version 4

IEWB-RS Technology Labs

QoS

Brian Dennis, CCIE # 2210 (R&S / ISP Dial / Security / Service Provider)
Brian McGahan, CCIE# 8583 (R&S / Service Provider)

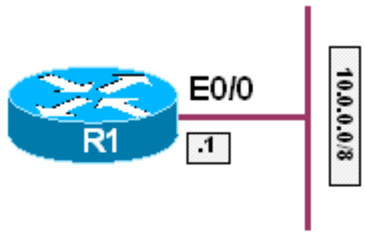
Copyright Information

Disclaimer

Legacy Custom Queueing

Objective: Configure custom queueing on R1 so that traffic leaving its Ethernet interface is guaranteed the following amount of bandwidth

- HTTP - 50%
- SMTP - 20%
- NNTP - 10%
- Other - 20%



Directions

- Configure R1's Ethernet interface with the IP address 10.0.0.1/8
- Create custom queue list 1
- Assign HTTP traffic to be in queue 1
- Assign SMTP traffic to be in queue 2
- Assign NNTP traffic to be in queue 3
- Assign all other traffic to be in queue 4
- Allocate the byte counts for queues 1, 2, 3 and 4 in a ratio of 5:2:1:2
- Apply the custom queue list to the Ethernet interface

Ask Yourself

- What is the legacy custom queue used to accomplish?
- How do I define what traffic is matched by the individual queues?
- How do I assign a byte count to these queues?
- Does it matter what specific byte count should I use?
- How do I apply the list to the interface?
- What direction is the list applied in?

Step-by-Step Configuration

1. Configure the IP address on the Ethernet interface of R1

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#interface ethernet0/0
R1(config-if)#ip address 10.0.0.1 255.0.0.0
```

2. Create the custom queue list and assign the protocol definitions

```
R1(config-if)#queue-list 1 protocol ip 1 tcp www
R1(config)#queue-list 1 protocol ip 2 tcp smtp
R1(config)#queue-list 1 protocol ip 3 tcp nntp
R1(config)#
```

3. Assign the default queue

```
R1(config)#queue-list 1 default 4
R1(config)#
```

4. Assign the byte-counts in a ratio of 5:2:1:2

```
R1(config)#queue-list 1 queue 1 byte-count 5000
R1(config)#queue-list 1 queue 2 byte-count 2000
R1(config)#queue-list 1 queue 3 byte-count 1000
R1(config)#queue-list 1 queue 4 byte-count 2000
```

5. Apply the queue-list

```
R1(config)#interface ethernet0/0
R1(config-if)#custom-queue-list 1
R1(config-if)#no shut
R1(config-if)#end
R1#
```

Final Configuration

```
R1:
interface Ethernet0/0
 ip address 10.0.0.1 255.0.0.0
 custom-queue-list 1
!
queue-list 1 protocol ip 1 tcp www
queue-list 1 protocol ip 2 tcp smtp
queue-list 1 protocol ip 3 tcp nntp
queue-list 1 default 4
queue-list 1 queue 1 byte-count 5000
queue-list 1 queue 2 byte-count 2000
queue-list 1 queue 3 byte-count 1000
queue-list 1 queue 4 byte-count 2000
```

Verification

```
R1#show queueing custom
Current custom queue configuration:
```

```
List   Queue  Args
1      4      default
1      1      protocol ip         tcp port www
1      2      protocol ip         tcp port smtp
1      3      protocol ip         tcp port nntp
1      1      byte-count 5000
1      2      byte-count 2000
1      3      byte-count 1000
1      4      byte-count 2000

R1#show interface ethernet0/0
Ethernet0/0 is up, line protocol is up
  Hardware is AmdP2, address is 0030.1969.81a0 (bia 0030.1969.81a0)
  Internet address is 10.0.0.1/8
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters 00:01:57
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: custom-list 1
  Output queues: (queue #: size/max/drops)
     0: 0/20/0 1: 0/20/0 2: 0/20/0 3: 0/20/0 4: 0/20/0
     5: 0/20/0 6: 0/20/0 7: 0/20/0 8: 0/20/0 9: 0/20/0
     10: 0/20/0 11: 0/20/0 12: 0/20/0 13: 0/20/0 14: 0/20/0
     15: 0/20/0 16: 0/20/0
<output omitted>
```

Breakdown

The legacy custom queue is used to create a bandwidth reservation in the output queue of an interface. In order to classify traffic, the first step in configuring the custom queue is to define what traffic belongs to which queue. In the above example this is accomplished by issuing the queue-list 1 protocol command, followed by the protocol stack, the queue number, and the protocol type within the stack. Once the queues are defined, the amount of bandwidth a certain queue is reserved the determined through a relative byte-count ratio. For example, if there are three queues in a custom queue, each with a byte count of 1500 bytes, each queue would be guaranteed bandwidth in a ratio of 1:1:1, or 33% of the total output queue. In the above example, the ratios are based on a total value of 10,000 bytes, with the queues being assigned bandwidth in the ratio of 5:2:1:2, which results in 5000/10000, 2000/10000, 1000/10000, and 2000/10000. The specific total value that is chosen is fairly arbitrary, as the queuing algorithm can go into debt from future intervals if excess bytes are needed to transmit a packet. However, over a long term average, the desired ratio will be achieved.

With the custom queue it is important to note that the behavior of the queuing mechanism only becomes evident once the output queue is congested. For example, suppose that we have three types of traffic, A, B, and C, that are all guaranteed 33% of the output queue. If there is traffic of type A and B waiting to be sent, but no traffic of type C, type A and B are not limited to a maximum of 33%. Instead, classes A, B, and C are guaranteed a minimum of 33% in the case of congestion, but can use excess above that amount if it not utilized by another queue.

Note that when the list is applied to the interface there is no direction option. This is due to the fact that queuing is always outbound.

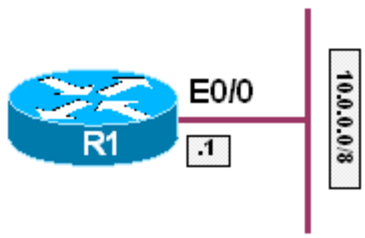
🕮 Recommended Reading

Configuring Custom Queueing

MQC Bandwidth

Objective: Configure the Modular Quality of Service on R1 so that traffic leaving its Ethernet interface is guaranteed the following amount of bandwidth

- HTTP - 50%
- SMTP - 20%
- NNTP - 10%
- Other - 20%



Directions

- Configure R1's Ethernet interface with the IP address 10.0.0.1/8
- Create a class-map named HTTP
- Assign HTTP traffic to this class
- Create a class-map named SMTP
- Assign SMTP traffic to this class
- Create a class-map named NNTP
- Assign NNTP traffic to this class
- Create a policy-map named QoS
- Configure class HTTP in this policy to reserve 50% of the output queue
- Configure class SMTP in this policy to reserve 20% of the output queue
- Configure class NNTP in this policy to reserve 10% of the output queue
- Configure the default class in this policy to reserve 20% of the output queue
- Increase the maximum amount of reservable bandwidth on the Ethernet interface to be 100% of the interface bandwidth
- Apply the policy QoS to the interface

Ask Yourself

- What are the three steps in configuring the MQC?
- Do I need to create access-lists to match the traffic or can I do it directly with NBAR?
- How do I match all other traffic besides HTTP, SMTP, and NNTP?

- What is the difference between a percentage reservation and a reservation in Kbps?
- How much bandwidth can be reserved on the interface by default?
- How do I change this value?
- How do I verify that the policy what applied?

Final Configuration

With NBAR

R1:

```
ip cef
!
class-map match-all NNTP
  match protocol nntp
 class-map match-all HTTP
  match protocol http
 class-map match-all SMTP
  match protocol smtp
!
 policy-map QoS
  class HTTP
   bandwidth percent 50
  class SMTP
   bandwidth percent 20
  class NNTP
   bandwidth percent 10
  class class-default
   bandwidth percent 20
!
interface Ethernet0/0
 ip address 10.0.0.1 255.0.0.0
 max-reserved-bandwidth 100
 service-policy output QoS
```

Without NBAR

R1:

```
 class-map match-all NNTP
  match access-group name NNTP
 class-map match-all HTTP
  match access-group name HTTP
 class-map match-all SMTP
  match access-group name SMTP
!
 policy-map QoS
  class HTTP
   bandwidth percent 50
  class SMTP
   bandwidth percent 20
  class NNTP
   bandwidth percent 10
  class class-default
   bandwidth percent 20
!
interface Ethernet0/0
 ip address 10.0.0.1 255.0.0.0
 max-reserved-bandwidth 100
 service-policy output QoS
!
```

```
ip access-list extended HTTP
 permit tcp any any eq www
 permit tcp any eq www any
!
ip access-list extended NNTP
 permit tcp any any eq nntp
 permit tcp any eq nntp any
!
ip access-list extended SMTP
 permit tcp any any eq smtp
 permit tcp any eq smtp any
```

Verification

Without NBAR

```
R1#show policy-map interface ethernet0/0
 Ethernet0/0

  Service-policy output: QoS

    Class-map: HTTP (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group name HTTP
      Queueing
        Output Queue: Conversation 265
        Bandwidth 50 (%)
        Bandwidth 5000 (kbps) Max Threshold 64 (packets)
        (pkts matched/bytes matched) 0/0
        (depth/total drops/no-buffer drops) 0/0/0

    Class-map: SMTP (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group name SMTP
      Queueing
        Output Queue: Conversation 266
        Bandwidth 20 (%)
        Bandwidth 2000 (kbps) Max Threshold 64 (packets)
        (pkts matched/bytes matched) 0/0
        (depth/total drops/no-buffer drops) 0/0/0

    Class-map: NNTP (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: access-group name NNTP
      Queueing
        Output Queue: Conversation 267
        Bandwidth 10 (%)
        Bandwidth 1000 (kbps) Max Threshold 64 (packets)
        (pkts matched/bytes matched) 0/0
        (depth/total drops/no-buffer drops) 0/0/0

    Class-map: class-default (match-any)
      10 packets, 1208 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
      Queueing
        Output Queue: Conversation 268
```

```
        Bandwidth 20 (%)
        Bandwidth 2000 (kbps) Max Threshold 64 (packets)
        (pkts matched/bytes matched) 2/728
        (depth/total drops/no-buffer drops) 0/0/0

R1#show queueing interface ethernet0/0
Interface Ethernet0/0 queueing strategy: fair
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/256 (active/max active/max total)
     Reserved Conversations 4/4 (allocated/max allocated)
     Available Bandwidth 0 kilobits/sec

With NBAR

R1#show policy-map interface ethernet0/0 | include (Class-map|Match)
    Class-map: HTTP (match-all)
      Match: protocol http
    Class-map: SMTP (match-all)
      Match: protocol smtp
    Class-map: NNTP (match-all)
      Match: protocol nntp
    Class-map: class-default (match-any)
      Match: any
```

Breakdown

Like the legacy custom queue, the purpose of the bandwidth statement in the modular quality of service is to reserve bandwidth in the output queue. This queueing strategy only comes into effect when there is congestion in the output queue, as if the queue isn't full there isn't any reason to reserve bandwidth. The main differences between the legacy custom queue and the bandwidth statement in the MQC is that the bandwidth statement does its reservation either as a percentage of the interface bandwidth or as a value in kilobits per second, as opposed to a ratio. In addition to this, since it is part of the MQC, this type of bandwidth reservation can be combined with other QoS mechanisms in the same direction on the same interface.

The first step in configuring a bandwidth reservation with the MQC is to match the traffic in question. This is accomplished by configuring a class-map. The class map is used to match the class, or type, of traffic that the QoS policy applies to. In the above case, two variations of the configuration are seen. The first method uses Network Based Application Recognition (NBAR) to match the protocol in question. The second method uses extended access-lists to match TCP port numbers. There is no effective difference between these methods, however as we will see in later labs, NBAR has additional functionality to match higher layer information in the packet.

Once the class-maps are defined, the next step is to define the policy-map. The policy-map is used to apply the specific QoS policy to the traffic that was matched in the class-maps. Once the policy-map QoS is created, the previously defined class-maps are referenced, and the bandwidth keyword is issued. This statement configures the reservation in the output queue, and can be configured as a percentage value or an absolute value in Kbps.

Lastly, the policy-map is applied to the interface with the service-policy output QoS keyword. In order to apply this, two additional statements are added, the max-reserved-bandwidth 100 command, and the ip cef command. The specific implications of these statements will be covered in the Advanced Technologies Labs series.

To verify the configuration, the show policy-map interface ethernet0/0 and the show queueing interface ethernet0/0 commands are issued. Note that the effective result of the configuration with and without NBAR is the same, simply the method of accomplishing the end-goal is different.

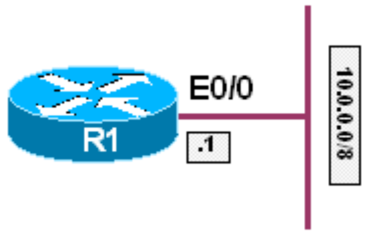
Recommended Reading

Comparing the bandwidth and priority Commands of a QoS Service Policy

Legacy Priority Queueing

Objective: Configure legacy priority queueing on R1 so that traffic leaving its Ethernet interface is serviced in the following manner

- Telnet - High Priority
- HTTP - Medium Priority
- IP - Normal Priority
- Other - Low Priority



Directions

- Configure R1's Ethernet interface with the IP address 10.0.0.1/8
- Create priority list 1
- Assign telnet traffic to the high queue
- Assign web traffic to the medium queue
- Assign all other IP traffic to the normal queue
- Assign all other traffic to the default queue
- Apply the priority list to the Ethernet interface

Ask Yourself

- What is the difference between the custom queue and the priority queue?
- How do I define what traffic is serviced in what order?
- Do I need to assign a bandwidth value to the queues?
- How to I apply the configuration?
- What direction is the configuration applied in?

Final Configuration

R1:

