Configuring Class Based Weighted Fair Queueing with FRTS

Document ID: 17920

Contents

Introduction

This document provides a sample configuration for Class−Based Weighted Fair Queueing (CBWFQ) with Frame Relay Traffic Shaping (FRTS).

CBWFQ extends the standard Weighted Fair Queueing (WFQ) functionality to provide support for user−defined traffic classes. FRTS uses queues on a Frame Relay network to limit surges that can cause congestion. Data is buffered and then sent into the network in regulated amounts to ensure that the traffic fits within the promised traffic envelope for the particular connection.

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

CBWFQ is supported as of the following Cisco IOS® Software Releases depending on the platform:

- Cisco 7500 Series with Versatile Interface Processors (VIP) (distributed CBWFQ) – Cisco IOS Software Release 12.1(5)T
- Cisco 7200 Series, 2600/3600 Series, and other non−7500 Series platforms – Cisco IOS Software Release 12.1(2)T

However both the routers used for this configuration document were running Cisco IOS Software Release 12.2(2).

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

Why Use CBWFQ with FRTS?

If you have specific data to protect, CBWFQ provides a way to further specify this data using specific classes. Using CBWFQ, the weight specified for a class becomes the weight of each packet that matches the class criteria. This weight is derived from the bandwidth you assign to the class. WFQ is then applied to these classes, instead of being applied to the flows themselves, and the classes can include several flows.

Configure

In this section, you are presented with the information to configure the features described in this document.

Note: To find additional information on the commands used in this document, use the Command Lookup Tool (registered customers only) .

The table below provides a quick reference guide to entries you might see in configurations:

Field	Description
`FR Interface`	Output interface.
`subinterface`	Logical interface.
`dlci`	Data–link connection identifier. The value that specifies a permanent virtual circuit (PVC) or switched virtual circuit (SVC) in a Frame Relay network.
`class XXX`	Applies the map–class frame–relay XXX.
`map-class frame-relay XXX`	FRTS parameters.
`service-policy ZZZ`	CBWFQ.
`policy-map ZZZ`	Named policy.
`class YYY`	Names the class.
`bandwidth, policing, priority`	Specifics for this flow.
`class class-default`	Syntax and spelling matters when creating your default classes.
`class-map match-all YYY`	Establishes match criteria against which packet is checked.

`match` `access-group 101`	~~Ties the class-map to an access list.~~
`access-list 101` `permit ip any any`	~~Normal access list.~~

Note: *Cisco 7500 Series:* As of Cisco IOS Software Release 12.1(5)T, Quality of Service (QoS) policies must run in distributed mode on the Versatile Interface Processor (VIP) because Route/Switch Processor (RSP)−based QoS is no longer supported. Therefore, use the **shape** command and other commands for the modular QoS Command Line Interface (CLI) to implement Distributed Traffic Shaping (DTS) for Frame Relay interfaces on VIPs on the Cisco 7500 series. DTS combines Generic Traffic Shaping (GTS) and FRTS.

Mandatory Procedure

Configuring CBWFQ with FRTS includes the following three mandatory steps:

1. Define Class Maps (class−map).

   Establish the match criteria against which a packet is checked to determine if it belongs to a class.
2. Configure the Policy Map (policy−map) and Defining Classes (class).

   Specifies the name of the policy map. Associates specifications for bandwidth guarantees, policing, and priority to each traffic class. This process entails configuration of bandwidth, and so on, to be applied to packets belonging to one of the previously defined class−maps. For this process, configure a policy map that specifies the policy for each traffic class.
3. Attach the Service Policy to the FRTS map−class (service−policy).

   Attach the prescribed policies identified with the specific service−policy to the map−class (and thus the DLCI or subinterface where the map−class frame−relay is applied).

Network Diagram

This document uses the network setup shown in the diagram below.
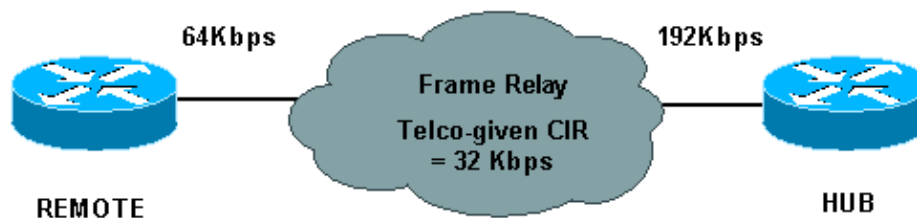
The network diagram above uses the following values:

- HUB – physical rate = 192 Kbps, guaranteed rate = 32 Kbps
- REMOTE – physical rate = 64 Kbps, guaranteed rate = 32 Kbps

Configurations

This document uses the configurations shown below.

