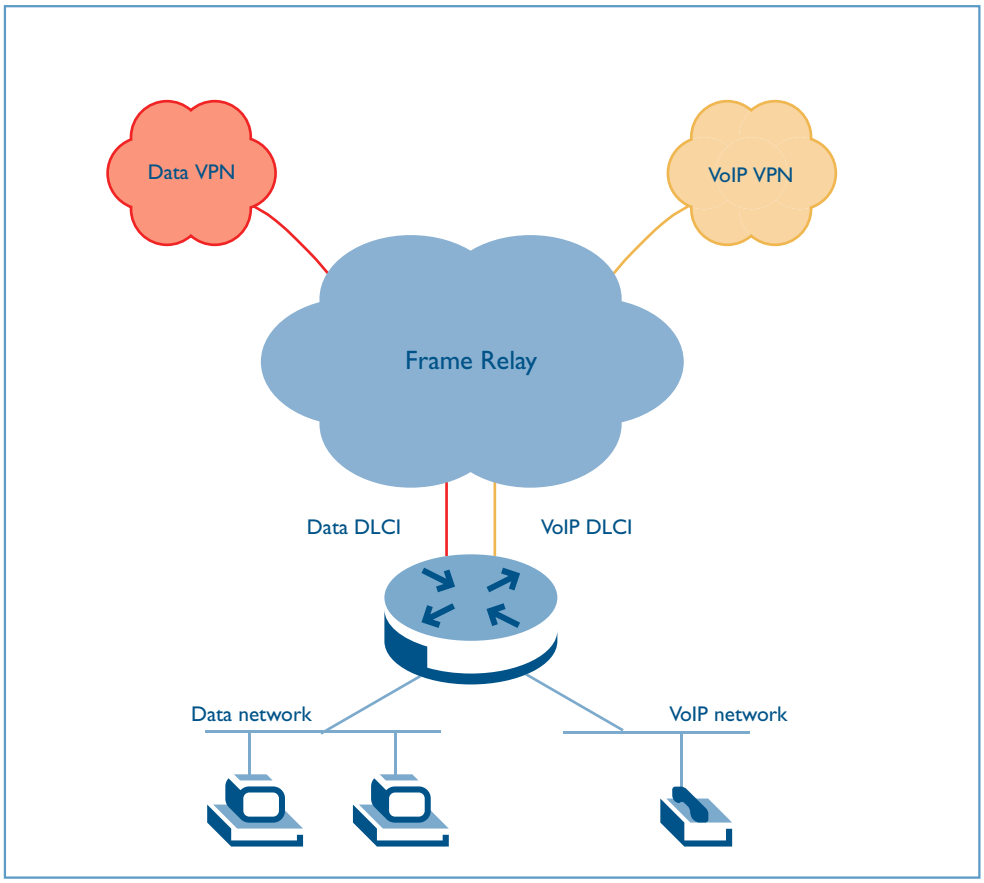
The information provided in this document applies to the following:

Products: AR400 series, AR700 series, Rapier series, AT8800

Software versions: 2.7.1 and above

Example1: Policy routing and VoIP traffic prioritisation

Scenario



A number of schools are attached to a Frame Relay network. Each school has a VoIP network and a data network.

Requirements

The customer requirements are as follows:

- The VoIP traffic is to be 802.1p marked in the LAN before entering the router, while the data traffic is to remain unmarked.
- The traffic is to enter the WAN using 2 Frame Relay Data Link Control Interfaces (DLCIs) in the same Frame Relay interface. One Data Link Control (DLC) is dedicated to VoIP traffic, while the other is for data. The traffic will be policy routed to the correct DLC.
- VoIP traffic has to be prioritised over data, at least for the guaranteed VoIP BW. This bandwidth is equal to "One call BW" multiplied by "number of telephone lines the school currently has". For example, if you consider a school with 10 lines and using a Codec with 30 Kbps per call, the guaranteed bandwidth for VoIP should be 300 Kbps.
- A minimum bandwidth for data has to be guaranteed as well (say 400 Kbps).

Solution

A solution for the customer requirements is outlined below:

▶ Create the Frame Relay interface with two FRLIs:

```
create fr=0 over=bay0.syn0 lmischeme=annexd
add fr=0 li=101 type=ptp
add fr=0 li=201 type=ptp
set fr=0 dlc=101 li=101
set fr=0 dlc=201 li=201
```

▶ Create a classifier to match RTP data:

```
create classifier=1 ipprotocol=udp udpd=13800-13880
```

▶ Create a classifier to match SIP signalling traffic:

```
create classifier=2 ipprotocol=udp udpd=5060
```

▶ Create two IP policy filters:

- IP filter 101 routes all the packets received from eth0 (tagged VoIP port) to DLCI 101.
- IP filter 102 routes all the packets received from eth1 (untagged data port) to DLCI 201.

```
enable ip
add ip filter=101 so=0.0.0.0 policy=1
add ip filter=102 so=0.0.0.0 policy=2
add ip int=eth0 ip=1.1.1.1 mask=255.255.255.0 policy=101 vlan=2
add ip int=eth1 ip=2.2.2.2 mask=255.255.255.0 policy=102
add ip int=fr0.101 ip=101.1.1.1 mask=255.255.255.0
add ip int=fr0.201 ip=201.1.1.1 mask=255.255.255.0
add ip rou=0.0.0.0 mask=0.0.0.0 int=fr0.101 next=101.1.1.2 policy=1
  dlc=101
add ip rou=0.0.0.0 mask=0.0.0.0 int=fr0.201 next=201.1.1.2 policy=2
  dlc=201
```

▶ Set the limit for VoIP traffic to 300kbps (note that the default min and max burst values are 10kbytes):

```
create sqos met=1 max=300kbps
```

▶ Set the limit for SIP traffic to 64Kbps:

In general, network signalling should never reach to 64Kbps.

```
create sqos met=2 max=64kbps
```

▶ Set the guaranteed bandwidth for the data traffic to 400Kbps:

```
create sqos met=3 max=400kbps
```

▶ Apply meter 1 to the VoIP traffic and give it a priority of 10:

```
create sqos trafficclass=1 priority=10 meter=1 bwclass3action=DROP
```

▶ Apply meter 2 to the SIP traffic and give a priority of 15 (which is the highest):

```
create sqos trafficclass=2 priority=15 meter=2 bwclass3action=DROP
```

▶ Apply meter 3 to the data traffic:

Do not configure this traffic class to drop non-conformant traffic.

```
create sqos trafficclass=3 meter=3
```

▶ Create a policy to attach these traffic classes to, and then attach the classifiers to the traffic classes:

```
create sqos policy=1
add sqos policy=1 trafficclass=1,2
add sqos tr=1 class=1
add sqos tr=2 class=2
```

▶ Set traffic class 3 to be the default traffic class on the policy, so it will match all the data that does not match traffic classes 1 or 2:

```
Set sqos policy=1 defaulttrafficclass=3
```

▶ Apply SQoS to the Frame Relay interface (not to the sub-interfaces of Frame Relay):

```
set sqos int=fr0 outpolicy=1
enable sqos
```

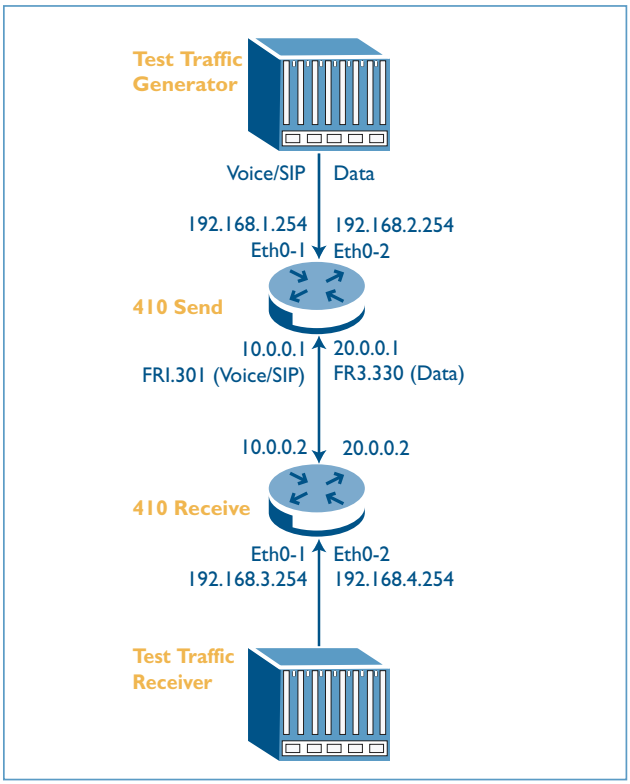
Example 2: VoIP over a Frame Relay connection

Voice over an IP network (VoIP) is becoming increasingly popular. As Service providers look to meet customers' complete communications needs, we see them offering packages that will support VoIP and data traffic together on a single connection.There are a few issues that arise when having multiple types of data on a single link that need to be addressed. Delay-sensitive traffic, like VoIP and streaming video needs to be delivered in a timely fashion, data traffic may not experience a problem if delayed, and can even be quite tolerant of packet loss.