```
interface Ethernet0/0
 ip address 10.0.0.1 255.0.0.0
 priority-group 1
!
priority-list 1 protocol ip high tcp telnet
priority-list 1 protocol ip medium tcp www
priority-list 1 protocol ip normal
priority-list 1 default low
```

Verification

```
R1#show queueing priority
Current DLCI priority queue configuration:
Current priority queue configuration:

List   Queue  Args
1      low    default
1      high   protocol ip          tcp port telnet
1      medium protocol ip          tcp port www
1      normal protocol ip

R1#show queueing interface ethernet0/0
Interface Ethernet0/0 queueing strategy: priority

Output queue utilization (queue/count)
        high/226 medium/0 normal/35 low/8

R1#show interface ethernet0/0
Ethernet0/0 is up, line protocol is up
  Hardware is AmdP2, address is 0030.1969.81a0 (bia 0030.1969.81a0)
  Internet address is 10.0.0.1/8
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:02, output 00:00:00, output hang never
  Last clearing of "show interface" counters 01:37:45
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: priority-list 1
  Output queue (queue priority: size/max/drops):
     high: 0/20/0, medium: 0/40/0, normal: 0/60/0, low: 0/80/0
<output omitted>
```

Breakdown

The legacy priority queue is used to change the order in which traffic exits the interface. This QoS mechanism allows delay sensitive traffic to be preferred over other types of traffic, regardless of the order it was received at the interface for transmission.

The legacy priority queue uses four queue definitions to determine what traffic gets serviced when. These queue are the high queue, the medium queue, the normal queue, and the low queue. Each time a packet is moved from the output queue to the interface for transmission, the high queue is checked for traffic. If there are packets in the high queue they are sent. If there aren't any packets in the high queue, the medium queue is checked. If there is a packet in the medium queue, it is are sent, otherwise, the normal queue is checked. If there is a packet in the normal queue, it is sent, otherwise, the low queue is checked. If there aren't any packets in the low queue, the process starts again. This round-robin sequence occurs for every single packet. Therefore, if there are consistently packets in the upper queues, packets in the lower queues will never get serviced.

To configure the legacy priority queue, issue the priority-list command in global configuration mode, followed by the list number. Next, like the legacy custom queue, issue the protocol keyword, followed by the protocol stack name, such as IP, followed by the queue definition, high, medium, normal, or low. For IP, more granular options can be chosen such as TCP or UDP port numbers, or an access list can be called. Like the custom queue, the priority queue also supports a default queue. This default queue is used for all other traffic that is not explicitly matched. If not manually specified, the default queue is automatically assigned to the normal queue.

Unlike the legacy custom queue, the four priority queues are not assigned a byte count or any type of bandwidth value. Instead, each of the four queues is assigned a queue depth. This queue depth dictates how many packets can be in a particular queue at any given time. If the queue is full and additional packets try to enter, they will be dropped. The size of the queues can be viewed by issuing the show interface command, as seen in the above example. The queue depths can be changed by issuing the queue-limit option of the priority-list statement.

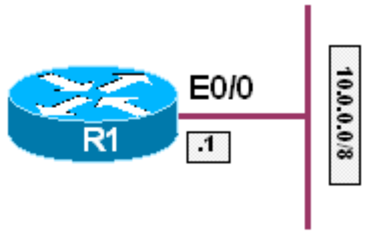
To apply the list, issue the interface level command priority-group followed by the list number. Note that like the legacy custom queue no direction option is applied, as queueing is always outbound.

Recommended Reading

Configuring Priority Queueing

MQC Low Latency Queue

Objective: Configure the Modular Quality of Service on R1 so that all telnet traffic up to 640Kbps is sent first out the Ethernet interface



Directions

- Configure R1's Ethernet interface with the IP address 10.0.0.1/8
- Create a class-map named TELNET
- Assign telnet traffic to this class
- Create a policy-map named QoS
- Configure class TELNET as a priority class for up to 640Kbps
- Apply the policy QoS to the interface

Ask Yourself

- What are the three steps in configuring the MQC?
- Do I need to create access-lists to match the traffic or can I do it directly with NBAR?
- What command is used to configure the Low Latency Queue?
- How does this mechanism differ from the bandwidth keyword?

Final Configuration

With NBAR

```
R1:
ip cef
!
class-map match-all TELNET
  match protocol telnet
!
 policy-map QoS
  class TELNET
   priority 640
!
interface Ethernet0/0
 ip address 10.0.0.1 255.0.0.0
 service-policy output QoS
```

Without NBAR

```
R1:
class-map match-all TELNET
  match access-group name TELNET
```



```
!
 policy-map QoS
  class TELNET
   priority 640
!
interface Ethernet0/0
 ip address 10.0.0.1 255.0.0.0
 service-policy output QoS
!
ip access-list extended TELNET
 permit tcp any any eq telnet
 permit tcp any eq telnet any
```

Verification

```
R1#show policy-map interface ethernet0/0
 Ethernet0/0

  Service-policy output: QoS

    Class-map: TELNET (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: protocol telnet
      Queueing
        Strict Priority
        Output Queue: Conversation 264
        Bandwidth 640 (kbps) Burst 16000 (Bytes)
        (pkts matched/bytes matched) 0/0
        (total drops/bytes drops) 0/0

    Class-map: class-default (match-any)
      15 packets, 909 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
```

Breakdown

The Low Latency Queue (LLQ) is the MQC's implementation of the priority queue. Unlike the legacy priority queue which uses four queue definitions to determine when traffic is serviced, the LLQ uses only one priority queue per QoS policy. Although multiple classes can be assigned to the priority queue, limiting the priority queue to one avoids the issue of starving non-priority traffic that occurs with the legacy priority queue.

Like the bandwidth statement in the MQC, the priority statement is used to create a bandwidth reservation in the output queue in kilobits per second, or as a percentage of the interface bandwidth. The difference between bandwidth and priority however is that the priority keyword is used to move traffic to the front of the output queue to send it before other traffic, and it has a built in policer. What this means is that when the priority class exceeds the specified bandwidth value it is not guaranteed low latency. In addition to this, if congestion occurs and the priority class is in excess of the configured bandwidth value, the excess traffic is dropped. Therefore, the bandwidth statement is used to configure a minimum bandwidth guarantee, while the priority statement is used to configure a maximum bandwidth guarantee.

To configure the priority queue, simply issue the priority command, followed by the bandwidth value in Kbps or a percentage in the policy-map class-map subconfiguration mode. To verify the configuration, the show policy-map interface ethernet0/0 command.

📖 Recommended Reading

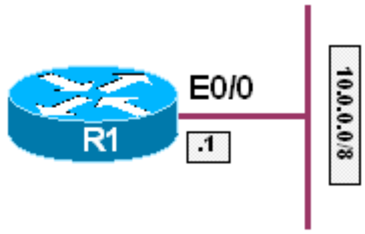
Comparing the bandwidth and priority Commands of a QoS Service Policy

📖 Recommended Reading

Congestion Management Overview: Low Latency Queueing

Legacy Generic Traffic Shaping

Objective: Configure legacy GTS on R1 to limit the output rate on the Ethernet interface to 640Kbps



Directions

- Configure R1's Ethernet interface with the IP address 10.0.0.1/8
- Configure GTS on the Ethernet interface to limit the output rate to 640Kbps
- Use a committed burst value of 80Kbps
- Do not configure excess burst

Ask Yourself

- What is traffic shaping used to accomplish?
- What does the field target bit rate mean?
- What does the field bits per interval sustained mean?
- What does the field bits per interval excess in first interval mean?
- Is shaping applied inbound or outbound? Why?

Final Configuration

```
R1:
interface Ethernet0/0
 ip address 10.0.0.1 255.0.0.0
 traffic-shape rate 640000 80000 0 1000
```

Verification

```
R1#show traffic-shape

Interface   Et0/0
       Access Target    Byte   Sustain   Excess    Interval  Increment Adapt
VC     List   Rate      Limit  bits/int  bits/int  (ms)      (bytes)   Active
-             640000    10000  80000     0         125       10000     -

R1#show traffic-shape statistics
                  Acc. Queue Packets   Bytes     Packets   Bytes     Shaping
I/F               List Depth                     Delayed   Delayed   Active
Et0/0                  0     2060       1693378   1037      1567110   no
```

Breakdown

Traffic shaping is used to slow down the output rate of an interface by buffering traffic that exceeds a configured rate. For example, suppose that the connection to your ISP is through a 10Mbps Ethernet interface, however the provider is configured to drop all traffic it receives above 2Mbps. In this case it would be more advantageous to configure your router to send at 2Mbps instead of having it send at 10Mbps and have the excess traffic dropped.

Legacy generic traffic shaping is controlled by the traffic-shape interface level command. In the above example, all traffic exiting the interface is limited to 640Kbps with the traffic-shape rate 640000 80000 command. This syntax means that the average output rate over a one second period will be no larger than 640000 bits, while this rate is subdivided into smaller intervals in which the rate will not exceed 80000 bits per interval.

This 80,000 value is known as the committed burst, or Bc, while the interval is known as the time committed, or Tc. In other words, Bc is CIR expressed in one Tc interval, while CIR is expressed per second. Specifically the above configuration says that there are eight shaping intervals per second, each of which are 125ms long. If we set our Bc to 40000, it means that there are 16 shaping intervals per second, each of which are 62.5ms long. The size of the Bc will ultimately determine the serialization delay of the interface being shaped, and will be explored in more detail in the Advanced Technologies Labs.

To verify traffic shaping configuration issue the show traffic-shape command in privilege level mode. This output shows the average output rate, the Bc, the Be, and the Tc. The show traffic-shape statistics command gives real-time information on what, if any, traffic has been delayed due to shaping.

🕮 Recommended Reading

[Comparing Traffic Policing and Traffic Shaping for Bandwidth Limiting]

🕮 Recommended Reading

[Configuring Generic Traffic Shaping]

Legacy Frame Relay Traffic Shaping

Objective: Configure FRTS on R1 and R2 to limit the output rate on the Serial interfaces to 640Kbps

Directions

- Configure R1's Serial interface with the IP address 10.0.0.1/8
- Configure R2's Serial interface with the IP address 10.0.0.2/8
- Configure a Frame Relay circuit between R1 and R2 using DLCIs 102 and 201 respectively
- Configure a Frame Relay map-class named FRTS on both R1 and R2
- Configure the class with a CIR of 640Kbps
- Use a committed burst value of 80Kbps
- Do not configure excess burst
- Apply the class to the Serial interfaces attached to the Frame Relay cloud

Ask Yourself

- What is traffic shaping used to accomplish?
- How does FRTS differ from GTS?
- Where are FRTS parameters defined?
- How do I apply the class once the parameters are defined?
- When the class is applied, what circuits does it apply to?

Final Configuration

```
R1:
interface Serial0/0
 ip address 10.0.0.1 255.0.0.0
 encapsulation frame-relay
 frame-relay class FRTS
 frame-relay traffic-shaping
 frame-relay map ip 10.0.0.2 102 broadcast
!
map-class frame-relay FRTS
 frame-relay cir 640000
 frame-relay bc 80000

