Frame-Relay Traffic-Shaping Example

```
interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type ansi
 frame-relay traffic-shaping
 frame-relay class slow_vcs
!
interface Serial0.1 point-to-point
 ip address 10.128.30.1 255.255.255.248
 ip ospf cost 200
 bandwidth 10
 frame-relay interface-dlci 101
!
interface Serial0.2 point-to-point
 ip address 10.128.30.9 255.255.255.248
 ip ospf cost 400
 bandwidth 10
 frame-relay interface-dlci 102
 frame-relay class fast_vcs
!
interface Serial0.3 point-to-point
 ip address 10.128.30.17 255.255.255.248
 ip ospf cost 200
 bandwidth 10
 frame-relay interface-dlci 103
```
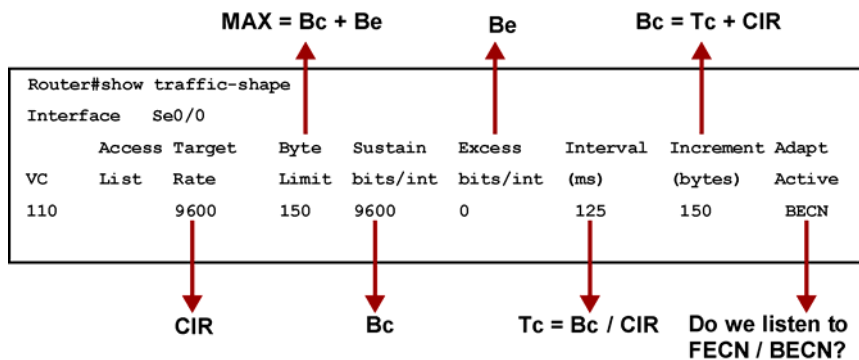
```
!
map-class frame-relay slow_vcs
 frame-relay traffic-rate 4800 9600
 frame-relay custom-queue-list 1
!
map-class frame-relay fast_vcs
 frame-relay traffic-rate 16000 64000
 frame-relay priority-group 2
!
access-list 100 permit tcp any any eq 2065
access-list 115 permit tcp any any eq 256
!
priority-list 2 protocol decnet high
priority-list 2 protocol ip normal
priority-list 2 default medium
!
queue-list 1 protocol ip 1 list 100
queue-list 1 protocol ip 2 list 115
queue-list 1 default 3
queue-list 1 queue 1 byte-count 1600 limit 200
queue-list 1 queue 2 byte-count 600 limit 200
queue-list 1 queue 3 byte-count 500 limit 200
```

0100_219

---

show traffic-shape Command

$MAX = Bc + Be$ Be $Bc = Tc + CIR$

```
Router#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
110            9600    150   9600      0        125       150       BECN
```

0100_221

CIR   Bc $Tc = Bc / CIR$ Do we listen to FECN / BECN?

1

Normalizing Custom Queueing Example

What percentage of service each queue will receive in a fully loaded network?

App	Percentage	Ave Packet size	Percent/Average size = X	$X/X_{smallest} = Y$	Y * multiplier = Z	Z * Average Packet Size = Byte Count
FTP	50%	972	50/972=.05144	.05144/.05144 = 1	1 * 3 = 3	3 * 972=2916
Telnet	25%	184	25/184=.1359	.1359/.05144= 2.65	2.65 * 3 = 7.95	8 * 184 = 1472
Other	25%	480	25/280=.0521	.0521/.05144 = 1.01	1 * 3 = 3	3 * 480 = 1440

To test your work, add all the byte counts and divide each by the total:
Total = 5828 FTP 2916/5828 = .5003 Telnet 1472/5828 = .2526 Other 1440/5828 = .2471
You want to be within +1%

© 2004 Global Knowledge - All rights reserved

3550 MLS QoS Maps

```
Dscp-cos map:
  d1 : d2 0 1 2 3 4 5 6 7 8 9
  ---------------------------------
   0 :   00 00 00 00 00 00 00 00 01 01
   1 :   01 01 01 01 01 01 02 02 02 02
   2 :   02 02 02 02 03 03 03 03 03 03
   3 :   03 03 04 04 04 04 04 04 04 04
   4 :   05 05 05 05 05 05 05 05 06 06
   5 :   06 06 06 06 06 06 07 07 07 07
      6 :   07 07 07 07

Cos-dscp map:
    cos:  0 1 2 3 4 5 6 7
    ----------------------------
    dscp:  0  8 16 24 32 40 48 56
```

```
IpPrecedence-dscp map:
  ipprec:  0 1 2 3 4 5 6 7
  ------------------------------
   dscp:  0  8 16 24 32 40 48 56


Dscp-dscp mutation map:
Default DSCP Mutation Map:
  d1 : d2 0 1 2 3 4 5 6 7 8 9
  ---------------------------------------
   0 :   00 01 02 03 04 05 06 07 08 09
   1 :   10 11 12 13 14 15 16 17 18 19
   2 :   20 21 22 23 24 25 26 27 28 29
   3 :   30 31 32 33 34 35 36 37 38 39
   4 :   40 41 42 43 44 45 46 47 48 49
   5 :   50 51 52 53 54 55 56 57 58 59
   6 :   60 61 62 63
```

© 2004 Global Knowledge - All rights reserved

Configuring PQ on Catalyst 3560 Switches

```
Switch(config)#
```

```
mls qos srr-queue output cos-map quid {cos1...cos8 |
threshold threshold-id cos1...cos8}
Interface gigabitethernet0/1
  queue-set quid
```

- **quid:** Specifies the queue ID of the CoS priority queue. (Ranges are 1 to 4 where 1 is the lowest CoS priority queue.)
- **cos1...cos8:** Specifies the CoS values that are mapped to the queue ID.
- **threshold-id** Map CoS values to a queue threshold ID. For *threshold-id*, the range is 1 to 3. For *cos1...cos8*, enter up to eight values, and separate each value with a space. The range is 0 to 7.

Default ID values are:

Queue ID	CoS Values	Queue ID	CoS Values
1	5	3	2, 3
2	0, 1	4	4, 6, 7

Configuring SRR on Catalyst 3560 Switches

```
Switch(config-if)#
```

```
srr-queue bandwidth {shape | share} weight1...weight4
```

- **Assigns SRR weights to the four egress queues**
- **Shape: Specify the weights to specify the percentage of the port that is shaped. The inverse ratio (1/*weight*) specifies the shaping bandwidth for this queue.**
- **Share: The ratios of *weight1*, *weight2*, *weight3*, and *weight4* specify the ratio of the frequency in which the SRR scheduler dequeues packets (same as WRR).**
- **Ranges for the SRR values:**
  - **Shape: The range is 1 to 65535.**
  - **Share: The range is 1 to 255.**

Global Knowledge

```
Switch(config)#
Priority-queue out
```

- **Assigns priority queue to queue 4 for the 3550 & queue 1 for the 3560**
- **Weight is ignored**
- **Queue 4 is serviced before all other queue**
- **1P3Q model**

4