Traffic management

A methodology needs to be employed to prioritise certain traffic types to ensure timely delivery and that adequate bandwidth is available. With a traditional Frame Relay link, one method for doing this is to allow long data frames to be fragmented into smaller pieces and interleaved with real-time frames. In this way, real-time VoIP and non real-time data frames can be carried together on lower speed links without causing excessive delay to the real-time traffic. This idea is detailed in the FRF.12 Implementation Agreement (also known as FRF.11 Annex C). While this will help in the timely delivery of the VoIP traffic, there are other ways to minimise VoIP delay, which also provide further options for traffic shaping and bandwidth management. Here we give an example of how to use policy-based routing and software quality of service (SQoS) to manage the bandwidth available in a multi-use customer connection to a service provider.

Scenario



The customer has a Frame Relay over PRI 2Mbps connection to their service provider. This connection is required to provide access for VoIP, SIP/management traffic and data traffic. The data traffic will use one portion of the available 2Mbps bandwidth and the VoIP/ SIP traffic will use the

other. We will show how this scenario was tested with 2 routers back to back in a lab environment to test this functionality.

The equipment used for this testing was two AR410 routers running 52-270 release (an early adopter release of SW QoS).

Traffic

Traffic used in testing this set up was as follows:

- 3 VoIP streams of 200 byte packets at 50 packets / sec.
- 1 SIP stream of 1500 byte packets at 32Kbps.
- 1 data stream of 1500 byte packets at 1.5Mbps.

Traffic streams were sent to different destination UDP ports to allow differentiation by the router. This could also have been done by some other mechanism like DSCP.

Solution

The configurations on the routers were mirrors of each other, as traffic was sent in both directions. The configuration for the **410 Send** router is shown below with details following:

▶ Set system name and split interface into 2 parts:

SYSTEM configuration

```
set system name="410_Send"

set pri=bay0.pri0 mode=tdm clock=internal
create tdm group=data3 interface=bay0.pri0 slots=7-31
create tdm group=voip1 interface=bay0.pri0 slots=1-6
```

▶ Create Frame Relay interfaces:

```
create fr=1 over=tdm-voip1 lmischeme=none
add fr=1 li=301 type=ptp
add fr=1 dlc=301
set fr=1 dlc=301 li=301

create fr=3 over=tdm-data3 lmischeme=none
add fr=3 li=330 type=ptp
add fr=3 dlc=330
set fr=3 dlc=330 li=330
```

▶ Set maximum packet size:

```
set int=fr1 mtu=256
```

▶ Configure policy routing:

```
enable ip
add ip fil=101 so=0.0.0.0 ent=1 poli=1 prot=udp dp=16384:16386
add ip fil=101 so=0.0.0.0 ent=2 poli=1 prot=udp dp=5060
add ip fil=101 so=0.0.0.0 ent=3 poli=2 prot=udp dp=5001
add ip int=fr1.301 ip=10.0.0.1 mask=255.255.255.0
add ip int=fr3.330 ip=20.0.0.1 mask=255.255.255.0
add ip int=eth0-1 ip=192.168.1.254 pol=101
add ip int=eth0-2 ip=192.168.2.254 pol=101
add ip rou=192.168.3.0 mask=255.255.255.0 int=fr1.301 next=10.0.0.2
  poli=1 dlc=301
add ip rou=192.168.4.0 mask=255.255.255.0 int=fr3.330 next=20.0.0.2
  poli=2 dlc=330
```

▶ Configure software QoS:

```
ena sqos
create class=1 udpd=16384-16386
create sqos meter=1 type=trtcm minband=280K maxband=280K minburst=10K
  maxburst=10K
create sqos trafficclass=1 priority=15 meter=1 bwclass3action=drop
create sqos trafficclass=2 weight=0
create sqos policy=1 default=2
add sqos policy=1 trafficclass=1
add sqos trafficclass=1 classifier=1
set sqos int=fr1 outpolicy=1
```

Configuration description

Let's look at the configuration used piece by piece:

The PRI 2Mbps interface is split into two parts. Each slot on a channelised connection is 64Kbps.

- TDM group 'voip1' for the VoIP and SIP traffic receives slots 1-6, or 384Kbps.
- TDM group 'data3' receives slots 7-31, or 1.5Mbps

```
set pri=bay0.pri0 mode=tdm
set pri=bay0.pri0 cl=int

create tdm group=data3 interface=bay0.pri0 slots=7-31
create tdm group=voip1 interface=bay0.pri0 slots=1-6
```

Frame Relay interfaces

The Frame Relay interfaces are created over the relevant TDM groups with Frame Relay logical interface (FRLI) 1.301 created for VoIP and SIP/management traffic and FRLI 3.330 created for data traffic.

```
create fr=1 over=tdm-voip1 lmischeme=none
add fr=1 li=301 type=ptp
add fr=1 dlc=301
set fr=1 dlc=301 li=301

create fr=3 over=tdm-data3 lmischeme=none
add fr=3 li=330 type=ptp
add fr=3 dlc=330
set fr=3 dlc=330 li=330
```

Packet size

The maximum packet size is set to 256 for the VoIP Frame Relay interface to maximise performance.

```
set int=fr1 mtu=256
```

Policy routing

Policy routing is used to separate the incoming traffic and send data and VoIP over the correct Frame Relay 'WAN' interfaces. As mentioned earlier, UDP destination ports were used as the method to differentiate the traffic types. The VoIP calls used ports 16384-16386. SIP traffic used port 5060 and Data traffic used port 5001.

Received traffic

The traffic is received on the multi-homed Ethernet interface of the router. This could well be tagged VLAN traffic in a customer network. The data traffic arrives from the 192.168.2.0 network and is routed across the FR3.330 interface (20.0.0.0). The VoIP and SIP traffic is received from the 192.168.1.0 network and routed across the FR1.301 interface (10.0.0.0).

```
enable ip

add ip fil=101 so=0.0.0.0 ent=1 poli=1 prot=udp dp=16384:16386

add ip fil=101 so=0.0.0.0 ent=2 poli=1 prot=udp dp=5060

add ip fil=101 so=0.0.0.0 ent=3 poli=2 prot=udp dp=5001

add ip int=fr1.301 ip=10.0.0.1 mask=255.255.255.0

add ip int=fr3.330 ip=20.0.0.1 mask=255.255.255.0

add ip int=eth0-1 ip=192.168.1.254 pol=101

add ip int=eth0-2 ip=192.168.2.254 pol=101

add ip rou=192.168.3.0 mask=255.255.255.0 int=fr1.301 next=10.0.0.2
  poli=1 dlc=301

add ip rou=192.168.4.0 mask=255.255.255.0 int=fr3.330 next=20.0.0.2
  poli=2 dlc=330

create class=1 udpd=16384-16386
```

Software QoS management

Software QoS is used to manage the VoIP and SIP traffic on the FR1 interface to ensure the timely delivery of the realtime VoIP traffic.

