WORLD METEOROLOGICAL ORGANIZATION

ISS/ET-DCST 99/Doc. 14
(21.IX.1999)

COMMISSION FOR BASIC SYSTEM
OPAG ON
INFORMATION SYSTEMS & SERVICES
EXPERT TEAM MEETING ON DATA-COMMUNICATION SYSTEMS AND TECHNIQUES

ITEM 3.3

ENGLISH only

GENEVA, 27-30 SEPTEMBER 1999

Network management issues

Submitted by Germany

The purpose of this document is to discuss some traffic management issues in an IP network.

ii. Traffic management

When going to an IP based network, the first step should be to examine the expected amount of traffic very closely to ensure that the ordered Leased Line or access line to a network has a sufficient capacity.
After installing the line, it is also recommended to check if the actual traffic meets the expected traffic. Tools to do this are listed in chapter 5.

In an environment with properly dimensioned Leased Lines between 2 centers, or Frame Relay PVC's like in RA VI, and one implemented mechanism of data transfer (either TCP/IP socket based or ftp), it normally shouldn't be necessary to have a special traffic management. However, it might be necessary to prevent the data traffic from using up all available bandwidth; in such a situation it is possible that the TCP connection used by BGP will be effected. In the worst case, the 2 BGP neighbors will lose their connection and with this the routing information.

If there is a mix of ftp and socket based transfers and also X.25 traffic on one line, some management may be needed to prevent the bulk ftp traffic from using up all the available bandwidth and causing delay to the other transfers.

There are several methods to achieve traffic management. One can use e.g. queuing mechanisms, the Type of Service (TOS) field in the IP header, or shape the traffic that is transmitted to the line. All these methods have effects on how the traffic is put on the line and may not have the desired results if configured improperly. Especially the use of the TOS field would require significant changes within the application and is therefor not recommended at this stage.

Queuing mechanisms depend on the brand of router used. Possible mechanisms for a Cisco router are First Come First Served (FCFS or FIFO) queuing, Simple Priority Queuing, Class-based (custom) queuing or Weighted Fair Queuing (which is the default setting).

FCFS Queuing is the classic algorithm: transmission occurs in the same order as messages are received.

Simple Priority Queuing combines a classification facility in the forwarding application (here IP) with a queuing algorithm in the interface. To do that, it uses a set of filters or access-list entries and applies a priority to the message. The queuing then places the messages in queues by priority, and in transmission, gives higher-priority queues preferential treatment over low-priority queues. Although it is easy to implement, one has to keep in mind that it can completely lock out a lower-priority queue for a period of time and can cause them to slow down too much.
With priority queuing, one can define 4 priorities (high, medium, normal, and low).

If MSS a in center A wanted to send it's socket based traffic to MSS b in center B with high priority, and would use port 11111 for the communication, the configuration on a Cisco router would look like the following:
(Assume that center A and center B are connected via a Leased Line that is connected to the port Serial0 on both routers.

Router at Center A:

```
!
interface Serial 0
description LL to Center B
ip address 193.105.178.5 255.255.255.252
encapsulation ppp
bandwidth 64
priority-group 1          ! this defines that traffic should be queued as defined
                          ! in priority-list 1
!
priority-list 1 protocol ip high tcp 11111
priority-list 1 default normal     ! this is an implicit default and won't show up in the
                                   ! configuration
!
```

Router at Center B:

```
!
interface Serial 0
description LL to Center A
ip address 193.105.178.6 255.255.255.252
encapsulation ppp
bandwidth 64
priority-group 1
!
priority-list 1 protocol ip high tcp 11111
!
```

XOT (X.25 over TCP) uses tcp port 1998, so by using 1998 in the above example, XOT traffic would always be transmitted with the highest priority.

Class-based or Custom queuing is more complicated to implement since it requires complex knowledge about the traffic pattern (packet size, window size of the application, sensibility to delay, etc.). It uses the same classification facility priority queuing uses in the forwarding application, but a different queuing algorithm in the interface because it places the messages into selected queues. These queues are then served in a round-robin order. The amount

removed from the queue on each pass varies by configuration. Coupled with a timer, it can be used to ensure that no class consumes more than a certain bandwidth under any circumstance. Because this method is complicated to set up, more investigation of the traffic patterns has to be done before it is possible to give a recommendation.

The standard queuing mechanism on all serial interfaces that run at or below 2Mbps is Weighted Fair Queuing (WFQ). It is enabled by default for physical interfaces that do not use LAPB, X.25 or SDLC encapsulation. It provides traffic priority management that automatically sorts among individual traffic flow without requiring any additional access lists. There are 2 categories of WFQ sessions: high bandwidth and low bandwidth. Low-bandwidth traffic has effective priority over high-bandwidth traffic, and high-bandwidth traffic shares the transmission service proportionally according to assigned weights. These weights and corresponding congestion thresholds all have default settings in the IOS software, and are results of traffic engineering done in Cisco Labs.
There are not many possibilities to change these settings.