- Hub with CBWFQ Configured

- Remote

Hub with CBWFQ Configured

```
<snip>
!
class-map match-all YYY
  match access-group 101
!
!
policy-map ZZZ
  class YYY
   bandwidth percent 50
<snip>
interface Serial0/0
  no ip address
  encapsulation frame-relay
  no fair-queue
  frame-relay traffic-shaping

 interface Serial0/0.1 point-to-point
  ip address 10.1.1.1 255.255.255.0
  frame-relay interface-dlci 16
  frame-relay class XXX
 !
 map-class frame-relay XXX
  frame-relay cir 64000
  frame-relay mincir 32000
  frame-relay adaptive-shaping becn
  frame-relay bc 8000
  service-policy output ZZZ
<snip>
!
access-list 101 permit ip host 10.0.0.1 host 11.0.0.1
```

Remote

```
interface Serial0/0
no ip address
encapsulation frame-relay
no fair-queue
frame-relay traffic-shaping
!
interface Serial0/0.1 point-to-point
ip address 10.1.1.2 255.255.255.0
frame-relay interface-dlci 16
frame-relay class XXX
!
map-class frame-relay XXX
frame-relay cir 64000
frame-relay mincir 32000
frame-relay adaptive-shaping becn
frame-relay bc 8000
!
```

Verify

This section provides information you can use to confirm your configuration is working properly.

Certain **show** commands are supported by the Output Interpreter Tool (registered customers only) , which allows you to view an analysis of **show** command output.

- **show frame–relay pvc** – Displays statistics about PVCs for Frame Relay interfaces.
- **show policy–map** – Displays the configuration of all classes comprising the specified service policy map or all classes for all existing policy maps.
- **show policy–map [interface]** – Displays the configuration of all classes configured for all service policies on the specified interface or to display the classes for the service policy for a specific PVC on the interface.

The following is sample output of the **show frame–relay pvc** command:

```
Hubrouter#show frame-relay pvc [interface interface ][dlci]
PVC Statistics for interface Serial0/0 (Frame Relay DTE)

              Active      Inactive      Deleted       Static
   Local        0            1             0             0
   Switched     0            0             0             0
   Unused       0            0             0             0

DLCI = 16, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0.1

   input pkts 0            output pkts 0           in bytes 0
   out bytes 0            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
   pvc create time 00:01:12, last time pvc status changed 00:01:12
Hubrouter#
```

You can use the following syntax with this command:

- interface – (Optional) Indicates a specific interface for which PVC information is displayed.
- interface – (Optional) Interface number containing the DLCIs for which you wish to display PVC information.
- dlci – (Optional) A specific DLCI number used on the interface. Statistics for the specified PVC are displayed when a DLCI is also specified.

The following is sample output of the **show policy–map** command:

```
Hubrouter#show policy-map
   Policy Map ZZZ
    Class YYY
      Weighted Fair Queueing
            Bandwidth 50 (%) Max Threshold 64 (packets)
    Class WWW
      Weighted Fair Queueing
            Bandwidth 25 (%) Max Threshold 64 (packets)
```

The following is sample output of the **show policy–map [interface]**.

```
Hubrouter#show policy-map interface s0/0.1
 Serial 0/0.1: DLCI 16
 Service-policy output: ZZZ (1057)
   Class-map: YYY (match-all) (1059/2)
     0 packets, 0 bytes
     30 second offered rate 0 bps, drop rate 0 bps
     Match: access-group 101 (1063)
     Weighted Fair Queueing
       Output Queue: Conversation 73
       Bandwidth 50 (%) Max Threshold 64 (packets)
       (pkts matched/bytes matched) 0/0
       (depth/total drops/no-buffer drops) 0/0/0
```

```
Class-map: WWW (match-all) (1067/3)
  0 packets, 0 bytes
  30 second offered rate 0 bps, drop rate 0 bps
  Match: access-group 102 (1071)
  Weighted Fair Queueing
    Output Queue: Conversation 74
    Bandwidth 25 (%) 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) (1075/0)
    2 packets, 706 bytes
    30 second offered rate 0 bps, drop rate 0 bps
    Match: any  (1079)
```

Other terms which you may also see in similar configurations are explained below:

- CIR – Committed Information Rate. Rate at which a Frame Relay network agrees to transfer information under normal conditions, averaged over a minimum increment of time.
- FIFO queueing – First–in, first–out queueing. FIFO involves buffering and forwarding of packets in the order of arrival. FIFO embodies no concept of priority or classes of traffic. There is only one queue, and all packets are treated equally. Packets are sent out an interface in the order in which they arrive.

Troubleshoot

There is currently no specific troubleshooting information available for this configuration.

Related Information

- **Configuring Frame Relay and Frame Relay Traffic Shaping**
- **Configuring and Troubleshooting Frame Relay**
- **Class–Based Weighted Fair Queueing**
- **Technical Support & Documentation – Cisco Systems**