The VoIP traffic is classified using (classifier 1 above) and put into traffic class 1 of our SQoS policy. This policy also has a default traffic class (traffic class2) which the SIP/ management traffic will go into. The SQoS policy is applied to the FR1 egress interface.

The FR1 interface is 384K (6 slots of 64K each). Each of the 3 VoIP streams is 80K, so we set a guaranteed minimum bandwidth of 280K (3 * 80K + a bit) for the VoIP traffic. A maximum of 280K is also set, so that some SIP/ management traffic can always get through when the network is congested.

The VoIP traffic (which is classified and put into traffic class 1), is given the highest priority available of 15. The SIP management traffic will have the default priority and is also given a low weight of 0. This should ensure that any VoIP packet that arrives at the interface is always sent before a SIP packet, but SIP does have some bandwidth available for times of congestion.

```
ena sqos

create sqos meter=1 type=trtcm minband=280K maxband=280K minburst=10K
  maxburst=10K

cre sqos tr=1 prio=15 meter=1 bwclass3action=drop

cre sqos tr=2 weight=0
```

```
cre sqos poli=1 default=2
add sqos poli=1 tr=1
add sqos tr=1 class=1
set sqos int=fr1 ou=1
```

Results

The results of this network test were very good. All traffic was received across the network and the VoIP traffic always had a low latency and low jitter.

Jitter (variation in delay) is critical for realtime traffic like voice and video. If there is some latency but it is even, this is not too much of a problem. If all packets are delayed by a few milliseconds no problem is perceived. If there is variation in delay (jitter) then the video or voice may come through sounding broken.

Testing with higher values of SIP and data still showed good results, as the VoIP traffic has high priority, so will be sent immediately by the router.

Example 3: Offering different bandwidth to different users

Requirements

The customer is a service provider who wants to perform limiting/guaranteeing bandwidth for their users.

The customer requirements are as follows:

- To offer a variety of price programs - 64/128/256/512Kbps, with the same speeds guarantees for every user.
- The ability to cater for tens of users on each price program.
- To give each user a static IP address - i.e. every time any given user connects, they get the same IP address.
- The ability to limit the amount of data that the user can send to the Internet and the amount of data they receive from the Internet.

Solution

Our solution to the customer requirements is outlined below:

First we need to create classifiers to match the traffic of different users. We need to match both incoming and outgoing traffic, so we need classifiers matching the users' addresses as dest addresses, and classifiers matching the users' addresses as source addresses. So we would have classifiers like:

```
create class=101 prot="ip" ipda=10.1.0.1/32
create class=102 prot="ip" ipda=10.1.0.2/32
create class=103 prot="ip" ipda=10.1.0.3/32
create class=104 prot="ip" ipda=10.1.0.4/32
create class=105 prot="ip" ipda=10.1.0.5/32
...
...
...
...
create class=201 prot="ip" ipsa=10.1.0.1/32
create class=202 prot="ip" ipsa=10.1.0.2/32
create class=203 prot="ip" ipsa=10.1.0.3/32
create class=204 prot="ip" ipsa=10.1.0.4/32
create class=205 prot="ip" ipsa=10.1.0.5/32
...
...
```

Then, we would create a traffic class for each user. Lets just discuss, first, the design of these traffic classes:

The aim of the QoS policy will be to limit/guarantee bandwidth for different users.

We will use single-rate 3-colour metering to do this job. Let us just think to ourselves - what does a single-rate 3-colour meter do?

Well, a meter assigns bandwidth class values (colours) to packets. For data that is under the CIR bandwidth, the packets are marked green; for data that is above the CIR, but within the burst size, packets are marked yellow; for data above the burst size, packets are marked red. So, a meter colours packets, but it does nothing else.

So, a meter colours the packets, but will it do any bandwidth limiting or bandwidth guarantee?

To actually limit bandwidth, we need to configure **something** that looks at the colours of the packets, and makes forwarding decisions based on the colours.

There are two possible 'somethings' that we could configure:

1. We could configure a traffic class to drop all packets that have been marked red:

   ```
   set trafficclass=<name> BWClass3action=drop
   ```

   This is called **policing**, and it is used to keep a traffic class strictly below a particular bandwidth limit.

2. We could configure RED curves that drop packets when congestion starts to occur - they can be aggressive in dropping red packets, not so aggressive in dropping yellow packets, and lenient in dropping green packets. This is called **shaping** - it allows a traffic class to use all the available bandwidth when no-one else wants it, but makes the traffic class drop back to its allocated bandwidth when congestion starts to occur.

The customer wants to **police** traffic.

So, that even if only one user is connected, and that user has a 64Kbps contract, they will be limited always to just 64Kbps. So, we will use option 1 on our traffic classes.

▶ Create meters for each contract

First we create 9 meters, for the 9 types of contract:

```
create sqos met=1 max=32kbps
create sqos met=2 max=64kbps
create sqos met=3 max=128kbps
create sqos met=4 max=192kbps
create sqos met=5 max=256kbps
create sqos met=6 max=384kbps
create sqos met=7 max=512kbps
create sqos met=8 max=768kbps
create sqos met=9 max=1.024Mbps
```

▶ Create traffic classes for the meters

Then create traffic classes that use these meters. Configure these traffic classes to drop packets that are marked red (so that it is policing traffic).

```
##The user with IP address 10.1.0.1  has a 64k contract:
    cre sqos tr=101 desc="user1 down 64" met=2 bwc=DROP
    cre sqos tr=201 desc="user1 up 64" met=2 bwc=DROP
##The user with IP address 10.1.0.2  has a 128k contract:
    cre sqos tr=102 desc="user2 down 128" met=3 bwc=DROP
    cre sqos tr=202 desc="user2 up 128" met=3 bwc=DROP
##The user with IP address 10.1.0.3 has a 128k contract:
    cre sqos tr=103 desc="user3 down 128" met=3 bwc=DROP
    cre sqos tr=203 desc="user3 up 128" met=3 bwc=DROP

    ...
    ...
    ...
```

▶ Create policies for outgoing and incoming traffic

Now create two policies; one policy is for outgoing traffic, and one for incoming traffic. Add the traffic classes to the policy, and then add classifiers to the traffic classes.

```
    cre sqos poli=1
    add sqos poli=1 tr=101,102,103,.....

    cre sqos poli=2
    add sqos poli=2 tr=201,202,203,.....

    add sqos tr=101 class=101
    add sqos tr=201 class=201
    add sqos tr=102 class=102
    add sqos tr=202 class=202
    add sqos tr=103 class=103
    add sqos tr=203 class=203
    ...
    ...
    ...
```

▶ Apply SQoS and a policy to the Ethernet interface

The public interface of the router is eth0. Policy 2 is applied as the outpolicy on this interface (the traffic classes on the policy all use classifiers that match on the users' IP addresses as source addresses).

```
set sqos int=eth0 outpolicy=2
```

The private interface of the router is the switch interface. We apply policy 1 to this interface. Now, there is one important parameter that must be configured on this policy because of the fact that the router is also going to be performing NAT. When packets arrive at the public interface of the router, their destination IP addresses will be the public IP address of the router. After they have passed through the firewall, their destination IP addresses will have been converted to the users' private IP addresses. By default, the destination IP address that will be considered when these packets are passed to the classifiers will be the dest IP address they had BEFORE they went through the firewall. This will not be very useful, because ALL packets would have the public IP address as destination address. So, the dest IP address to be considered by the classifiers needs to be the dest IP address on the packets AFTER they passed through the firewall. To ensure that this post-firewall address is the one considered by the classifiers, you must set the policy to ignore pre-NAT information:

▶ Set policy to ignore pre-NAT information

```
set sqos policy=1 ignoreprenatinfo=yes
set sqos int=swi0 outpolicy=1
```