R2:
interface Serial0/0
 ip address 10.0.0.2 255.0.0.0
 encapsulation frame-relay
 frame-relay class FRTS
```

```
 frame-relay traffic-shaping
 frame-relay map ip 10.0.0.1 201 broadcast
!
map-class frame-relay FRTS
 frame-relay cir 640000
 frame-relay bc 80000
```

Verification

```
R1#show traffic-shape

Interface   Se0/0
       Access Target    Byte   Sustain   Excess    Interval  Increment Adapt
VC     List   Rate      Limit  bits/int  bits/int  (ms)      (bytes)   Active
103           640000    10000  80000     0         125       10000     -
104           640000    10000  80000     0         125       10000     -
105           640000    10000  80000     0         125       10000     -
113           640000    10000  80000     0         125       10000     -
102           640000    10000  80000     0         125       10000     -

R1#show traffic-shape statistics
                  Acc. Queue Packets   Bytes     Packets   Bytes     Shaping
I/F               List Depth                     Delayed   Delayed   Active
Se0/0                  0     0         0         0         0         no
Se0/0                  0     3         102       0         0         no
Se0/0                  0     0         0         0         0         no
Se0/0                  0     0         0         0         0         no
Se0/0                  0     5         520       0         0         no
```

Breakdown

Frame Relay Traffic Shaping (FRTS), like generic traffic shaping, is used to control the output rate on an interface. The main differences between FRTS and GTS is that frame relay traffic shaping has extra provisions to adapt to the traffic conditions of the frame relay cloud, and can be implemented on a per DLCI basis.

FRTS parameters are defined in a Frame Relay map-class, not to be confused with the modular quality of service class-map. In the above example, a map-class named FRTS is created. Next, the traffic shaping parameters, such as the CIR and Bc, are defined with the frame-relay cir and frame-relay bc commands.

Once the parameters are defined, the next step is to enable traffic shaping on the interface. This is accomplished with the interface level command frame-relay traffic shaping. Note that even if traffic shaping parameters are applied to subinterfaces, the command frame-relay traffic-shaping must be applied to the main interface.

Next, the class is applied with either the interface level command frame-relay class, or the DLCI level command class, both followed by the name of the map-class. The difference between the two is that with the frame-relay class command, the class applies to all DLCIs on the main and subinterfaces, where the VC level command class applies only to that circuit, and overrides any previous class defined with the frame-relay class command.

Like GTS, FRTS is verified with the show traffic-shape and show traffic-shape statistics commands in privilege level mode.

🕮 Recommended Reading

Understanding Frame Relay Traffic Shaping

MQC Frame Relay Traffic Shaping

Objective: Configure FRTS on R1 and R2 to limit the output rate on the Serial interfaces to 640Kbps using the MQC

Directions

- Configure R1's Serial interface with the IP address 10.0.0.1/8
- Configure R2's Serial interface with the IP address 10.0.0.2/8
- Configure a Frame Relay circuit between R1 and R2 using DLCIs 102 and 201 respectively
- Configure a policy-map named QoS on R1 and R2
- Configure the default class to shape all traffic to 640Kbps
- Use a committed burst value of 80Kbps
- Do not configure excess burst
- Configure a Frame Relay map-class named FRTS on R1 and R2
- Bind the policy-map QoS to the map-class
- Apply the map-class to the Serial interfaces attached to the Frame Relay cloud

Ask Yourself

- What is Frame Relay traffic shaping used to accomplish?
- How does FRTS differ from GTS?
- How does FRTS in the MQC differ from legacy FRTS?
- Where are FRTS parameters defined?
- How do I apply the parameters once they are defined?
- When the class is applied, what circuits does it apply to?

Final Configuration

R1:
```
policy-map QoS
  class class-default
   shape average 640000 80000 0
!
interface Serial0/0
 ip address 10.0.0.1 255.0.0.0
 encapsulation frame-relay
 frame-relay class FRTS
 frame-relay map ip 10.0.0.2 102 broadcast
!
```

```
map-class frame-relay FRTS
 service-policy output QoS

R2:
policy-map QoS
  class class-default
   shape average 640000 80000 0
!
interface Serial0/0
 ip address 10.0.0.2 255.0.0.0
 encapsulation frame-relay
 frame-relay class FRTS
 frame-relay map ip 10.0.0.1 201 broadcast
!
map-class frame-relay FRTS
 service-policy output QoS
```

Verification

```
R1#show policy-map interface serial0/0
       Serial0/0: DLCI 102 -
        Service-policy output: QoS
          Class-map: class-default (match-any)
            0 packets, 0 bytes
            5 minute offered rate 0 bps, drop rate 0 bps
            Match: any
            Traffic Shaping
                Target/Average   Byte   Sustain   Excess    Interval  Increment
                   Rate           Limit  bits/int  bits/int  (ms)      (bytes)
                 640000/640000    10000  80000     0         125       10000
              Adapt  Queue     Packets   Bytes     Packets   Bytes     Shaping
              Active Depth                         Delayed   Delayed   Active
              -      0         0         0         0         0         no
```

Breakdown

Configuring Frame Relay traffic shaping within the Modular Quality of Service CLI enhances FRTS functionality by allowing different shaping parameters to be configured for different traffic classes on the one of more virtual circuits. Configuring FRTS in the MQC involves many of the same steps as the legacy FRTS.

The first step in configuring FRTS in the MQC is to define the class of traffic that will be shaped. In the above example, all traffic is shaped, therefore no class-map need be defined. Next, the policy-map is defined with the command policy-map QoS in global configuration mode. Next, shaping parameters are applied to the class-default within this policy by issuing the shape average command. Additional functionality of peak and adaptive shaping will be covered in additional labs.

Once the shaping parameters have been assigned, a Frame Relay map-class is created with the command map-class frame-relay FRTS in global configuration mode. From the map-class, the policy-map is then called with the command service-policy output. Although the map-class is not used to define any traffic shaping parameters, this step is still required, as a policy-map can not be directly applied to an individual Frame Relay PVC as a map-class can.

Lastly, the map-class is applied to the interface with the frame-relay class FRTS command. Note that the class can also be applied on a per-VC basis with the VC subcommand class FRTS. When the class is applied to the interface itself it applies to all DLCIs on that interface and any subinterfaces, while the VC subcommand only applies to that circuit. Note that the command frame-relay traffic-shaping is not required when configuring FRTS through the MQC.

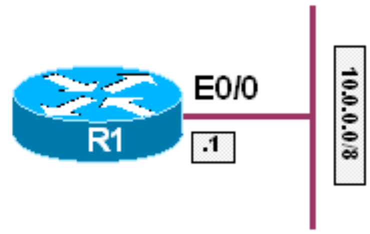
To verify the configuration, issue the show policy-map interface serial0/0 command in privilege level mode. This output shows the shaping parameters on a per-VC as well as per-class basis if configured.

🕮 Recommended Reading

MQC-Based Frame Relay Traffic Shaping

Legacy Committed Access Rate

Objective: Configure legacy Committed Access Rate on R1 to limit the input rate on the Ethernet interface to 640Kbps. All traffic above this rate should be dropped



Directions

- Configure R1's Ethernet interface with the IP address 10.0.0.1/8
- Configure CAR on R1's Ethernet interface to limit all inbound traffic to 640Kbps
- Use a normal burst size of 10000 bytes
- Use an excess burst size of 10000 bytes
- Traffic within this rate should be transmitted
- Traffic outside of this rate should be dropped

Ask Yourself

- What is Committed Access Rate used to accomplish?
- What is the difference between policing and shaping?
- What direction can CAR be applied in?
- How does this differ from the previously seen QoS mechanisms?

Final Configuration

```
R1:
interface Ethernet0/0
 ip address 10.0.0.1 255.0.0.0
 rate-limit input 640000 10000 10000 conform-action transmit exceed-action drop
```

Verification

```
R1#show interface ethernet0/0 rate-limit
Ethernet0/0
  Input
    matches: all traffic
      params:  640000 bps, 10000 limit, 10000 extended limit
      conformed 739 packets, 1113246 bytes; action: transmit
      exceeded 7085 packets, 10726690 bytes; action: drop
      last packet: 12ms ago, current burst: 8636 bytes
      last cleared 00:05:47 ago, conformed 25000 bps, exceeded 246000 bps
```

Breakdown

Committed Access Rate, otherwise known as CAR, rate-limiting, or policing, is used to limit the amount of traffic that can enter or exit an interface. Unlike the other QoS mechanisms we have seen so far, policing can be configured inbound as well as outbound on an interface. While both shaping and policing are used to limit traffic, policing does not buffer traffic that exceeds the rate. With traffic shaping excess traffic is delayed in the shaping buffer on the premise that it will be transmitted at a later time. With policing, excess traffic is not buffered, and is typically just dropped.

To configuring legacy policing, issue the rate-limit command at the interface level followed by the direction. Next, choose the target rate in bits per second. Traffic less than or equal to this rate will have "conformed" to the limit, while traffic above this rate will have "exceeded" the limit. Next, choose the normal burst size in bytes. Like traffic shaping, changing the policing burst size determines how often the router enforces the rate over the second. Note that this option is taken in bytes, while the traffic shaping Bc is taken in bits. Next, choose the excess burst value. Note that excess burst is only configured when the burst size is configured to be greater than the normal burst, which is different from traffic shaping. In the above example both the normal and excess burst are set to 10,000. Therefore, there is effectively no excess burst. For excess burst to be configured in this case it would have to be above 10,000.

Once the target rate and burst values are determined, the next two options are the conform-action and the exceed-action. These values determine what will happen to a packet if it within the rate limit or outside of the rate limit. Options for these actions include to transmit the traffic, drop the traffic, or remark the IP precedence or DSCP values of the traffic.

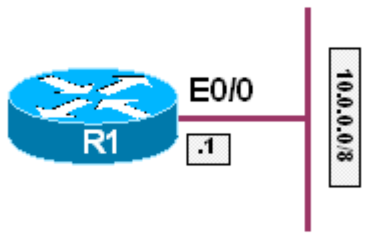
Once the rate-limit statement has been configured, verify the configuration by issuing the show interface ethernet0/0 rate-limit. This output shows how many packets have been sent or received, depending on the configured direction, and how many of these packets have conformed or exceeded in both a packet count value and in bits per second.

Recommended Reading

[Configuring Committed Access Rate]

MQC Policing

Objective: Configure MQC Policing on R1 to limit the input rate on the Ethernet interface to 640Kbps. All traffic above this rate should be dropped



Directions

- Configure R1's Ethernet interface with the IP address 10.0.0.1/8
- Configure a policy-map named QoS
- Configure the default class within this policy to police all traffic to 640Kbps
- Use a normal burst size of 10000 bytes
- Use an excess burst size of 10000 bytes
- Traffic within this rate should be transmitted
- Traffic outside of this rate should be dropped

Ask Yourself

- What is policing used to accomplish?
- What is the difference between legacy CAR and MQC policing?
- What is the difference between policing and shaping?
- What direction can policing be applied in?
- How does this differ from the previously seen QoS mechanisms?

Final Configuration

```
R1:
policy-map QoS
  class class-default
   police cir 640000 bc 10000 be 10000
     conform-action transmit
     exceed-action drop
!
interface Ethernet0/0
 ip address 10.0.0.1 255.0.0.0
 service-policy input QoS
```

Verification