When WFQ is enabled for an interface, new messages for high-bandwidth traffic streams are discarded after the configured or default congestive message threshold has been reached. However, low-bandwidth conversations continue to enqueue data.

With standard WFQ, packets are classified by flow. Packets with the same source IP address, destination IP address, source TCP or UDP port, or destination TCP or UDP port belong to the same flow. WFQ allocates an equal share of the bandwidth for each flow.

The newest method of ensuring a “guaranteed” bandwidth for the socket based application is to implement Class-based Weighted Fair Queuing. This is a new feature in the Cisco IOS software, and only available from the IOS 12.0.5T releases and higher. It also requires that a technique called Cisco Express Forwarding is enabled.

Class-based weighted fair queuing (CBWFQ) extends the standard WFQ functionality to provide support for user-defined traffic classes. One defines traffic classes based on match criteria including protocols, access control lists, and input interfaces. Taking into account available bandwidth on the interface, one can configure up to 64 classes and control distribution among them, which is not the case with WFQ.
A queue is reserved for each class. The characteristics for a class consist of a bandwidth, weight, and maximum packet limit. The bandwidth assigned to a class is the minimum bandwidth delivered to the class during congestion.
If a default class is configured, all unclassified traffic is treated as belonging to the default class. If no default class is configured, then by default the traffic that does not match any of the configured classes is flow classified and given best-effort treatment.

Configuring CBWFQ is basically to define and configure a class policy. The following processes are necessary:

- define traffic classes to specify the classification policy (class maps)
  This process determines how many types of packets are to be differentiated from one another

- associating policies (class characteristics) with each traffic class (policy maps)
  This process provides configuration of policies to be applied to packets belonging to one of the classes previously defined through a class map.

- attaching policies to interfaces (service policies)
  This process requires that one associates an existing policy map, or service policy, with an interface to apply the map's particular set of policies to that interface.

Below is an example of how to reserve bandwidth for the sockets application. Every other traffic is best-effort via the (implicit) default queue.
Again, the TCP port number for the sockets application is 11111, and center A is connected to Center B through a 64Kbps Leased Line. The LL is connected to the interface Serial0 on both routers.
The MSS host at center A has the IP address 10.1.1.10, and the MSS host at Center B has the IP address 10.2.2.10.

Router at center A:

```
!
! define the traffic class
class-map sockettraffic
        match access-group 120
!
! define class characteristics
policy-map GTStraffic
        class sockettraffic
        bandwidth 40    ! provides at least 40Kbps for the sockets application
        random-detect   ! selects how packets are being dropped/discarded
!
ip cef                  ! enables cisco express forwarding
!
interface Serial0
description 64Kbps leased Line to Center B
ip address 193.105.178.5 255.255.255.252
encapsulation ppp
bandwidth 64
!
! define the IP TOS (CAR) for the outgoing traffic on that interface and the actions
rate-limit output access-group 120 64000 5000 1000 conform-action
set-prec-transmit 3 exceed-action set-prec-transmit 3
! this defines that the sockets transfer always is transmitted, and gets a TOS value of 3
service-policy output GTStraffic        ! if congestion, 40Kbps are guaranteed
!
...
! access-list 120 defines the sockets traffic
access-list 120 permit tcp host 10.1.1.10 eq 11111 host 10.2.2.10 eq 11111
!
```

Router at center B:

```
!
! define the traffic class
class-map sockettraffic
        match access-group 120
!
! define class characteristics
```

```
policy-map GTStraffic
        class sockettraffic
        bandwidth 40    ! provides at least 40Kbps for the sockets application
        random-detect   ! selects how packets are being dropped/discarded
!
ip cef                  ! enables cisco express forwarding
!
interface Serial0
description 64Kbps leased Line to Center B
ip address 193.105.178.6 255.255.255.252
encapsulation ppp
bandwidth 64
!
! define the IP TOS (CAR) for the outgoing traffic on that interface and the actions
rate-limit output access-group 120 64000 5000 1000 conform-action
set-prec-transmit 3 exceed-action set-prec-transmit 3
! this defines that the sockets transfer always is transmitted, and gets a TOS value of 3
service-policy output GTStraffic        ! if congestion, 40Kbps are guaranteed
!
...
! access-list 120 defines the sockets traffic
access-list 120 permit tcp host 10.2.2.10 eq 11111 host 10.1.1.10 eq 11111
!
```

It is possible to add more classes, e.g. for the BGP traffic etc..

Traffic shaping is useful on links where a certain bandwidth is always available (like the Committed Information Rate CIR on Frame Relay links), but more bandwidth can be used if there is free capacity in the network. But, if the bandwidth exceeds the CIR and there is no free capacity, the packets will be dropped. In that case, traffic shaping set to the CIR would avoid such drops.

It should be noted here that the above described traffic management is only applicable if one has direct access to the routers that are connected to the links.
If networks (either the Internet or other by providers managed networks) are used, it is up to the center and the provider to agree on any traffic prioritization.

Another way of managing the available bandwidth and traffic is to use dedicated devices (e.g. PacketShaper from Packeteer, http://www.packeteer.com). Of course, this dedicated device might also be another single point of failure.