Full configuration of the router

```
#
# CLASSIFIER general configuration
#
    create class=101 prot="ip" ipda=10.1.0.1/32
    create class=102 prot="ip" ipda=10.1.0.2/32
    create class=103 prot="ip" ipda=10.1.0.3/32
    create class=104 prot="ip" ipda=10.1.0.4/32
    create class=105 prot="ip" ipda=10.1.0.5/32
    create class=106 prot="ip" ipda=10.1.0.6/32
    create class=107 prot="ip" ipda=10.1.0.7/32
    create class=201 prot="ip" ipsa=10.1.0.1/32
    create class=202 prot="ip" ipsa=10.1.0.2/32
    create class=203 prot="ip" ipsa=10.1.0.3/32
    create class=204 prot="ip" ipsa=10.1.0.4/32
    create class=205 prot="ip" ipsa=10.1.0.5/32
    create class=206 prot="ip" ipsa=10.1.0.6/32
    create class=207 prot="ip" ipsa=10.1.0.7/32
    create class=1008 prot="ip" ipda=10.1.1.0/24
    create class=1009 prot="ip" ipda=10.1.2.0/24
    create class=1010 prot="ip" ipda=10.1.3.0/24
    create class=1011 prot="ip" ipda=10.1.4.0/24
    create class=1012 prot="ip" ipda=10.1.5.0/24
    create class=1013 prot="ip" ipda=10.1.6.0/24
```

```
    create class=1014 prot="ip" ipda=10.1.7.0/24
    create class=1108 prot="ip" ipsa=10.1.1.0/24
    create class=1109 prot="ip" ipsa=10.1.2.0/24
    create class=1110 prot="ip" ipsa=10.1.3.0/24
    create class=1111 prot="ip" ipsa=10.1.4.0/24
    create class=1112 prot="ip" ipsa=10.1.5.0/24
    create class=1113 prot="ip" ipsa=10.1.6.0/24
    create class=1114 prot="ip" ipsa=10.1.7.0/24
    create class=2008 prot="ip" ipda=82.119.251.64/28
    create class=2009 prot="ip" ipda=82.119.251.80/28
    create class=2010 prot="ip" ipda=82.119.251.96/29
    create class=2011 prot="ip" ipda=82.119.251.104/29
    create class=2012 prot="ip" ipda=82.119.251.112/30
    create class=2013 prot="ip" ipda=82.119.251.116/30
    create class=2014 prot="ip" ipda=82.119.251.120/29
    create class=2108 prot="ip" ipsa=82.119.251.64/28
    create class=2109 prot="ip" ipsa=82.119.251.80/28
    create class=2110 prot="ip" ipsa=82.119.251.96/29
    create class=2111 prot="ip" ipsa=82.119.251.104/29
    create class=2112 prot="ip" ipsa=82.119.251.112/30
    create class=2113 prot="ip" ipsa=82.119.251.116/30
    create class=2114 prot="ip" ipsa=82.119.251.120/29
#
# IP configuration
#
    enable ip
    ena ip dnsrelay
    add ip fil=1 so=0.0.0.0 ent=1 des=10.255.1.0 dm=255.255.255.0
      ac=exclude
    add ip fil=1 so=0.0.0.0 ent=2 ac=include
    add ip int=eth0 ip=193.179.159.123 mask=255.255.255.128
    set ip int=eth0 pro=off
    add ip int=vlan1 ip=82.119.251.1 mask=255.255.255.224
    add ip int=vlan10 ip=82.119.251.65 mask=255.255.255.192 fil=1
    add ip int=vlan11 ip=10.1.1.1 mask=255.255.248.0 fil=1
    add ip int=vlan12 ip=10.255.1.1 mask=255.255.255.0
    add ip rou=0.0.0.0 mask=0.0.0.0 int=eth0 next=193.179.159.65
    add ip dns prim=193.179.159.65
#
# FIREWALL configuration
#
    enable firewall
    create firewall policy="1"
    enable firewall policy="1" icmp_f=all
    add firewall policy="1" int=vlan12 type=private
    add firewall policy="1" int=vlan11 type=private
    add firewall policy="1" int=vlan10 type=private
    add firewall policy="1" int=vlan1 type=public
    add firewall policy="1" int=eth0 type=public
```

```
    add firewall poli="1" nat=enhanced int=vlan11 gblin=eth0
      gblip=193.179.159.123
#
# SQOS general configuration
#
    ena sqos
    cre sqos met=1 max=32kbps
    cre sqos met=2 max=64kbps
    cre sqos met=3 max=128kbps
    cre sqos met=4 max=192kbps
    cre sqos met=5 max=256kbps
    cre sqos met=6 max=384kbps
    cre sqos met=7 max=512kbps
    cre sqos met=8 max=768kbps
    cre sqos met=9 max=1.024Mbps
    cre sqos tr=101 desc="user1 down 64" met=2 bwc=DROP
    cre sqos tr=102 desc="user2 down 128" met=3 bwc=DROP
    cre sqos tr=103 desc="user3 down 128" met=3 bwc=DROP
    cre sqos tr=104 desc="user4 down 256" met=5 bwc=DROP
    cre sqos tr=105 desc="user5 down 256" met=5 bwc=DROP
    cre sqos tr=106 desc="user6 down 384" met=6 bwc=DROP
    cre sqos tr=107 desc="user7 down 512" met=7 bwc=DROP
    cre sqos tr=201 desc="user1 up 64" met=2 bwc=DROP
    cre sqos tr=202 desc="user2 up 128" met=3 bwc=DROP
    cre sqos tr=203 desc="user3 up 128" met=3 bwc=DROP
    cre sqos tr=204 desc="user4 up 256" met=5 bwc=DROP
    cre sqos tr=205 desc="user5 up 256" met=5 bwc=DROP
    cre sqos tr=206 desc="user6 up 384" met=6 bwc=DROP
    cre sqos tr=207 desc="user7 up 512" met=7 bwc=DROP
    cre sqos tr=1008 desc="user8 down 384" met=6 bwc=DROP
    cre sqos tr=1009 desc="user9 down 512" met=7 bwc=DROP
    cre sqos tr=1010 desc="user10 down 768" met=8 bwc=DROP
    cre sqos tr=1011 desc="user11 down 1024" met=9 bwc=DROP
    cre sqos tr=1012 desc="user12 down 32" met=1 bwc=DROP
    cre sqos tr=1013 desc="user13 down 64" met=2 bwc=DROP
    cre sqos tr=1014 desc="user14 down 128" met=3 bwc=DROP
    cre sqos tr=1108 desc="user8 up 384" met=6 bwc=DROP
    cre sqos tr=1109 desc="user9 up 512" met=7 bwc=DROP
    cre sqos tr=1110 desc="user10 up 768" met=8 bwc=DROP
    cre sqos tr=1111 desc="user11 up 1024" met=9 bwc=DROP
    cre sqos tr=1112 desc="user12 up 32" met=1 bwc=DROP
    cre sqos tr=1113 desc="user13 up 64" met=2 bwc=DROP
    cre sqos tr=1114 desc="user14 up 128" met=3 bwc=DROP
    cre sqos poli=1 desc=download
    add sqos poli=1
      tr=101,102,103,104,105,106,107,1008,1009,1010,1011,1012,1013,1014
    add sqos tr=101 class=101
    add sqos tr=102 class=102
    add sqos tr=103 class=103
    add sqos tr=104 class=104
    add sqos tr=105 class=105
    add sqos tr=106 class=106
    add sqos tr=107 class=107
    add sqos tr=1008 class=1008
    add sqos tr=1009 class=1009
    add sqos tr=1010 class=1010
    add sqos tr=1011 class=1011
```

```
add sqos tr=1012 class=1012
add sqos tr=1013 class=1013
add sqos tr=1014 class=1014
cre sqos poli=2 desc=upload
add sqos poli=2
  tr=201,202,203,204,205,206,207,1108,1109,1110,1111,1112,1113,1114
add sqos tr=201 class=201
add sqos tr=202 class=202
add sqos tr=203 class=203
add sqos tr=204 class=204
add sqos tr=205 class=205
add sqos tr=206 class=206
add sqos tr=207 class=207
add sqos tr=1108 class=1108
add sqos tr=1109 class=1109
add sqos tr=1110 class=1110
add sqos tr=1111 class=1111
add sqos tr=1112 class=1112
add sqos tr=1113 class=1113
add sqos tr=1114 class=1114
set sqos policy=1 ignoreprenatinfo=yes
set sqos int=swi0 outpolicy=1
set sqos int=eth0 outpolicy=2
```

Example 4: Providing different levels of service to different types of traffic

Requirements

The customer wants to limit/guarantee a percentage of bandwidth to each of several different flows:

Priority	Name	%Bandwidth	Source IP	Destination IP	Port
3	Messagerie	25	any	128.1.0.231	
1	Tele maintenance	20	any	any	5631,5632,4899 tcp/udp
2	Intranet	15	any	128.1.0.75, 192.168.254.1	
4	Internet	10	any	128.1.0.18	
5	Replication sql	10	any	128.1.0.4	
6	Default traffic class	20			

The router is connected to the Ethernet port of a separate WAN-connected device. The WAN connection on that device is a 512k leased line.