```
R1#show policy-map interface ethernet0/0
```

```
Ethernet0/0

 Service-policy input: QoS

   Class-map: class-default (match-any)
     18178 packets, 27521492 bytes
     5 minute offered rate 593000 bps, drop rate 516000 bps
     Match: any
     police:
         cir 640000 bps, bc 10000 bytes
       conformed 5107 packets, 7731998 bytes; actions:
         transmit
       exceeded 13074 packets, 19794036 bytes; actions:
         drop
       conformed 84000 bps, exceed 516000 bps
```

Breakdown

Traffic policing in the MQC is similar to legacy policing with CAR, with the added advantage of being able to apply granular matches on what traffic will be policed. MQC policing also adds additional functionality with a feature known as the two-rate policer.

To configure MQC policing, first define what type of traffic will be limited with a class-map. In the above example all traffic is policed so no class need be defined. Next, define the policy-map where the policing will be configured. Call the class in question (class-default in the above case) and issue the police command. The options of this command are similar to the legacy rate limit statement, such as the target rate, burst in bytes, excess burst in bytes, but also has additional functionality for policing a percentage of the interface bandwidth.

Once the values are chosen we are brought to the policing sub-configuration mode. In this mode the conform and exceed actions are chosen. Note that the continue option of the legacy rate-limit statement is not available, but additional set options, such as ATM cell loss priority are available.

Once the conform and exceed actions are configured (they default to transmit and drop respectively), apply the policy-map to the interface with the service-policy command, followed by the direction and the policy name. Note that like legacy CAR, MQC policing can be applied both inbound and outbound. However, if policing is configured in a class in tandem with queueing mechanisms such as traffic shaping or bandwidth reservations, the policy can only be applied outbound.

To verify the configuration issue the show policy-map interface Ethernet0/0 command in privilege level mode. Like the legacy CAR, this output shows the configured rates, as well as the actual conform and exceed rates.

Recommended Reading

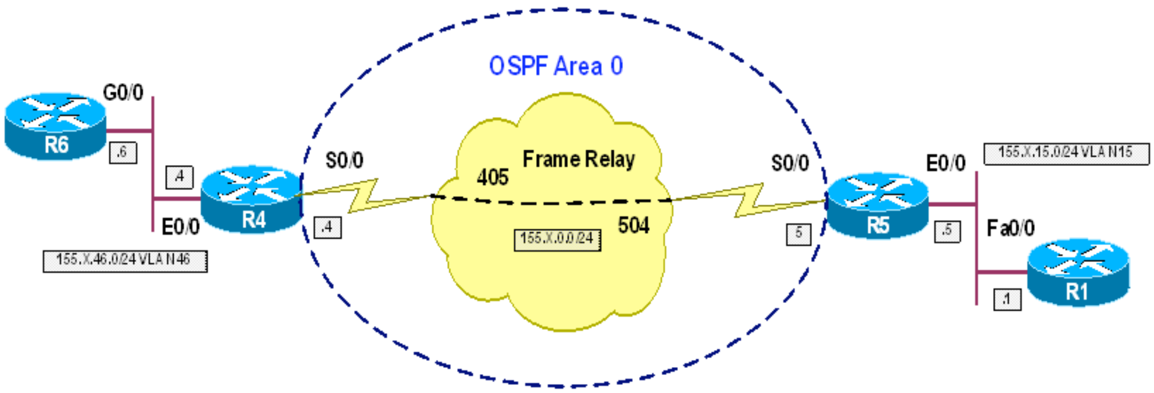
Configuring Traffic Policing

Recommended Reading

Two-Rate Policer

Common Configuration

Objective: Perform configuration steps common for QoS scenarios



Directions

- Configure VTP mode transparent on SW1 and SW2.
- Create VLANs 46 and 15 on SW1 and SW2. Assign the respective switchports to corresponding VLANs:

Catalyst Port	Interface	VLAN
SW1 Fa0/1	R1 – Fa0/0	15
SW1 Fa0/5	R5 – E0/0	15
SW1 Fa0/13	SW2 – Fa0/13	Trunk
SW2 Fa0/4	R4 – E0/0	46
SW2 Fa0/6	R6 – G0/0	46
SW2 Fa0/13	SW1 – Fa0/13	Trunk

- Configure Frame-Relay Interfaces, use physical interface type and static mappings. Map broadcasts on each end
- Configure OSPF area 0 on FR cloud, use broadcast network type on FR interfaces
- Advertise all connected interfaces into OSPF on R4 and R5.
- Configure default route on R1 and R6 to point at R5 and R4 respectively
- Configure RTR (IP SLA Monitor) on R1 and R6. R1 should poll R6, and R6 should respond
- Configure RTP type UDP Echo with destination and source port 16384. Poll every 1 second with timeout 200ms
- Keep 10 statistic distribution buckets with 10ms interval each.

Final Configuration

```
SW1:
vtp mode transparent
vlan 15,46
!
interface Fa 0/1
 switchport host
 switchport access vlan 15
!
interface Fa 0/5
 switchport host
 switchport access vlan 15
!
interface Fa 0/13
 switchport trunk encaps dot1q
 switchport mode trunk

SW2:
vtp mode transparent
vlan 15,46
!
interface Fa 0/4
 switchport host
 switchport access vlan 46
!
interface Fa 0/6
 switchport host
 switchport access vlan 46
!
interface Fa 0/13
 switchport trunk encaps dot1q
 switchport mode trunk

R1:
interface Fa 0/0
 no shutdown
 ip address 155.1.15.1 255.255.255.0
!
ip route 0.0.0.0 0.0.0.0 155.1.15.5
!
rtr 1
 type udpEcho dest-ipaddr 155.1.46.6 dest-port 16384 source-port 16384
 timeout 200
 frequency 1
 distributions-of-statistics-kept 10
 statistics-distribution-interval 10
!
rtr schedule 1 life forever start-time now

R4:
inter ethernet 0/0
 ip address 155.1.46.4 255.255.255.0
 no shut
!
interface Serial 0/0
 encaps frame-relay
 no frame-relay inverse
 ip address 155.1.0.4 255.255.255.0
 frame map ip 155.1.0.5 405 broad
 ip ospf network broadcast
```

```
 no shutdown
!
router ospf 1
 network 0.0.0.0 0.0.0.0 area 0
```

R5:

```
interface Serial 0/0
 encaps frame-relay
 no frame-relay inverse
 ip address 155.1.0.5 255.255.255.0
 frame map ip 155.1.0.4 504 broad
 ip ospf network broadcast
 no shut
!
interface Ethernet 0/0
 no shut
 ip address 155.1.15.5 255.255.255.0
!
router ospf 1
 network 0.0.0.0 0.0.0.0 area 0
```

R6:

```
interface Gig 0/0
 no shutdown
 ip address 155.1.46.6 255.255.255.0
!
ip route 0.0.0.0 0.0.0.0 155.1.46.4
!
ip sla monitor responder
```

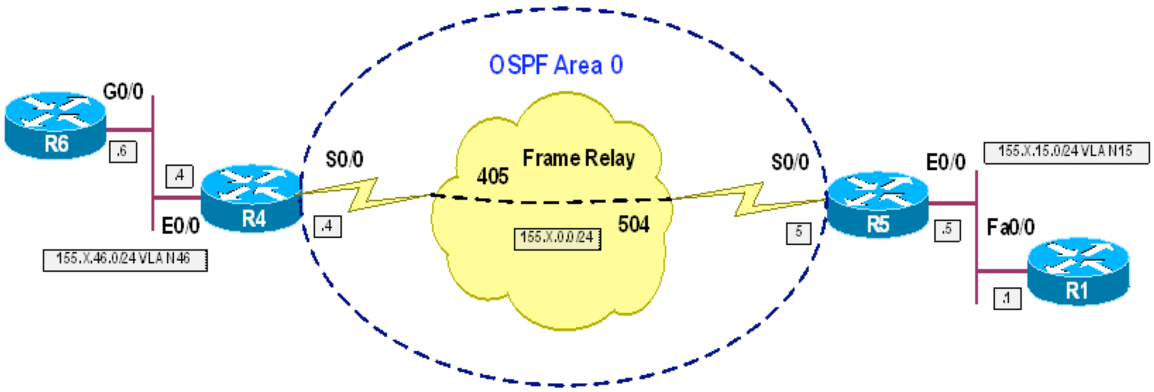
Verification

```
R4#show ip route ospf
     155.1.0.0/24 is subnetted, 3 subnets
O       155.1.15.0 [110/74] via 155.1.0.5, 00:02:59, Serial0/0
R4#

R5#show ip route ospf
     155.1.0.0/24 is subnetted, 3 subnets
O       155.1.46.0 [110/74] via 155.1.0.4, 00:03:09, Serial0/0

R1#ping 155.1.46.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.46.6, timeout is 2 seconds:
..!!!
Success rate is 60 percent (3/5), round-trip min/avg/max = 56/58/60 ms

R1#show rtr configuration 1
SA Agent, Infrastructure Engine-II.
Entry number: 1
Owner:
Tag:
Type of operation to perform: udpEcho
Target address: 155.1.46.6
Source address: 0.0.0.0
Target port: 16384
Source port: 16384
Request size (ARR data portion): 16
Operation timeout (milliseconds): 200
```

```
Type Of Service parameters: 0x0
Verify data: No
Data pattern:
Vrf Name:
Control Packets: enabled
Operation frequency (seconds): 1
Next Scheduled Start Time: Start Time already passed
Life (seconds): Forever
Entry Ageout (seconds): never
Status of entry (SNMP RowStatus): Active
Connection loss reaction enabled: No
Timeout reaction enabled: No
Verify error enabled: No
Threshold reaction type: Never
Threshold (milliseconds): 5000
Threshold Falling (milliseconds): 3000
Threshold Count: 5
Threshold Count2: 5
Reaction Type: None
Number of statistic hours kept: 2
Number of statistic distribution buckets kept: 10
Statistic distribution interval (milliseconds): 10
Enhanced History:
Number of history Lives kept: 0
Number of history Buckets kept: 15
History Filter Type: None
```

Legacy FRTS

Objective: Configure routers to conform to provisioned FR link rates



Directions

- Configure routers as per the QoS scenario “Common Configuration”
- Consider that link access-rate (AIR) on both ends is 64Kbps, and provisioned committed-rate (CIR) is 56Kbps
- Configure both routers to shape traffic on PVCs 405 and 504, using Tc value of 10ms, to allow for minimum delay
- Please note, that sending less than 1000 bits per interval does not make real sence, since shaper’s queue is emptied on per-packet basis, and statistically desired CIR could not be achieved with such small Bc and Be
- However, just for reference, the small values of Bc and Be are acceptable
- Allow for extended bursting in case if routers have accumulated enough spare credits, up to link Access Rate of 64Kbps
- Calculate Bc and Be value, using the Tc and CIR/AIR values
  - o Bc = CIR*Tc = 56*0.01 = 560 bits (70 bytes)
  - o Be = (AIR-CIR)*Tc = (64000-56000)*0.01 = 80 bits (10 bytes)
- Apply configuration using the map-class command and legacy FRTS syntax
- Create map-class SHAPE and configure calculated value within
- Enable frame-relay traffic shaping on R4 and R5 FR interfaces, and apply map-class SHAPE to PVCs 405 and 504

Final Configuration

R4:

```
map-class frame-relay SHAPE
 frame-relay cir 56000
 frame-relay bc 560
 frame-relay be 80
!
interface Serial 0/0
 frame-relay traffic-shaping
 frame-relay interface-dlci 405
   class SHAPE
```

```
R5:
map-class frame-relay SHAPE
 frame-relay cir 56000
 frame-relay bc 560
 frame-relay be 80
!
interface Serial 0/0
 frame-relay traffic-shaping
 frame-relay interface-dlci 504
   class SHAPE
```

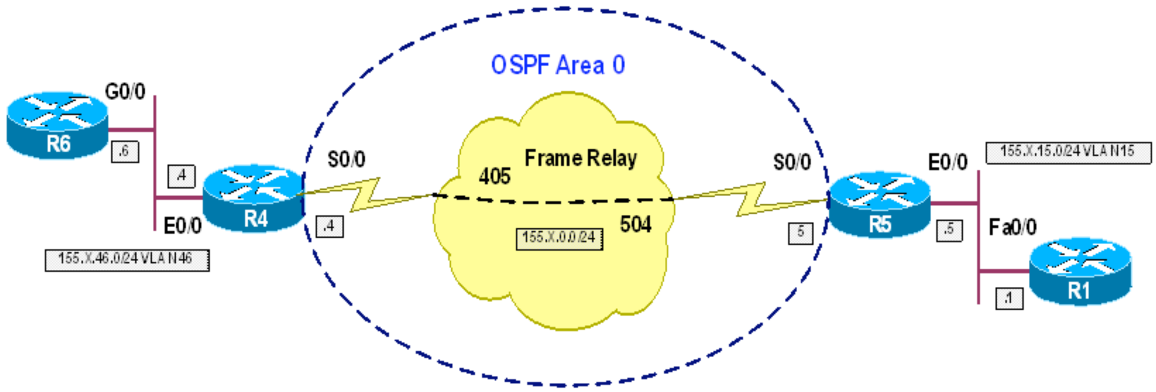
Verification

```
R6#ping 155.1.15.1 repeat 100 size 100

Type escape sequence to abort.
Sending 100, 100-byte ICMP Echos to 155.1.15.1, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 56/58/64 ms

R4#show traffic-shape

Interface   Se0/0
       Access Target    Byte   Sustain   Excess    Interval  Increment Adapt
VC     List   Rate      Limit  bits/int  bits/int  (ms)      (bytes)   Active
401           56000     875    7000      0         125       875       -
402           56000     875    7000      0         125       875       -
403           56000     875    7000      0         125       875       -
413           56000     875    7000      0         125       875       -
405           56000     80     560       80        10        70        -

