

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

NÚCLEO DE COMPUTAÇÃO ELETRONICA

Diniz Martins Souto

POLÍTICAS DE FILAS EM ROTEADORES CISCO

Rio de Janeiro

2010

Diniz Martins Souto

POLÍTICAS DE FILAS EM ROTEADORES CISCO

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Redes de Computadores e Tecnologia Internet do Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro – NCE/UFRJ.

ORIENTADOR:

Moacyr Henrique Cruz de Azevedo, M. Sc., UFRJ, Brasil

Rio de Janeiro

2010

Diniz Martins Souto

POLÍTICAS DE FILAS EM ROTEADORES CISCO

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Redes de Computadores e Tecnologia Internet do Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro – NCE/UFRJ.

Aprovada em Março de 2010



Moacyr Henrique Cruz de Azevedo , M.Sc., UFRJ, Brasil

AGRADECIMENTOS

Agradeço primeiramente a Deus por não me deixar desistir, quando, muitas vezes, senti-me desmotivado e sem esperança de terminar este curso. Agradeço também aos mestres do MOT-CN por suas competências, e também aos colegas de turma por manterem uma união e ajudarem-se mutuamente e principalmente a mim.

RESUMO

SOUTO, Diniz Martins. **POLÍTICAS DE FILAS EM ROTEADORES CISCO.** Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2007.

Este trabalho está focado em apresentar as diferentes políticas de filas existentes nos roteadores Cisco com suas utilizações e finalidades. As suas principais soluções em gerência de tráfego e quais medidas a serem adotadas para solucionar os problemas de congestionamento e medidas para se evitá-lo.

ABSTRACT

SOUTO, Diniz Martins. **POLÍTICAS DE FILAS EM ROTEADORES CISCO.** Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2007.

This work is focused in demonstrate the different queue politics in Cisco routers with utilizations and applications. What are Cisco main solutions to traffic management and what measures to be adopted to solve the problems of traffic congestion and avoidance.

LISTA DE FIGURAS

	Página
Figura 1 – FIFO	15
Figura 2 – Fair Queueing	16
Figura 3 – Precedência IP	18
Figura 4 – Weighted Fair Queueing	20
Figura 5 – Custom Queueing	36
Figura 6 – Priority Queueing	39

LISTA DE TABELAS

	Página
Tabela 1 – Identificação do Fluxo WFQ	19
Tabela 2 – Comparação entre as WFQ	32
Tabela 3 – Comparação ente as 4 principais filas	42

LISTA DE ABREVIATURAS E SIGLAS

AAL	ATM Adaptation Layer
ACL	Access Control List
ATM	Asynchronous Transfer Mode
CAR	Committed Access Rate
CLI	Command Line Interface
CLNS	Connectionless Network Service
CoS	Class of Service
CR	Committed Rate – taxa acordada
DLCI	Datalink Connection Identifier
DSCP	Diffservices Code Point
DTS	Distributed Traffic Shapping
EFA	Egress Fowarding AESIC
ECN	Explict Congestion Notification
FIFO	First In First Out
GTS	Generic Traffic Shapping
MAC	Media Access Control
MLP	Multi Link PPP
MUX	Multiplexador
NLPID	Network Level Protocol ID
NSAP	Network Service Access Point
PHB	Per Hop Behavior
PR	Peak Rate – Taxa de pico
QoS	Qualidade de Serviço
RED	Random Early Detection
RSVP	Resource Reservation Protocol
RTP	Real Time Protocol
RTT	Round Trip Time
SAR	Segmentation and Reassembly
ToS	Type of Service
UDP	User Datagram Protocol
VIP	Versatile Interface Processor
VOIP	Voice Over Internet Protocol
WRED	Weighted RED

SUMÁRIO

	Página
1 INTRODUÇÃO	11
2 MOTIVAÇÃO	13
3 O CONTROLE DO TRÁFEGO	14
3.1. FIFO	14
3.2 FAIR QUEUEING	15
3.2.1 Weighted Fair Queueing	16
3.2.2 WFQ – Flow-based WFQ	19
3.2.3 DWFQ – Distributed WFQ	21
3.2.4 CBWFQ – Class-based WFQ	22
3.2.5 DCBWFQ – Distributed Class-Based WFQ	26
3.2.6 LLQ – Low Latency Queueing	29
3.2.6.1 Distributed Low Latency Queueing	31
3.3 CUSTOM QUEUEING	33
3.4 PRIORITY QUEUEING	38
4 CONCLUSÃO	43
5 REFERÊNCIAS	44
5.1 PESQUISA EM LIVROS	44
5.2 INTERNET	44
5.3 RFCs	45

1 INTRODUÇÃO

Com a rápida evolução da tecnologia e a crescente demanda por largura de banda nas comunicações surge a necessidade de se aprimorar o conceito de transmissão de dados.

É neste cenário que entram os roteadores como um meio de ligar o mundo interior com o mundo exterior, mas não é esta a única finalidade pois atualmente espera-se que estes possam transmitir dados da melhor forma possível e de forma eficiente, sabendo separar tráfegos distintos, dando-lhes uma prioridade maior ou menor conforme a necessidade.

Visto que o principal gargalo em uma corporação é o seu link de comunicação com o mundo exterior e que o aumento da largura de banda pode tornar-se inviável devido a fatores de ordem monetária, pois o aumento da mesma pode fazer o valor crescer exponencialmente, ou por não haver uma tecnologia que torne possível o seu aumento, torna-se necessário implementar, de alguma maneira, uma melhor otimização do link existente. Por isso os roteadores têm uma importante função, não só de interconectar duas zonas, mas também de tentar prover a melhor distribuição de tráfego e saber para onde encaminhar os pacotes que passam por eles.