Solution

The solution for the customer requirements is outlined below:

▶ Create classifiers to match traffic types:

Create classifiers to match the traffic types specified by the customer:

```
create class=10 ippr=tcp tcpd=5631-5632
create class=11 ippr=udp udpd=5631-5632
create class=12 tcpd=4899
create class=13 udpd=4899
create class=20 prot="ip" ipda=128.1.0.75/32
create class=21 prot="ip" ipda=192.168.254.1/32
create class=30 prot="ip" ipda=128.1.0.231/32
create class=40 prot="ip" ipda=128.1.0.18/32
create class=50 prot="ip" ipda=128.1.0.4/32
```

Create metrics to mark packets as conformant or non-conformant to the bandwidth limits.

Understanding the meaning of Min and Max parameters

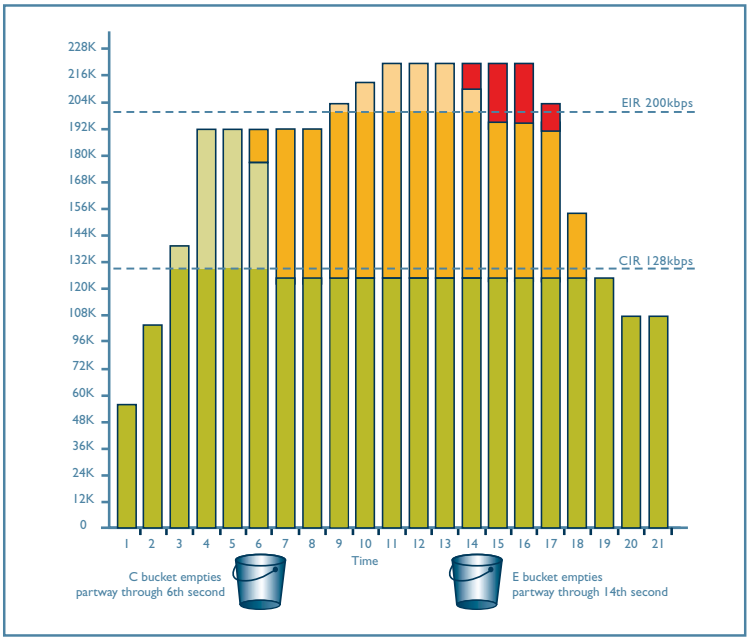
It possible to think "For traffic class 1 (maintenance) I want to guarantee at least 102 Kbps (20% of 512, and I want to allow the traffic to use up to a maximum of 512Kbps (the maximum rate of the WAN link)", and so configure a meter like:

```
cre sqos met=1 desc=maintenace min=102kbps max=512kbps
```

But, unfortunately, that would be based on a wrong understanding of the meaning of the **min** and **max** parameters. It is important to understand the min and max parameters on a meter are equivalent to the parameters that have been used for many years on Frame Relay: Committed Information Rate (CIR) and Excess Information Rate (EIR).

So, the min and max parameters on a meter are not the minimum and maximum bandwidths the associated traffic class will be allowed to use. But, what exactly are the CIR and the EIR?

CIR and EIR explained



In the diagram above you will see that the metering process marks all packets up to the CIR as being Green, then, most of the packets between CIR and EIR are marked Yellow, and then **most** packets above the EIR are marked RED. So, the CIR and the EIR are parameters used in the metering process to decide which packets are conformant to the bandwidth limit (Green); partially conformant to the bandwidth limit (Yellow); or Non-conformant to the bandwidth limit (Red).

You will note above that it is stated that most of the packets between CIR and EIR are Yellow. You might think, "Why is it most packets, why is it not all packets? Well, the answer to that question is that the metering also includes a burst parameter. So, if the data rate makes a brief burst above the CIR, it is still considered conformant, because not all traffic is perfectly smooth, sometimes it has bursts, but it makes no sense to not deliver the packets belonging to a brief burst.

Bandwidth limits and guarantees

OK, so, what do these colours (Green, Yellow, Red) mean? And, why are we talking about bandwidth limits when the customer was also interested in bandwidth guarantees?

Well, lets answer the second question first. Actually, guaranteed bandwidth is a myth! It is not actually possible to guarantee bandwidth on Ethernet. All you can do is stop other traffic flows from stealing bandwidth from the flow that is being guaranteed. So, you set a maxbandwidth on each of the flows you are interested in. Then, you use RED curves to shape the flows back to their assigned bandwidth when things start to get congested. So, each one is not stealing bandwidth from the others. But, if there is no congestion, the flows are not limited down to their maxbandwidth.

The answer to the first question is: the colours are used to decide which packets to try always to guarantee to deliver (Green), which to try hard to deliver (Yellow) and which to drop as soon as congestion starts (Red).

The next question to consider is "What is the process that looks at the colours of the packets, and decides to deliver the green ones, maybe drop the yellow ones, and be very keen to drop the red ones?" The answer is Random Early Dropping (RED) curves, which we will look at further down.

At this point, lets write down the configuration for the meters:

```
cre sqos met=1 desc=maintenace min=102kbps max=150kbps minburst=20kb
  maxburst=20kb type=TRTCM

cre sqos met=2 desc=Intranet min=77kbps max=120kbps minburst=15kb
  maxburst=20kb type=TRTCM

cre sqos met=3 desc=messagerie min=128kbps max=170kbps minburst=20kb
  maxburst=20kb type=TRTCM

cre sqos met=4 desc=Internet min=51kbps max=90kbps minburst=10kb
  maxburst=15kb type=TRTCM

cre sqos met=5 desc=SQL min=51kbps max=90kbps  minburst=10kb
  maxburst=15kb type=TRTCM

cre sqos met=6 desc=Defaut min=102kbps max=150kbps minburst=20kb
  maxburst=20kb type=TRTCM
```

Why choose the particular values for the parameters in these meters?

Well, you will see that the min values are exactly the values that match the percentage bandwidths the customer would like to guarantee for each traffic class - these are the CIR values. So for each of the traffic types, the packets up to the min will be marked green, and so will be delivered, even when things are very busy.

The max values have been chosen to be about 40Kbps to 60Kbps bigger than the min values. This is so that when things are starting to get congested, the traffic classes that are transmitting just a bit above their CIR will have there excess packets marked Yellow, while the traffic classes that are well above their limits will have a lot of packets marked RED. So, it is the traffic classes that are well above their limit that will have the most packets discarded.

The burst values have been chosen to be about 20% of the min values. Therefore, if traffic classes have a brief burst that is equivalent of 0.2 seconds of transmitting at their CIR, then the burst will be allowed through. Also, because of the way that the metering algorithm works, you need to define some burst.

Scheduling

The next matter to consider is how to assign priorities to the different traffic classes. If you use strict priority scheduling on the classes, you will not actually achieve bandwidth guarantee. This is because the class with the top priority will always get to transmit a packet when it has a packet to transmit. Given that RED curves work by dropping packets when queues start to fill up with packets, then the RED algorithm will never come into play on the top-priority queue, as that queue will never fill with packets. If the traffic class with top priority is sending packets at 512Kbps, then none of the other traffic classes will have an opportunity to transmit anything. So, the lower-priority traffic classes will have no bandwidth guarantee.