R4#show frame-relay pvc 405

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 405, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 5763          output pkts 5735         in bytes 968745
  out bytes 554569         dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 305       out bcast bytes 25600
  5 minute input rate 1000 bits/sec, 2 packets/sec
  5 minute output rate 0 bits/sec, 2 packets/sec
  pvc create time 00:50:04, last time pvc status changed 00:49:32
  cir 56000     bc 560      be 80        byte limit 80     interval 10
  mincir 28000     byte increment 70    Adaptive Shaping none
  pkts 323       bytes 20652     pkts delayed 1         bytes delayed 48
  shaping inactive
  traffic shaping drops 0
  Queueing strategy: fifo
  Output queue 0/40, 0 drop, 1 dequeued

R6#ping 155.1.15.1 repeat 100 size 200

Type escape sequence to abort.
Sending 100, 200-byte ICMP Echos to 155.1.15.1, timeout is 2 seconds:
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 108/109/120 ms
R6#

R4#show frame-relay pvc 405

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 405, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 6088          output pkts 6061         in bytes 1004193
  out bytes 585313         dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 315       out bcast bytes 26440
  5 minute input rate 4000 bits/sec, 5 packets/sec
  5 minute output rate 3000 bits/sec, 5 packets/sec
  pvc create time 00:51:52, last time pvc status changed 00:51:20
  cir 56000     bc 560       be 80        byte limit 80     interval 10
  mincir 28000     byte increment 70    Adaptive Shaping none
  pkts 647       bytes 51308     pkts delayed 18        bytes delayed 980
  shaping inactive
  traffic shaping drops 0
  Queueing strategy: fifo
  Output queue 0/40, 0 drop, 18 dequeued
```

Legacy FRTS with Per-VC Priority Queueing

Objective: Configure router for priority-queueing on per-VC basis



Directions

- Configure routers as per the QoS scenario “Configuring Legacy FRTS”
- Create priority group 1 on R4 and R5. Configure this group to assign UDP port 16384 packets to High queue
- Assign this priority-group to map-class SHAPE

Final Configuration

R4 & R5:

```
priority-list 1 protocol ip high udp 16384
!
map-class frame-relay SHAPE
 frame-relay priority-group 1
```

Verification

After PQ has been configured:

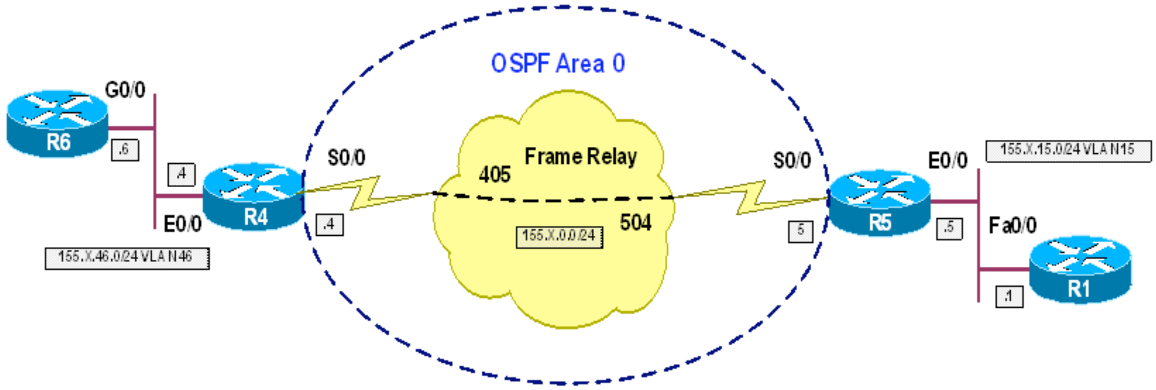
```
R4#show frame-relay pvc 405

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 405, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 11195         output pkts 11140         in bytes 2347693
  out bytes 871214         dropped pkts 0            in pkts dropped 0
  out pkts dropped 0                out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0            out FECN pkts 0
  out BECN pkts 0          in DE pkts 0              out DE pkts 0
  out bcast pkts 492       out bcast bytes 41308
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 01:20:49, last time pvc status changed 01:20:17
  cir 56000     bc 560       be 80        byte limit 80     interval 10
  mincir 28000     byte increment 70    Adaptive Shaping none
  pkts 5726      bytes 337209    pkts delayed 119       bytes delayed 27178
```



```
  shaping inactive
  traffic shaping drops 0
  Queueing strategy: priority-list 1

List   Queue  Args
1      high   protocol ip          udp port 16384
  Output queue: high 0/20/0, medium 0/40/0, normal 0/60/0, low 0/80/0

R5(config)#tftp-server flash:c3640-jk9o3s-mz.123-14.T7.bin alias bulk

R4#copy tftp: null:
Address or name of remote host []? 155.1.0.5
Source filename []? bulk
Accessing tftp://155.1.0.5/bulk...
Loading bulk from 155.1.0.5 (via Serial0/0): !!!!!!!!!!!!!!!!!!!!

R5#debug priority
Priority output queueing debugging is on
R5#
PQ: Serial0/0 dlci 504 : ip (defaulting) -> normal
PQ: Serial0/0 dlci 504 : ip (udp 16384) -> high
PQ: Serial0/0 dlci 504 : ip (udp 16384) -> high
PQ: Serial0/0 dlci 504 : ip (udp 16384) -> high
PQ: Serial0/0 dlci 504 : ip (udp 16384) -> high
PQ: Serial0/0 dlci 504 : ip (defaulting) -> normal
PQ: Serial0/0 dlci 504 : ip (udp 16384) -> high
PQ: Serial0/0 dlci 504 : ip (udp 16384) -> high
PQ: Serial0/0 dlci 504 : ip (udp 16384) -> high
PQ: Serial0/0 dlci 504 : ip (udp 16384) -> high

To stop file transfer, apply access-list to R5 FR interface:

R5(config)#access-list 100 deny ip any any
R5(config)#int se 0/0
R5(config-if)#ip access-group 100 in
```

Frame-Relay Adaptive Shaping

Objective: Configure router to throttle PVC sending rate in response to interface congestion



Directions

- Configure routers as per the QoS scenario "Configuring Legacy FRTS"
- This time, set CIR=AIR=64Kbps, and set minCIR=CIR=56Kbps
- Calculate new Bc and Be values. Leave Tc value the same (Tc=10ms)
  Since CIR=AIR we can not burst about CIR, hence Be=0
  Bc=CIR*Tc = AIR*Tc=64000*0.01=640
- To adapt to network congestions, configure map class SHAPE to respond to interface congestion (when interface queue starts filling up) as soon as queue depth is 1

Final Configuration

```
R4 & R5:
map-class frame-relay SHAPE
 frame-relay cir 64000
 frame-relay mincir 56000
 frame-relay bc 640
 frame-relay be 0
 frame-relay adaptive-shaping interface-congestion 1
```

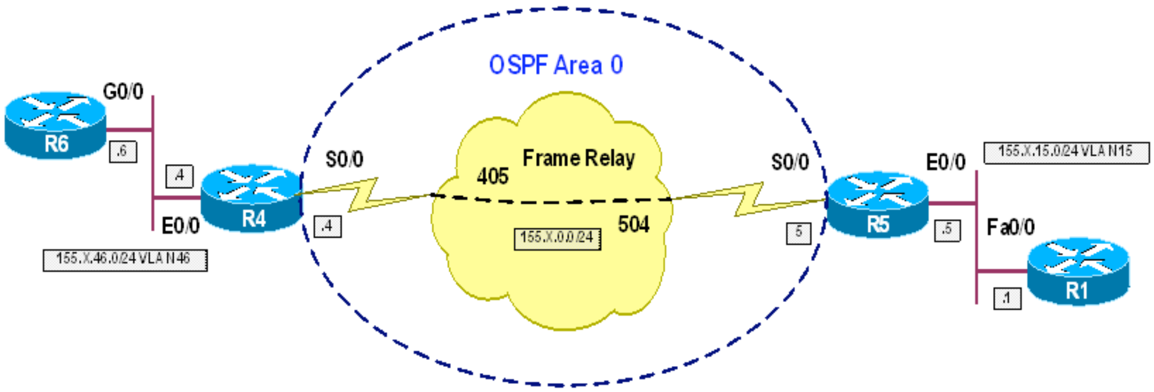
Verification

```
R5#show frame-relay pvc 504

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 504, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 3926          output pkts 3909         in bytes 158522
  out bytes 258232         dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
```

```
  out bcast pkts 205       out bcast bytes 17208
  5 minute input rate 0 bits/sec, 2 packets/sec
  5 minute output rate 0 bits/sec, 2 packets/sec
  pvc create time 00:54:50, last time pvc status changed 00:54:50
  cir 64000     bc 640       be 0         byte limit 80     interval 10
  mincir 56000     byte increment 80    Adaptive Shaping IF_CONG
  pkts 5763      bytes 258232    pkts delayed 3708      bytes delayed 174296
  shaping inactive
  traffic shaping drops 0
  Queueing strategy: fifo
  Output queue 0/40, 0 drop, 0 dequeued

R1#ping 155.1.46.6 size 500 repeat 10000 timeout 0

Type escape sequence to abort.
Sending 10000, 500-byte ICMP Echos to 155.1.46.6, timeout is 0 seconds:
......................................................................
......................................................................
......................................................................
......................................................................
......................................................................
......................................................................
......................................................................
..................................

R5#show frame-relay pvc 504

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 504, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 5406          output pkts 5460         in bytes 307639
  out bytes 850852         dropped pkts 8099        in pkts dropped 0
  out pkts dropped 8099             out bytes dropped 4080612
  late-dropped out pkts 8099        late-dropped out bytes 4080612
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 227       out bcast bytes 19056
  5 minute input rate 9000 bits/sec, 10 packets/sec
  5 minute output rate 29000 bits/sec, 11 packets/sec
  pvc create time 00:58:30, last time pvc status changed 00:58:30
  cir 64000     bc 640       be 0         byte limit 80     interval 10
  mincir 56000     byte increment 80    Adaptive Shaping IF_CONG
  pkts 7269      bytes 828128    pkts delayed 4191      bytes delayed 323040
  shaping active
  traffic shaping drops 0
  Queueing strategy: fifo
  Output queue 27/40, 8363 drop, 501 dequeued
```

Frame-Relay Fragmentation (FRF.12)

Objective: Configure routers to fragment and interleave large packets



Directions

- Configure routers as per the QoS scenario "Legacy FRTS"
- Configure the fragment size to accommodate for 10ms serialization delay
  Since serialization is performed at AIR speed, fragment-size should be
  Frag=64000*0.01/8=80 bytes
- Configure this fragment size under map-class on R4 and R5

Final Configuration

```
R4 & R5:
map-class frame-relay SHAPE
 frame-relay fragment 80
```

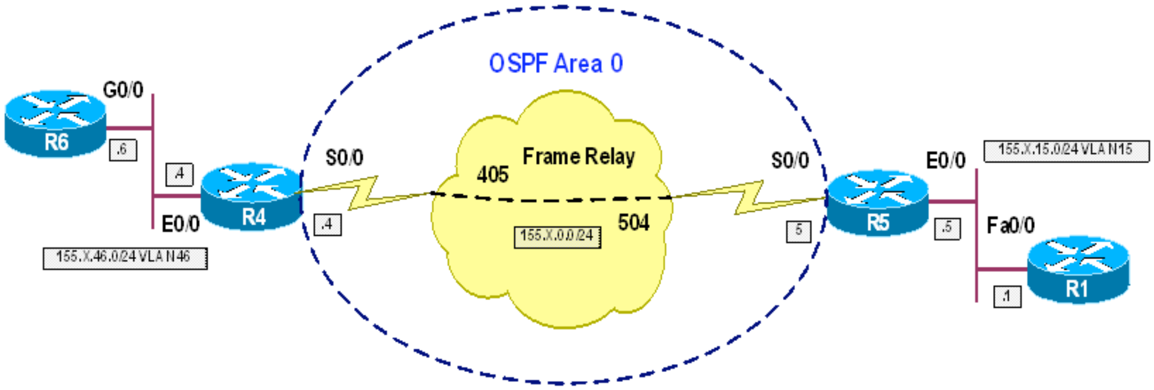
Verification

```
R5#show frame-relay fragment
interface                 dlci frag-type  size in-frag    out-frag    dropped-
frag
Se0/0                     504  end-to-end 80   48         54          0