Instead, use Weighted Round Robin scheduling, but give higher weight to the more important traffic classes, so that they get their packets through more quickly on average.

The configuration of the traffic classes would be:

```
create sqos trafficclass=1 weight=4 meter=1
create sqos trafficclass=2 weight=3 meter=2
create sqos trafficclass=3 weight=2 meter=3
create sqos trafficclass=4 weight=2 meter=4
create sqos trafficclass=5 weight=1 meter=5
create sqos trafficclass=6 weight=1 meter=6
```

▶ Create a policy and add it to an interface:

Then, create the policy, add the traffic classes to the policy, add the classifiers to the traffic classes, and apply the policy to an interface:

```
create sqos policy=1
add sqos policy=1 tr=1,2,3,4,5
set sqos policy=1 defaulttrafficclass=6
add sqos trafficclass=1 class=10,11,12,13
add sqos trafficclass=2 class=20,21
add sqos trafficclass=3 class=30
add sqos trafficclass=4 class=40
add sqos trafficclass=5 class=50
set sqos int=eth0 outpolicy=1
```

▶ Set RED curves on the Traffic classes:

Next, we need to set RED curves on the Traffic classes.

The idea of the RED curves is that when there is congestion on the eth0 interface, there will be packets queueing up on the traffic classes. So, as the queues start to fill, the RED curves will drop packets out of the queues. They will drop red packets even when the queues are short. Then, as the queues get longer, they will drop Yellow packets. Finally, if the congestion is very severe and the queues get very long, they will drop Green packets.

The predefined RED curves are described on page 2-26 of the SW release note:

http://www.alliedtelesyn.co.nz/documentation/relnotes/pdf/c613-10423-00-A.pdf.

▶ Apply a medium RED curve to each traffic class:

Let us apply a medium RED curve on every traffic class:

```
set sqos traff=1 red=1
set sqos traff=2 red=1
set sqos traff=3 red=1
set sqos traff=4 red=1
set sqos traff=5 red=1
set sqos traff=6 red=1
```

Now, the WAN link from the customer's premises has bandwidth 512K. The Ethernet port on the router can transmit up to 100Mbps. So, it would be easy for the traffic from the Ethernet of the router to easily oversubscribe the WAN link. Therefore, the router needs to be constrained to transmit data no faster than the bandwidth of the WAN link. This is achieved by configuring a virtual bandwidth on the policy.

▶ Set a virtual bandwidth on the policy:

```
set sqos poli=1 virt=512kbps
```

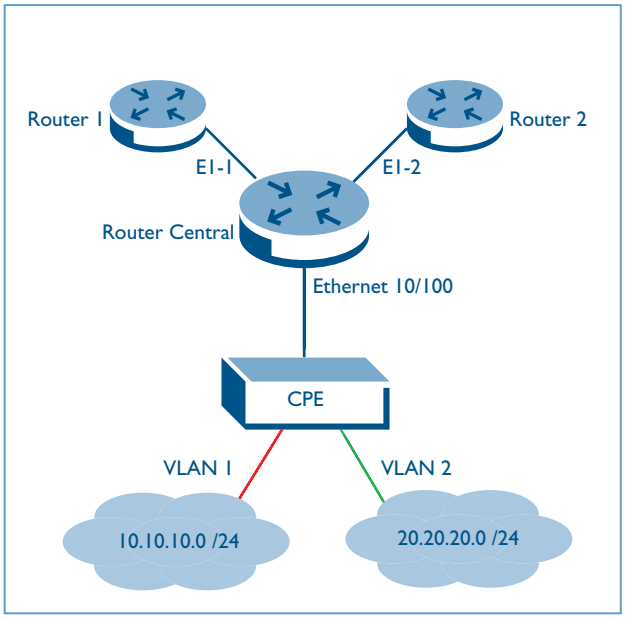
The full configuration

```
create class=10 ippr=tcp tcpd=5631-5632

create class=11 ippr=udp udpd=5631-5632

create class=12 tcpd=4899

create class=13 udpd=4899

create class=20 prot="ip" ipda=128.1.0.75/32

create class=21 prot="ip" ipda=192.168.254.1/32

create class=30 prot="ip" ipda=128.1.0.231/32

create class=40 prot="ip" ipda=128.1.0.18/32

create class=50 prot="ip" ipda=128.1.0.4/32


ena sqos

cre sqos met=1 desc=maintenace ty=TRTCM min=102kbps max=150kbps
  minbu=20kB maxbu=20kB

cre sqos met=2 desc=Intranet ty=TRTCM min=77kbps max=120kbps
  minbu=15kB maxbu=20kB

cre sqos met=3 desc=messagerie ty=TRTCM min=128kbps max=170kbps
  minbu=20kB maxbu=20kB

cre sqos met=4 desc=Internet ty=TRTCM min=51kbps max=90kbps maxbu=15kB

cre sqos met=5 desc=SQL ty=TRTCM min=51kbps max=90kbps maxbu=15kB

cre sqos met=6 desc=Defaut ty=TRTCM min=102kbps max=150kbps minbu=20kB
  maxbu=20kB

cre sqos tr=1 wei=4 red=1 met=1

cre sqos tr=2 wei=3 red=1 met=2

cre sqos tr=3 wei=2 red=1 met=3

cre sqos tr=4 wei=2 red=1 met=4

cre sqos tr=5 wei=1 red=1 met=5

cre sqos tr=6 wei=1 red=1 met=6

cre sqos poli=1 def=6 virt=512kbps

add sqos poli=1 tr=1,2,3,4,5

add sqos tr=1 class=10,11,12,13

add sqos tr=2 class=20,21

add sqos tr=3 class=30

add sqos tr=4 class=40

add sqos tr=5 class=50

set sqos int=eth0 ou=1
```

Example 5: Marking Packets

Scenario

The CPE is connected according to the following diagram:



Requirements

There are four different scenarios, each with a different set of requirements:

Scenario 1

1. Packets arriving on VLAN1 are marked with priority=5
2. Packets arriving on VLAN2 are marked with priority=3

Scenario 2

1. Packets with source address 10.10.10.0/24 are marked with priority=5
2. Packets with source address 20.20.20.0/24 are marked with priority=3

Scenario 3

1. Packets destined for Router 1 (30.30.30.0/24) are marked with priority=5
2. Packets destined for Router 1 (40.40.40.0/24) are marked with priority=3

Scenario 4

1. Identify ports being used by VoIP and mark packets to those ports with priority=5
2. Identify ports not being used by VoIP and mark packets to those ports with priority=3

Solution for scenario 1

The following shows the configuration required on the router to achieve these requirements (when using an Ethernet interface as the interface sending out tagged packets):

```
# Assumes that appropriate VLANs are already created.
    enable ip
    add ip int=vlan1 ip=10.10.10.254 mask=255.255.255.0
    add ip int=vlan2 ip=20.20.20.254 mask=255.255.255.0
    add ip int=eth0 ip=80.80.80.254 mask=255.255.255.0 vl=3
# The vl parameter sets the eth0 interface to use VLAN tagging, putting
# VID=3 into the tags.
    create class=1 svlan=1
    create class=2 svlan=2
# These will match all packets arriving on VLAN 1 or 2.
    ena sqos
    cre sqos trafficclass=1 remarkv=5
    cre sqos trafficclass=2 remarkv=3
# Packets arriving on VLAN 1 will be marked as priority 5 in the 802.1p
   #user priority field of the VLAN tag. Packets arriving on VLAN 2 will
   #be marked as priority 3.
    cre sqos poli=1
    add sqos poli=1 trafficclass=1,2
    add sqos trafficclass=1 class=1
# Traffic from VLAN 1, which match classifier 1, will be associated to
# traffic class 1
    add sqos trafficclass=2 class=2
# Traffic from VLAN 2, which match classifier 2, will be associated to
# traffic class 2
    set sqos int=eth0 outpolicy=1
# Set policy 1 to be the egress QoS policy on interface eth0
```