R5#show frame-relay pvc 504

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 504, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 1541          output pkts 1539         in bytes 71376
  out bytes 103526         dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 83        out bcast bytes 6972
  5 minute input rate 0 bits/sec, 2 packets/sec
  5 minute output rate 1000 bits/sec, 2 packets/sec
  pvc create time 00:13:13, last time pvc status changed 00:13:13
```

```
  Queueing strategy: weighted fair
  Current fair queue configuration:
   Discard     Dynamic      Reserved
   threshold   queue count  queue count
    64          16           0
  Output queue size 0/max total 600/drops 0
  fragment type end-to-end fragment size 80
  cir 56000     bc   560       be 80        limit 80     interval 10
  mincir 28000     byte increment 70    BECN response no  IF_CONG no
  frags 801       bytes 51516     frags delayed 72        bytes delayed 3384
  shaping inactive
  traffic shaping drops 0

R5#show queueing interface serial 0/0
Interface Serial0/0 queueing strategy: priority

Output queue utilization (queue/count)
        high/26 medium/0 normal/568 low/0
```

Frame-Relay IP RTP Priority

Objective: Configure routers to give voice traffic priority treatment on per-VC basis



Directions

- Configure routers as per the QoS scenario “Frame-Relay Fragmentation (FRF.12)”
- Note that IP RTP Priority has no effect until FRF.12 is turned on
- Enable Frame-Relay IP RTP Priority under map-class SHAPE on R4 and R5
- Specify RTP port starting at 16364 and lengthening for 16383 more ports
- Permit voice traffic to use up to all PVC bandwidth (CIR=56Kbps)

Final Configuration

```
R4 & R5:
map-class frame-relay SHAPE
 frame-relay ip rtp priority 16384 16383 56
```

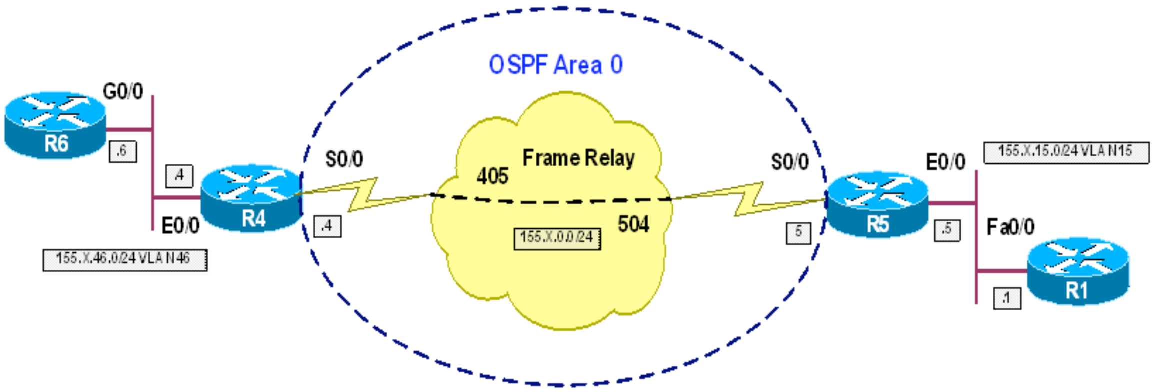
Verification

```
R4#show frame-relay pvc 405

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 405, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 4539          output pkts 4473          in bytes 477336
  out bytes 405534         dropped pkts 0            in pkts dropped 0
  out pkts dropped 0                out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0            out FECN pkts 0
  out BECN pkts 0          in DE pkts 0              out DE pkts 0
  out bcast pkts 205       out bcast bytes 17200
  5 minute input rate 1000 bits/sec, 2 packets/sec
  5 minute output rate 0 bits/sec, 2 packets/sec
  pvc create time 00:33:31, last time pvc status changed 00:33:31
  Queueing strategy: weighted fair
  Current fair queue configuration:
```

```
   Discard     Dynamic      Reserved
   threshold   queue count  queue count
    64          16           0
  Output queue size 0/max total 600/drops 0
  fragment type end-to-end fragment size 80
  cir 56000     bc   560       be 80        limit 80     interval 10
  mincir 28000     byte increment 70    BECN response no  IF_CONG no
  frags 6263      bytes 370180    frags delayed 3028      bytes delayed 236084
  shaping inactive
  traffic shaping drops 0
  ip rtp priority parameters 16384 32767 56000

R5#show frame-relay pvc 504

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 504, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 1847          output pkts 1846         in bytes 74372
  out bytes 122620         dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 90        out bcast bytes 7580
  5 minute input rate 0 bits/sec, 2 packets/sec
  5 minute output rate 1000 bits/sec, 2 packets/sec
  pvc create time 00:36:05, last time pvc status changed 00:36:05
  Queueing strategy: weighted fair
  Current fair queue configuration:
   Discard     Dynamic      Reserved
   threshold   queue count  queue count
    64          16           0
  Output queue size 0/max total 600/drops 0
  fragment type end-to-end fragment size 80
  cir 56000     bc   560       be 80        limit 80     interval 10
  mincir 28000     byte increment 70    BECN response no  IF_CONG no
  frags 2817      bytes 122750    frags delayed 1938      bytes delayed 91106
  shaping inactive
  traffic shaping drops 0
  ip rtp priority parameters 16384 32767 56000

R5#debug priority
Priority output queueing debugging is on
*Nov 17 21:32:39.251: PQ: Serial0/0 output (Pk size/Q 13/0)
*Nov 17 21:32:39.931: PQ: Serial0/0 output (Pk size/Q 86/2)
*Nov 17 21:32:39.939: PQ: Serial0/0 output (Pk size/Q 8/2)
*Nov 17 21:32:40.931: PQ: Serial0/0 output (Pk size/Q 86/2)
*Nov 17 21:32:40.939: PQ: Serial0/0 output (Pk size/Q 8/2)
*Nov 17 21:32:41.907: PQ: Serial0/0 output (Pk size/Q 86/2)
```

Frame-Relay Per-VC CBWFQ

Objective: Configure the router to use CBWFQ as Per-VC queueing strategy



Directions

- Configure routers as per the QoS scenario “Frame-Relay Adaptive Shaping”
- Create class-map VOICE on R4 and R5, and match “ip RTP” with it. Select ports starting at 16384 and ranging for 16383 more ports This class will distinguish voice traffic
- Create policy map PER_VC_POLICY on R4 and R5.
  - o Configure class VOICE within this policy map, and give it priority treatment of up to 32Kbps. Set burst size to 4000 bytes (1 second of bit-rate)
  - o Configure class-default to use fair-queue.
- Apply policy-map PER_VC_POLICY as service-policy for map-class SHAPE
- Note that bandwidth available to CBWFQ is taken from minCIR value, and not CIR

Final Configuration

R4 & R5:

```
class-map VOICE
 match ip rtp 16384 16383
!
policy-map PER_VC_POLICY
 class VOICE
  priority 32 4000
 class class-default
  fair-queue
!
map-class frame-relay SHAPE
 service-policy output PER_VC_POLICY
```

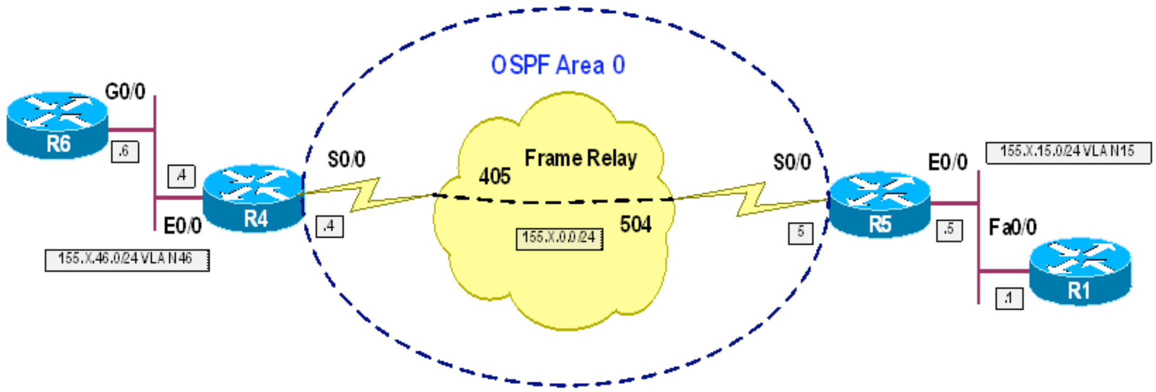
Verification

```
R5#show frame-relay pvc 504

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 504, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 210           output pkts 209          in bytes 9136
  out bytes 13152          dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 19        out bcast bytes 1596
  5 minute input rate 0 bits/sec, 2 packets/sec
  5 minute output rate 0 bits/sec, 2 packets/sec
  pvc create time 00:02:42, last time pvc status changed 00:02:42
  cir 64000     bc 640       be 0         byte limit 80     interval 10
  mincir 56000     byte increment 80    Adaptive Shaping IF_CONG
  pkts 142       bytes 8664      pkts delayed 0         bytes delayed 0
  shaping inactive
  traffic shaping drops 0
  service policy PER_VC_POLICY
 Serial0/0: DLCI 504 -

  Service-policy output: PER_VC_POLICY

    Class-map: VOICE (match-all)
      7 packets, 252 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip rtp 16384 16383
      Queueing
        Strict Priority
        Output Queue: Conversation 24
        Bandwidth 32 (kbps) Burst 4000 (Bytes)
        (pkts matched/bytes matched) 0/0
        (total drops/bytes drops) 0/0

    Class-map: class-default (match-any)
      8 packets, 672 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
      Queueing
```

```
        Flow Based Fair Queueing
        Maximum Number of Hashed Queues 16
        (total queued/total drops/no-buffer drops) 0/0/0
  Output queue size 0/max total 600/drops 0

R5#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R5(config)#policy-map PER_VC_POLICY
R5(config-pmap)#class VOICE
R5(config-pmap-c)#priority 56 7000
R5(config-pmap-c)#priority 64 8000
I/f Serial0/0 DLCI 504 Class VOICE requested bandwidth 64 (kbps) Only 56 (kbps)
available
```

MQC-Only FRTS Configuration

Objective: Configure the router to shape FR traffic using MQC only



Directions

- Configure routers as per the QoS scenario “Common Configuration”
- The goal is to provide voice traffic with priority treatment, and let the other traffic use fair-queue scheduling
- Configure bandwidth 64K on R4 and R5 FR interfaces, and set the maximum-reserved-bandwidth to 100%
- Consider interfaces access rate AIR=64Kpbs, and provisioned CIR=56Kbps
- Create class-map FR_PVC on R4 and R5, and match DLCI 405 on R4 and DLCI 504 on R4 within this class. This class encompasses all the traffic flowing on respective VC
- Create class-map VOICE on R4 and R5, and match “ip RTP” with it. Select ports starting at 16384 and ranging for 16383 more ports. This class will distinguish voice traffic
- Create policy map PER_VC_POLICY on R4 and R5
  - o Configure class VOICE within this policy map, and give it priority treatment of up to 32Kbps. Set burst size to 4000 bytes (1 second of bit-rate)
  - o Configure class-default to use fair-queue
- Create policy-map PER_INTERFACE_POLICY on R4 and R5
  - o Configure class FR_PVC within this policy map, and shape it up to 56Kpbs.
  - o Use Tc value of 125ms to yield Bc=7000 bits
  - o Additionally, permit excessive bursting of up to AIR rate, i.e. Be=(AIR-CIR)*Tc=8000*0.125=1000 bits
- Assign PER_VC_POLICY as nested policy map for class FR_PVC

Final Configuration

R4:
```
class-map FR_PVC
 match fr-dlci 405
!
class-map VOICE
 match ip rtp 16384 16383
!
policy-map PER_VC_POLICY
 class VOICE
  priority 32 4000
 class class-default
  fair-queue
!
policy-map PER_INTERFACE_POLICY
 class FR_PVC
  shape average 56000 7000 1000
  service-policy PER_VC_POLICY
!
interface Serial 0/0
 bandwidth 64
 max-reserved 100
 service-policy output PER_INTERFACE_POLICY
```

R5:
```
class-map FR_PVC
 match fr-dlci 504
!
class-map VOICE
 match ip rtp 16384 16383
!
policy-map PER_VC_POLICY
 class VOICE
  priority 32 4000
 class class-default
  fair-queue
!
policy-map PER_INTERFACE_POLICY
 class FR_PVC
  shape average 56000 7000 1000
  service-policy PER_VC_POLICY
!
interface Serial 0/0
 bandwidth 64
 max-reserved 100
 service-policy output PER_INTERFACE_POLICY
```

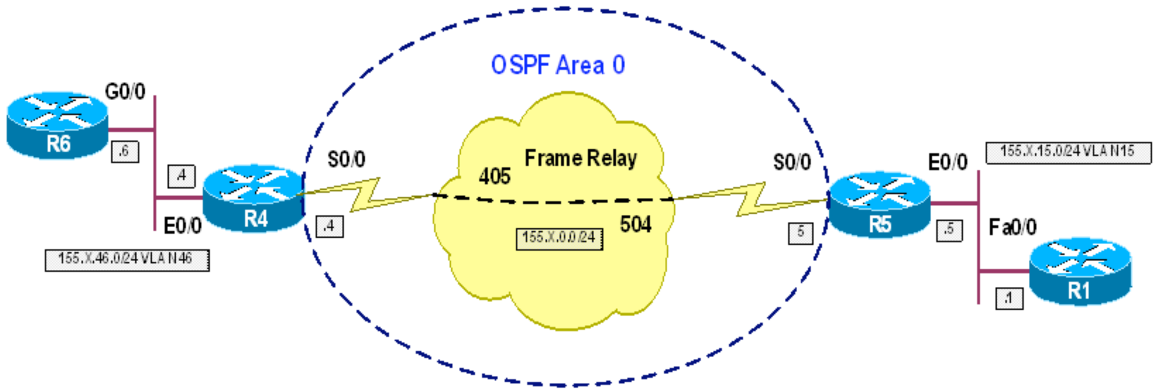
Verification

```
R5#show policy-map interface serial 0/0

 Serial0/0

  Service-policy output: PER_INTERFACE_POLICY

    Class-map: FR_PVC (match-all)
      621 packets, 37988 bytes
```

```
  5 minute offered rate 1000 bps, drop rate 0 bps
  Match: fr-dlci 504
  Traffic Shaping
       Target/Average   Byte   Sustain   Excess    Interval  Increment
         Rate           Limit  bits/int  bits/int  (ms)      (bytes)
        56000/56000     1000   7000      1000      125       875

    Adapt  Queue     Packets   Bytes     Packets   Bytes     Shaping
    Active Depth                         Delayed   Delayed   Active
    -      0         621       37988     0         0         no

  Service-policy : PER_VC_POLICY

    Class-map: VOICE (match-all)
      59 packets, 2124 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip rtp 16384 16383
      Queueing
        Strict Priority
        Output Queue: Conversation 24
        Bandwidth 32 (kbps) Burst 4000 (Bytes)
        (pkts matched/bytes matched) 0/0
        (total drops/bytes drops) 0/0

    Class-map: class-default (match-any)
      562 packets, 35864 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
      Queueing
        Flow Based Fair Queueing
        Maximum Number of Hashed Queues 16
    (total queued/total drops/no-buffer drops) 0/0/0

Class-map: class-default (match-any)
  30 packets, 390 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
```

MQC FRTS

Objective: Configure the router for FRTS using per-VC MQC configuration



Directions

- Configure routers as per the QoS scenario "Common Configuration"
- Consider interfaces access rate AIR=64Kpbs, and provisioned CIR=56Kbps
- Create class-map VOICE on R4 and R5, and match "ip RTP" with it. Select ports starting at 16384 and ranging for 16383 more ports This class will distinguish voice traffic
- Create policy map CBWFQ on R4 and R5.
  - o Configure class VOICE within this policy map, and give it priority treatment of up to 32Kbps. Set burst size to 4000 bytes (1 second of bit-rate)
  - o Configure class-default to use fair-queue.
- Create policy-map PER_VC_POLICY on R4 and R5
  - o Configure class class-default within this policy map, and shape it up to 56Kpbs
  - o Use Tc value of 125ms to yield Bc=7000 bits
  - o Additionally, permit excessive bursting of up to AIR rate, i.e. Be=(AIR-CIR)*Tc=8000*0.125=1000 bits
  - o Assign CBWFQ as nested policy map for class PER_VC_POLICY
- Create Frame-Relay map-class SHAPE and assign PER_VC_POLICY as service-policy for this map-class
- Assign map-class SHAPE to PVCs 405 and 504 on R4 and R5

Final Configuration

R4:

```
class-map VOICE
 match ip rtp 16384 16383
!
policy-map CBWFQ
 class VOICE
  priority 32 4000
```

```
 class class-default
  fair-queue
!
policy-map PER_VC_POLICY
  class class-default
    shape average 56000 7000 1000
    service-policy CBWFQ
!
map-class frame-relay SHAPE
 service-policy output PER_VC_POLICY
!
interface Serial 0/0
 frame-relay interface-dlci 405
   class SHAPE
```

R5:

```
class-map VOICE
 match ip rtp 16384 16383
!
policy-map CBWFQ
 class VOICE
  priority 32 4000
 class class-default
  fair-queue
!
policy-map PER_VC_POLICY
  class class-default
    shape average 56000 7000 1000
    service-policy CBWFQ
!
map-class frame-relay SHAPE
 service-policy output PER_VC_POLICY
!
interface Serial 0/0
 frame-relay interface-dlci 504
   class SHAPE
```

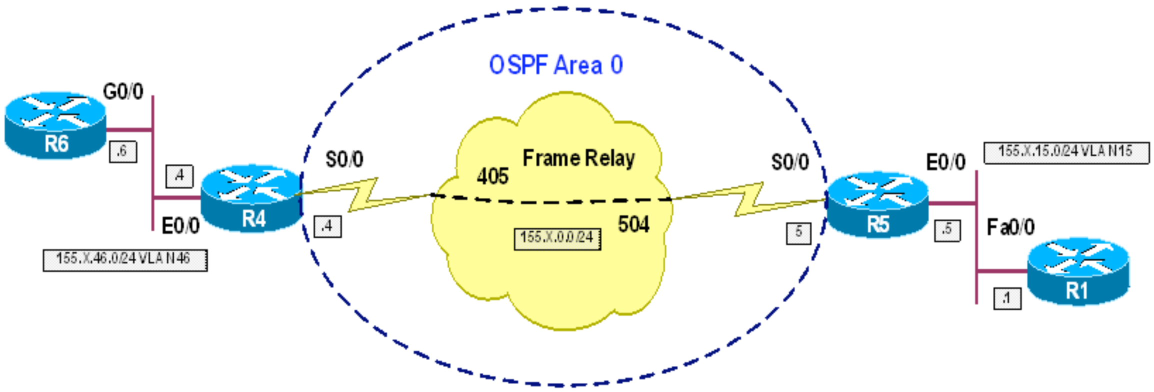
Verification

```
R4#show policy-map interface serial 0/0
 Serial0/0: DLCI 405 -

  Service-policy output: PER_VC_POLICY

    Class-map: class-default (match-any)
      61 packets, 2456 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
      Traffic Shaping
           Target/Average   Byte   Sustain   Excess    Interval  Increment
             Rate           Limit  bits/int  bits/int  (ms)      (bytes)
            56000/56000     1000   7000      1000      125       875

        Adapt  Queue     Packets   Bytes     Packets   Bytes     Shaping
        Active Depth                         Delayed   Delayed   Active
        -      0         61        2456      0         0         no

      Service-policy : CBWFQ

        Class-map: VOICE (match-all)
```

```
          58 packets, 2204 bytes
          5 minute offered rate 0 bps, drop rate 0 bps
          Match: ip rtp 16384 16383
          Queueing
            Strict Priority
            Output Queue: Conversation 24
            Bandwidth 32 (kbps) Burst 4000 (Bytes)
            (pkts matched/bytes matched) 0/0
            (total drops/bytes drops) 0/0

        Class-map: class-default (match-any)
          3 packets, 252 bytes
          5 minute offered rate 0 bps, drop rate 0 bps
          Match: any
          Queueing
            Flow Based Fair Queueing
            Maximum Number of Hashed Queues 16
        (total queued/total drops/no-buffer drops) 0/0/0

R5#show policy-map interface serial 0/0
 Serial0/0: DLCI 504 -

  Service-policy output: PER_VC_POLICY

    Class-map: class-default (match-any)
      15 packets, 924 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: any
      Traffic Shaping
           Target/Average   Byte   Sustain   Excess    Interval  Increment
             Rate           Limit  bits/int  bits/int  (ms)      (bytes)
            56000/56000     1000   7000      1000      125       875

        Adapt  Queue     Packets   Bytes     Packets   Bytes     Shaping
        Active Depth                         Delayed   Delayed   Active
        -      0         15        924       0         0         no

      Service-policy : CBWFQ

        Class-map: VOICE (match-all)
          7 packets, 252 bytes
          5 minute offered rate 0 bps, drop rate 0 bps
          Match: ip rtp 16384 16383
          Queueing
            Strict Priority
            Output Queue: Conversation 24
            Bandwidth 32 (kbps) Burst 4000 (Bytes)
            (pkts matched/bytes matched) 0/0
            (total drops/bytes drops) 0/0

        Class-map: class-default (match-any)
          8 packets, 672 bytes
          5 minute offered rate 0 bps, drop rate 0 bps
          Match: any
          Queueing
            Flow Based Fair Queueing
            Maximum Number of Hashed Queues 16
        (total queued/total drops/no-buffer drops) 0/0/0
```

Voice-Adaptive FRTS

Objective: Configure the router for adaptive shaping based on voice presence



Directions

- Configure routers as per the QoS scenario "MQC FRTS"
- Configure policy-map PER_VC_POLICY to shape class-default down to 32Kbps adapting to congestion
- Configure this shaping to react to presence of traffic in priority queue.

Final Configuration

R4 & R5:
```
policy-map PER_VC_POLICY
 class class-default
   shape adaptive 32000
   shape fr-voice-adapt
```

Verification

```
R5#show policy-map interface serial 0/0
 Serial0/0: DLCI 504 -

  Service-policy output: PER_VC_POLICY

    Class-map: class-default (match-any)
      656 packets, 41360 bytes
      5 minute offered rate 1000 bps, drop rate 0 bps
      Match: any
      Traffic Shaping
           Target/Average   Byte   Sustain   Excess    Interval  Increment
             Rate           Limit  bits/int  bits/int  (ms)      (bytes)
            56000/56000     1000   7000      1000      125       875

        Adapt  Queue     Packets   Bytes     Packets   Bytes     Shaping
        Active Depth                         Delayed   Delayed   Active
        BECN   0         644       40352     22        1800      no
        Voice Adaptive Shaping active, time left 29 secs

      Service-policy : CBWFQ
```

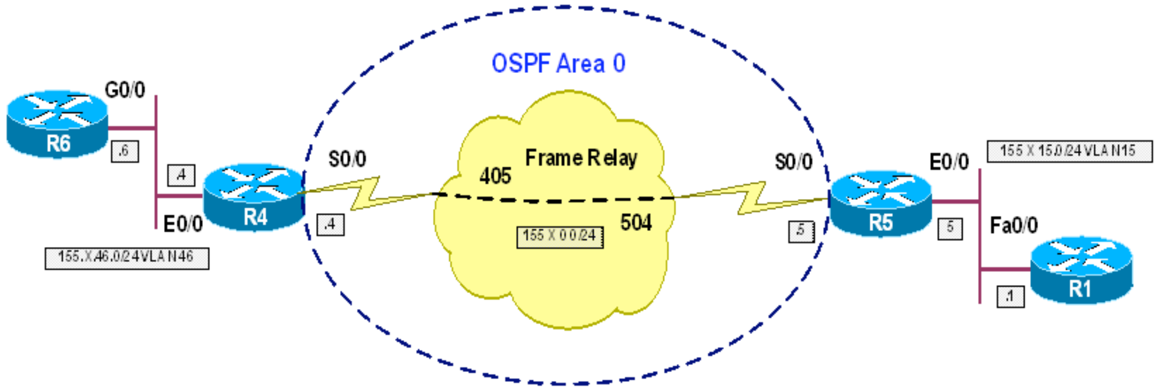


```
        Class-map: VOICE (match-all)
          287 packets, 10332 bytes
          5 minute offered rate 0 bps, drop rate 0 bps
          Match: ip rtp 16384 16383
          Queueing
            Strict Priority
            Output Queue: Conversation 24
            Bandwidth 32 (kbps) Burst 4000 (Bytes)
            (pkts matched/bytes matched) 188/6768
            (total drops/bytes drops) 0/0

        Class-map: class-default (match-any)
          369 packets, 31028 bytes
          5 minute offered rate 0 bps, drop rate 0 bps
          Match: any
          Queueing
            Flow Based Fair Queueing
            Maximum Number of Hashed Queues 16
        (total queued/total drops/no-buffer drops) 0/0/0

R1#ping 155.1.46.6 size 800 repeat 20000 timeout 0

Type escape sequence to abort.
Sending 20000, 800-byte ICMP Echos to 155.1.46.6, timeout is 0 seconds:
......................................................................
......................................................................
......................................................................
......................................................................
......................................................................
......................................................................
......................................................................
......................................................................
......................................................................
......................................................................
......................................................................
.................................