Solution for scenario 2

```
# Assumes that appropriate VLANs are already created.
    enable ip
    add ip int=vlan1 ip=10.10.10.254 mask=255.255.255.0
    add ip int=vlan2 ip=20.20.20.254 mask=255.255.255.0
    add ip int=eth0 ip=80.80.80.254 mask=255.255.255.0 vl=3
# The vl parameter sets the eth0 interface to use VLAN tagging, putting
   VID=3 into the tags.
    create class=1 ipsa=10.10.10.0/24
    create class=2 ipsa=20.20.20.0/24
# These will match sessions based on source address.
    ena sqos
    cre sqos tr=1 remarkv=5
    cre sqos tr=2 remarkv=3
# Packets from 10.10.10.0/24 will be marked as priority 5 in the 802.1p
   user priority field of the VLAN tag.  Packets from 20.20.20.0/24 will
   be marked as priority 3.
    cre sqos poli=1
    add sqos poli=1 tr=1,2
    add sqos tr=1 class=1
# Classifier 1 (10.10.10.0/24), will be associated to traffic class 1
    add sqos tr=2 class=2
# Classifier 2 (20.20.20.0/24), will be associated to traffic class 2
    set sqos int=eth0 ou=1
# Set policy 1 to be the egress QoS policy on interface eth0
```

Solution for scenario 3

```
# Assumes that appropriate VLANs are already created.
    enable ip
    add ip int=vlan1 ip=10.10.10.254 mask=255.255.255.0
    add ip int=vlan2 ip=20.20.20.254 mask=255.255.255.0
    add ip int=eth0 ip=80.80.80.254 mask=255.255.255.0 vl=3
# The vl parameter sets the eth0 interface to use VLAN tagging, marking as
   VLAN 3.
    add ip rou=0.0.0.0 mask=0.0.0.0 int=eth0 next=80.80.80.8
    create class=1 ipda=30.30.30.0/24# packets for router I
    create class=2 ipda=40.40.40.0/24# packets for router II
# These will match sessions based on destination address.
    ena sqos
    cre sqos tr=1 remarkv=5
    cre sqos tr=2 remarkv=3
# Packets to 30.30.30.0/24 will be marked as priority 5 in the 802.1p user
   priority field of the VLAN tag.  Packets to 40.40.40.0/24 will be
   marked as priority 3.
    cre sqos poli=1
    add sqos poli=1 tr=1,2
    add sqos tr=1 class=1
# Classifier 1 (da 30.30.30.0/24), will be associated to traffic class 1
    add sqos tr=2 class=2
# Classifier 2 (da 40.40.40.0/24), will be associated to traffic class 2
    set sqos int=eth0 ou=1
# Set policy 1 to be the egress QoS policy on interface eth0
```

Solution for scenario 4

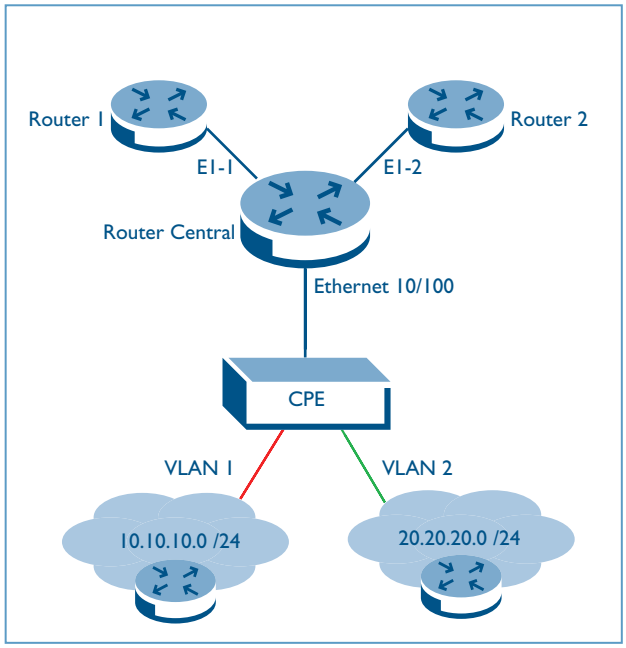
Here is a config we are using for SW QoS to use dynamic application recognition to automatically prioritise packets that are part of SIP-signalled VoIP sessions:

```
    ena sqos
    create sqos dar=0 prot=sip
#This will match all sessions initiated by SIP signaling
    create class=1 udpdport=5060
# The DAR does not actually match the signaling traffic itself, so we need
   a separate classifier to match on the signaling traffic
    add sqos int=swi0 dar=0
# The DAR has to be put onto the incoming switch instance so it will check
   for VoIP packets arriving via the switch ports.
    cre sqos tr=1 remarkv=5 prio=14
    cre sqos tr=2 remarkv=5 prio=15
# Prioritise the SIP signaling traffic above the VoIP data packets.
    cre sqos tr=3 remarkv=3 weight=0
    cre sqos poli=1
    set sqos poli=1 def=3
# Set traffic class 3 to be the default traffic class on the policy, the
   class that matches "everything else"
    add sqos poli=1 tr=1,2
    add sqos tr=1 dar=0
# The packets that DAR 0 recognises as voice packets will be associated to
   traffic class 1
    add sqos tr=2 class=1
# The SIP signaling packets, which match classifier 1, will be associated
   to traffic class 2
    set sqos int=eth0 ou=1
# Set policy 1 to be the egress QoS policy on interface eth0
```

Example 6: Policing / Shaping / Dropping

Scenario

The CPE is connected according to the following diagram:



Requirements

1. Traffic coming from gear in the subnet 10.10.10.0/24 will be marked with DSCP=10 up to a limit of 1024Kbps. Traffic above 1024Kbps will be marked with DSCP=14.
2. Send traffic in from the 10.10.10.0/24 subnet at 2048 Kbps.
3. Traffic coming from gear in the subnet 20.20.20.0/24 will be marked with DSCP=10 up to a limit of 1024Kbps. Traffic above 1024Kbps will mostly be dropped, or possibly marked with DSCP=14 according to which bandwidth class (2 or 3) each packet is marked with. In practice, only those packets marked as bandwidth class 3 are dropped).

Solution

```
# Assumes that appropriate VLANs are already created.
    enable ip
    add ip int=vlan1 ip=10.10.10.254 mask=255.255.255.0
    add ip int=vlan2 ip=20.20.20.254 mask=255.255.255.0
    add ip int=eth0 ip=80.80.80.254 mask=255.255.255.0 vlantag=3
# The vlantag parameter sets the eth0 interface to use VLAN tagging,
   marking as VLAN 3.
    create class=1 prot="ip" ipsa=10.10.10.0/24
```

```
    create class=2 prot="ip" ipsa=20.20.20.0/24
# These will match sessions based on source address.
    ena sqos
    cre sqos met=1 desc=ratelimit max=1.024Mbps minbu=3K maxbu=3K
# This designates a single-rate metering maxbandwidth of 1.024Mbps.  The
   #burst sizes have been set low to ensure excessive traffic is
   #immediately classed as bwc3, therefore allowing no tolerance before
   #dropping traffic from 20.20.20.0/24.
# Note that burst size settings need to be bigger than the size of your
   #biggest expected packets - otherwise those large packets will always
   #be dropped.  (The default size is 10kB).
    cre sqos dscpm=1 desc="remark traffic above 1024kbps"
    set sqos dscpm=1 ta=rem dscp=0-63 bwc=1 newd=10
    set sqos dscpm=1 ta=rem dscp=0-63 bwc=2 newd=14
    set sqos dscpm=1 ta=rem dscp=0-63 bwc=3 newd=14
# Using a DSCP Map definition we can 're-mark' new DSCP values based on the
   #measured metering bandwidth classes.
    cre sqos tr=1 met=1 rem=USEDSCPMAP
    cre sqos tr=2 met=1 rem=USEDSCPMAP bwclass3action=drop
# Two traffic classes are defined.  Metering is assigned to both, and will
# remark using the DSCP map definition.  The second traffic class will
   drop
# non-conformant packets.
    cre sqos poli=1 dscpm=1
# The DSCP Map must be assigned to the SQoS policy.
    add sqos poli=1 tr=1,2
# Both traffic classes are assigned to the SQoS policy.
    add sqos tr=1 class=1
    add sqos tr=2 class=2
# Classifier 1 (10.10.10.0/24) is associated to traffic class 1.
# Classifier 2 (20.20.20.0/24) is associated to traffic class 2.
    set sqos int=eth0 ou=1
# Set policy 1 to be the egress QoS policy on interface eth0
```