R5#show policy-map interface serial 0/0
 Serial0/0: DLCI 504 -

  Service-policy output: PER_VC_POLICY

    Class-map: class-default (match-any)
      4949 packets, 3417908 bytes
      5 minute offered rate 1000 bps, drop rate 0 bps
      Match: any
      Traffic Shaping
           Target/Average   Byte   Sustain   Excess    Interval  Increment
             Rate           Limit  bits/int  bits/int  (ms)      (bytes)
            56000/56000     1000   7000      1000      125       875

        Adapt  Queue     Packets   Bytes     Packets   Bytes     Shaping
        Active Depth                         Delayed   Delayed   Active
        BECN   64        764       64784     53        20004     yes
        Voice Adaptive Shaping active, time left 29 secs

      Service-policy : CBWFQ

        Class-map: VOICE (match-all)
          333 packets, 11988 bytes
          5 minute offered rate 0 bps, drop rate 0 bps
```

```
        Match: ip rtp 16384 16383
        Queueing
          Strict Priority
          Output Queue: Conversation 24
          Bandwidth 32 (kbps) Burst 4000 (Bytes)
          (pkts matched/bytes matched) 235/8460
          (total drops/bytes drops) 0/0

      Class-map: class-default (match-any)
        4616 packets, 3405920 bytes
        5 minute offered rate 0 bps, drop rate 0 bps
        Match: any
        Queueing
          Flow Based Fair Queueing
          Maximum Number of Hashed Queues 16
      (total queued/total drops/no-buffer drops) 64/5927/0
```

Frame-Relay Voice-Adaptive Fragmentation

Objective: Configure the router for fragment large packets only in presence of voice traffic



Directions

- Configure routers as per the QoS scenario “Voice adaptive FRTS”
- Configure Frame-Relay map-class SHAPE to fragment packets
- Configure fragment size to accommodate for 10ms serialization delay
- Since AIR=64Kbps, Fragment Size = 64000*10ms/8=80 bytes
- Configure FR interface to fragment packets only if voice is present in priority queue

Final Configuration

```
R4 & R5:
map-class frame-relay SHAPE
 frame-relay fragment 80
!
interface Serial 0/0
 frame-relay fragmentation voice-adaptive
```

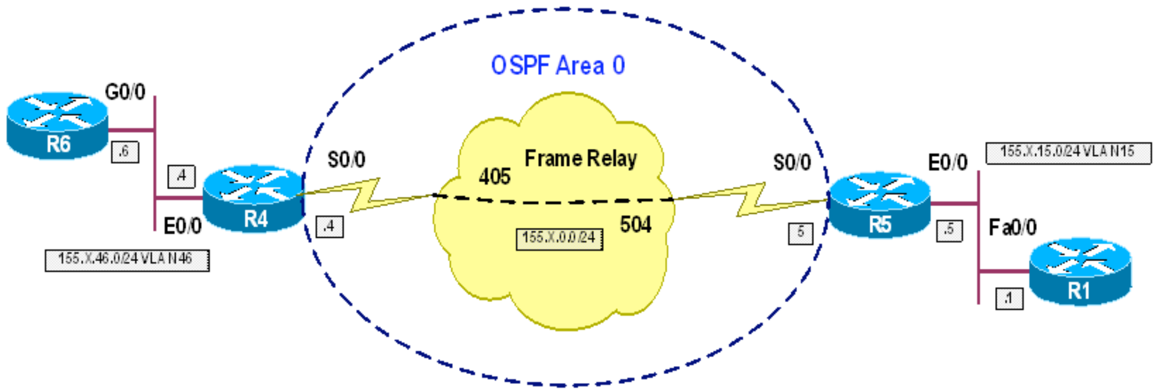
Verification

```
R5#show frame-relay fragment
interface                 dlci frag-type  size in-frag     out-frag    dropped-
frag
Se0/0                     504  end-to-end 80   22          254         0

R5#show policy-map interface serial 0/0
 Serial0/0: DLCI 504 -

  Service-policy output: PER_VC_POLICY

    Class-map: class-default (match-any)
      64 packets, 4192 bytes
      5 minute offered rate 1000 bps, drop rate 0 bps
      Match: any
      Traffic Shaping
```

```
    Target/Average   Byte   Sustain   Excess    Interval  Increment
      Rate           Limit  bits/int  bits/int  (ms)      (bytes)
     56000/56000     1000   7000      1000      125       875

  Adapt  Queue     Packets   Bytes     Packets   Bytes     Shaping
  Active Depth                         Delayed   Delayed   Active
  BECN   0         89        4442      50        2350      no
  Voice Adaptive Shaping active, time left 29 secs

Service-policy : CBWFQ

  Class-map: VOICE (match-all)
    24 packets, 864 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    Match: ip rtp 16384 16383
    Queueing
      Strict Priority
      Output Queue: Conversation 24
      Bandwidth 32 (kbps) Burst 4000 (Bytes)
      (pkts matched/bytes matched) 31/1116
      (total drops/bytes drops) 0/0

  Class-map: class-default (match-any)
    40 packets, 3328 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    Match: any
    Queueing
      Flow Based Fair Queueing
      Maximum Number of Hashed Queues 16
  (total queued/total drops/no-buffer drops) 0/0/0
```

FRF.11 Annex C Fragmentation for VoFR

Objective: Configure the routers to use fragmentation scheme that never fragments voice packets



Directions

- Configure routers as per the QoS scenario "Legacy FRTS"
- The goal is to use standard multiprotocol encapsulation for data, and use FRF.11 encapsulation for voice packets
- Configure map-class SHAPE for fragmentation, using fragment size of 80 bytes, to accommodate for 10ms delay over 64Kpbs link
- Allocate 56Kbps of bandwidth to VoFR traffic, using "**frame-relay voice bandwidth**" command under map-class SHAPE
- Configure "**vofr cisco**" on DLCIs 504 and 405 to use FRF.11 encapsulation for voice packets

Final Configuration

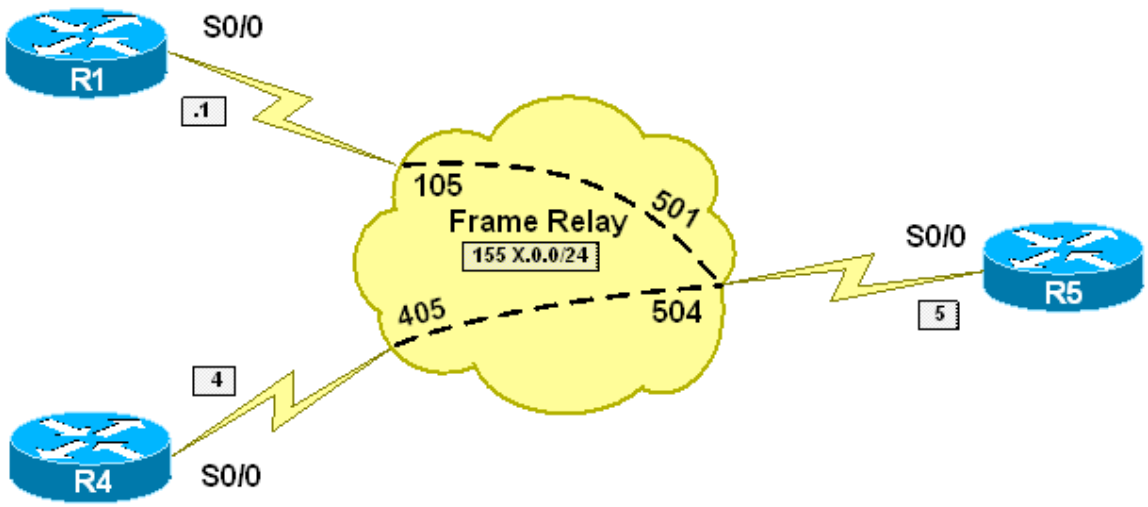
```
R4:
map-class frame-relay SHAPE
  frame-relay fragment 80
  frame-relay voice bandwidth 56000
!
interface Serial 0/0
  frame-relay interface-dlci 405
    vofr cisco

R5:
map-class frame-relay SHAPE
  frame-relay fragment 80
  frame-relay voice bandwidth 56000
!
interface Serial 0/0
  frame-relay interface-dlci 504
    vofr cisco
```

Verification

```
R5#show frame-relay fragment
interface                 dlci frag-type  size in-frag    out-frag   dropped-
frag
Se0/0                     504  VoFR-cisco 80   266        381        0

R5#show frame pvc 504

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 504, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 673           output pkts 404          in bytes 25493
  out bytes 28567          dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 88        out bcast bytes 7184
  5 minute input rate 0 bits/sec, 2 packets/sec
  5 minute output rate 1000 bits/sec, 2 packets/sec
  pvc create time 00:13:56, last time pvc status changed 00:13:34
  Service type VoFR-cisco
  Queueing strategy: weighted fair
   Voice Queueing Stats: 0/0 (size/dropped)
  Current fair queue configuration:
   Discard     Dynamic      Reserved
   threshold   queue count  queue count
    64          16           0
  Output queue size 0/max total 600/drops 0
  configured voice bandwidth 56000, used voice bandwidth 0
  fragment type VoFR-cisco fragment size 80
  cir 56000     bc   560       be 80        limit 80     interval 10
  mincir 28000     byte increment 70    BECN response no  IF_CONG no
  frags 537       bytes 27881     frags delayed 404       bytes delayed 19456
  shaping inactive
  traffic shaping drops 0
```

Frame-Relay PIPQ

Objective: Configure router to map DLCIs to interface priority-groups



Directions

- Configure IP addressing as per the diagram
- Use physical frame-relay interface types, and static IP to DLCI mapping
- PIPQ enables Priority Queue as interface-level queueing mechanism, and permits mapping of DLCIs to different priority groups (high, medium, normal, low)
- Enable PIPQ as R5 FR interface queue, using interface command "**frame-relay interface-queue priority**"
- Create frame-relay map-class DLCI_504 on R5. Assign this class to high priority interface queue (**frame-relay interface-queue priority high**) Apply this map-class to DLCI 504
- Create frame-relay map-class DLCI_501 on R5. Assign this class to low priority interface queue (**frame-relay interface-queue priority high**) Apply this map-class to DLCI 501

Final Configuration

```
R1:
interface Serial 0/0
 encapsulation frame
 no frame inverse-arp
 no shutdown
 ip address 155.1.0.1 255.255.255.0
 frame-relay map ip 155.1.0.5 105

R4:
interface Serial 0/0
 encapsulation frame
 no frame inverse-arp
 no shutdown
```

```
 ip address 155.1.0.4 255.255.255.0
 frame-relay map ip 155.1.0.5 405

R5:
interface Serial 0/0
 encapsulation frame
 no frame inverse-arp
 no shutdown
 ip address 155.1.0.5 255.255.255.0
 frame-relay map ip 155.1.0.4 504
 frame-relay map ip 155.1.0.1 501
 frame-relay interface-queue priority
 frame-relay interface-dlci 504
   class DLCI_504
frame-relay interface-dlci 501
   class DLCI_501
!
map-class frame-relay DLCI_504
 frame-relay interface-queue priority high
!
map-class frame-relay DLCI_501
 frame-relay interface-queue priority low
```

Verification

```
R5#ping 155.1.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/58/60 ms

R5#ping 155.1.0.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 155.1.0.4, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/59/60 ms

R5#show queueing interface serial 0/0
Interface Serial0/0 queueing strategy: priority

Output queue utilization (queue/count)
        high/38 medium/0 normal/0 low/14

R5#show frame-relay pvc 504

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 504, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 5             output pkts 16           in bytes 520
  out bytes 1664           dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 0         out bcast bytes 0
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
```

```
  pvc create time 00:04:27, last time pvc status changed 00:03:28
  priority high

R5#show frame-relay pvc 501

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

DLCI = 501, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0

  input pkts 5             output pkts 14           in bytes 520
  out bytes 1456           dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 0         out bcast bytes 0
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 00:04:31, last time pvc status changed 00:03:42
  priority low
```