Example 7: Bandwidth limiting and VoIP prioritisation

Requirements

The customer requirements are as follows:

1. Internet access (E1/T1) using an AR410
2. AR410 WAN interface speed of E1 or T1.
3. Bandwidth allocation of min/max 450/500Kbps to VoIP (VoIP to Internet).
4. Other application to use remaining bandwidth. (E1 - 1500kbps, or T1 - 1000kbps)

Solution

Start by creating classifiers for the VoIP content, and the VoIP signalling:

▶ Create classifiers for the VoIP content and the VoIP signalling:

```
create class=1 ippr=udp udpd=16300-16320
create class=2 udpd=5060
```

To do the bandwidth limiting, we need to create a meter. But, let us just think to ourselves - what does a meter do?

Well, a meter assigns bandwidth class values (colours) to packets. For data that is under the min bandwidth, the packets are marked green; for data between the min and max, packets are marked yellow; for data above the max bandwidth, packets are marked red. So, a meter colours packets, but it does NOTHING ELSE.

A meter will have assigned colours to packets, but it will not have done any bandwidth limiting or bandwidth guarantee.To actually limit bandwidth, we need to configure something that looks at the colours of the packets, and makes forwarding decisions based on the colours.

▶ Configure bandwidth limiting:

There are two possible 'somethings' that we could configure:

a We could configure a traffic class to drop all packets that have been marked red:

```
set trafficclass=<name> BWClass3action=drop
```

This is called **policing**, and it is used to keep a traffic class strictly below a particular bandwidth limit.

b We could configure RED curves that drop packets when congestion starts to occur - they can be aggressive in dropping red packets, not so aggressive in dropping yellow packets, and lenient in dropping green packets. This is called **shaping** - it allows a traffic class to use all the available bandwidth when no-one else wants it, but makes the traffic class drop back to its allocated bandwidth when congestion starts to occur.

In this case we want to use option (b) shaping. So, let us now proceed to create a configuration that performs shaping.

At this point, let us discuss the choice of parameter values on the meter.

The fact is that we are interested in limiting the traffic class to 500Kbps when the link is busy, so we need to set 500Kbps to be the CIR (the limit that the traffic class is held to when things are busy). In fact the *guaranteed* traffic rate of 450 Kbps is not actually relevant to the meter (we will discuss this further down).

▶ Create a one-rate meter:

So, we can actually create a one-rate meter instead of a two-rate meter.

```
cre sqos met=1 ty=SRTCM max=500kbps
```

This will have the effect that there will be really very few yellow packets created by this meter. Packets below 500Kbps will be marked green, and be transmitted even when the link is busy. Packets above 500Kbps will be marked red, and will mostly be dropped when the link is busy.

Now, you are probably still thinking “Yes, but what about the guarantee of 450Kbps".

Well, actually, guaranteed bandwidth is a myth! It is not actually possible to guarantee bandwidth in IP. All you can do is stop other traffic flows from stealing bandwidth from the flow that is being guaranteed. So, you set a maxbandwidth on each of the flows you are interested in.

Then, you use RED curves to shape the flows back to their assigned bandwidth when things start to get congested. So, each one is not stealing bandwidth from the others. BUT, if there is no congestion, the flows are not limited down to their maxbandwidth

That means that to guarantee 450 Kbps to VoIP, we need to *limit* all other applications to T1-450 kbps or E1-450kbps.

So, we need to make a meter for 'all other applications'.

To make sure it is brought under control before it gets too close to stealing bandwidth from VoIP, the meter for ‘all other applications' should be a 2-rate meter, with a CIR rather below the max limit, so that packets start to be marked yellow a bit before the 'all other applications' traffic starts stealing bandwidth from VoIP.

▶ Create a meter for the E1 link:

So, for the E1 link, we make this meter:

```
create sqos meter=2 ty=trtcm min=1400kbps  max=1598kbps
```

Now that we have created this meter, what traffic class do we apply it to? Well, we need to create a default traffic class (i.e. a traffic class that matches 'all other traffic') and apply the meter to that class:

▶ Create a default traffic class:

```
create sqos trafficclass=3 meter=2
set sqos poli=1 defaulttrafficclass=3
```

Then we need to apply RED curves to the traffic classes. The idea of the RED curves is that when there is congestion on the eth0 interface, there will be packets queueing up on the traffic classes. So, as the queues start to fill, the RED curves will drop packets out of the queues. They will drop RED packets even when the queues are short. Then, as the queues get longer, they will drop yellow packets. Finally, if the congestion is very severe and the queues get very long, they will drop green packets.

The predefined RED curves are described on page 2-26 of the SW release note:

http://www.alliedtelesyn.co.nz/documentation/relnotes/pdf/c613-10423-00-A.pdf.

▶ Apply a RED curve on the traffic classes:

Let's apply a medium RED curve on both traffic classes:

```
set sqos traff=1 red=1
set sqos traff=3 red=1
```

We are very near to the end of our configuration, not far to go now...

Let us now look at the details of traffic class 2. The fact is that this traffic class should be given the highest priority. The VoIP **signalling** traffic is not high-volume traffic, just a few packets at the start and finish of every call. But, if the signalling packets are lost, then the call will fail. If you lose some voice packets, then you will have some bad sound on the call, but if you lose the signalling packets, you have not call happening at all. So, the signalling packets must have the highest priority, and have no bandwidth limit:

▶ Apply a priority to traffic class 2:

```
cre sqos tr=2 prio=15
add sqos tr=2 class=2
```

What should we do about priority or weighting on the traffic classes 1 and 3? The fact is that you cannot use strict priority on these classes, because then you would not actually achieve bandwidth guarantee. This is because the class with the higher priority will always get to transmit a packet when it has a packet to transmit. So, if the traffic class with higher priority is sending packets at 2 Mbps, then none of the other traffic class will have an opportunity to transmit anything. So, the lower-priority traffic classes will have no bandwidth guarantee. I.e, if you gave traffic class 1 a priority of 14 and the default traffic class a priority of 13, then if the VoIP was very busy, the rest of the applications would not get any bandwidth.

Scheduling

So, instead, use Weighted Round Robin scheduling, but give higher weight to the more important traffic class, so that it get its packets through more quickly on average.

▶ Apply a weighting to traffic classes 1 and 3:

```
set trafficclass=1 weight=10
set trafficclass=3 weight=8
```

Full configuration

```
create class=1 ippr=udp udpd=16300-16320
create class=2 udpd=5060

ena sqos

cre sqos meter=1 type=SRTCM  max=500kbps
create sqos meter=2 type=trtcm min=1400kbps  max=1598kbps

cre sqos trafficclass=1 meter=1 weight=10 red=1
cre sqos trafficclass=2 prio=15
cre sqos trafficclass=3 met=2 weight=8 red=1
cre sqos poli=1 defaulttrafficclass=3
add sqos poli=1 tr=1,2
add sqos trafficclass=1 class=1
add sqos trafficclass=2 class=2
set sqos int=ppp0 outpolicy=1
```

USA Headquarters | 19800 North Creek Parkway | Suite 200 | Bothell | WA 98011 | USA | T: +1 800 424 4284 | F: +1 425 481 3895
European Headquarters | Via Motta 24 | 6830 Chiasso | Switzerland | T: +41 91 69769.00 | F: +41 91 69769.11
Asia-Pacific Headquarters | 11 Tai Seng Link | Singapore | 534182 | T: +65 6383 3832 | F: +65 6383 3830
www.alliedtelesyn.com

C613-16063-00 REV A