Saber quais são os principais problemas enfrentados hoje em dia em uma rede é muito importante, principalmente se estes problemas forem de viabilidade de serviços, largura de banda reduzida, demora na resposta de certas aplicações, dados corrompidos, timeouts, atrasos, enfim, é uma grande gama de problemas que podem ocorrer.

Atualmente há uma grande necessidade de se implementar QoS nos enlaces de comunicações, principalmente os links de baixa velocidade, onde o atraso na

transmissão dos dados pode ser um fator crítico em aplicações interativas como voz e vídeo. E QoS nada mais é que priorizar um tráfego em relação a outro.

Priorizar um tráfego em relação a outro significa criar um política de fila de pacotes segundo certos critérios em que se dá uma prioridade maior a uma fila ou mais, dentre as diversas filas, do roteador.

2 MOTIVAÇÃO

Gostaria de começar por uma citação de Sally Floyd e Van Jacobson em “Random Early Detection Gateways for Congestion Avoidance de 1993”

“A detecção mais efetiva de congestionamento pode ocorrer no próprio roteador. O roteador pode distinguir, com certeza, entre atraso de propagação e persistente atraso em fila. Somente o roteador tem o comportamento de filas o tempo inteiro; a perspectiva de conexões individuais é limitada pelos padrões de chegada de pacotes. Além disso, o roteador, é compartilhado por muitas conexões ativas, com uma larga abrangência de RTT, tolerância a atrasos, requerimentos de throughput, etc.; Decisões sobre a duração e a magnitude do congestionamento transiente a ser permitido no roteador, são tomadas por ele mesmo.”

E de Leonardo Balliache

“É muito importante deixar claro que QoS não é uma solução miraculosa poderá ser aplicada com sucesso somente quando as seguintes premissas são previamente preenchidas: a carga é gerenciável e a rede está bem designada.”

Diante destes argumentos, nota-se que é extremamente necessário criar uma política de gerenciamento de todo o tráfego transiente de uma rede. E um dos lugares onde se deve implementar esta gerência é no roteador.

3 O CONTROLE DO TRÁFEGO

Este capítulo é a base deste trabalho. Não se pode falar de filas em roteadores sem se ter idéia do que é policiamento de tráfego como Committed Access Rate (CAR), e alguns modeladores de tráfego como Generic Traffic Shapping (GTS), Distributed Traffic Shapping (DTS), entre outros. O significado de Random Early Detection (RED) e suas políticas para detecção de congestionamento, Resouce Reservation Protocol (RSVP), Per Hop Behavior (PHB) e outras nomenclaturas acerca de QoS não é o escopo deste trabalho, mas é de grande importância para o entendimento do mesmo.

Este trabalho trata mais da gerência do congestionamento do que a detecção do congestionamento. Estes são fatores totalmente diferentes mas que deveriam sempre ser tratados conjuntamente.

A empresa escolhida foi a Cisco Inc. pois é a empresa que mais informações pode-se coletar. Além de dispor uma grande quantidade de informações sobre seus produtos também, oferece vasta documentação para pesquisa. Esta empresa tornou-se a referência para este trabalho.

Serão apresentadas quatro políticas de filas mais comuns e algumas variantes.

3.1 FIFO:

Esta política não implementa nenhum conceito de prioridade, à medida que os pacotes chegam são servidos. O problema desta fila é que o tráfego geralmente é em rajadas e poderá ocorrer descarte de pacotes ou longos atrasos em aplicações sensíveis ao tempo de resposta. Esta situação é chamada de "tail drop", pois os

pacotes são descartados do final da fila em caso de sobrecarga no tráfego de dados. Uma desvantagem da FIFO vem da sua própria simplicidade, pois não há a distinção dos pacotes a serem descartados, mas uma vantagem no uso da FIFO é que não há grande necessidade de processamento, permitindo ao roteador rotear pacotes muito mais rapidamente. Somente em um caso pode-se considerar útil o uso da FIFO, quando o link de saída possui uma largura de banda maior que o link da rede interna.

Em algumas literaturas pode-se também encontrar referências para FIFO como FCFS: “First Come First Serve”, ou seja, primeiro a chegar, primeiro a ser servido.

Esta política não serve para implementação de QoS, uma vez que não prioriza nenhum tipo de pacote.

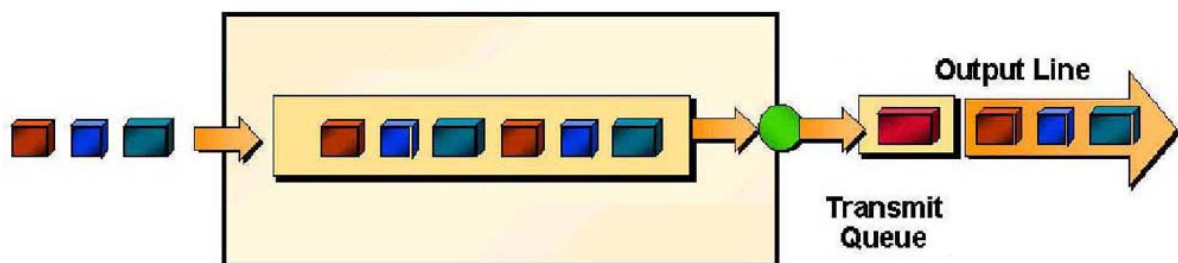


Figura1: FIFO

3.2 FAIR QUEUEING

Antes de se apresentar os algoritmos de enfileiramento adotados nos Roteadores Cisco, deve-se entender o processo de enfileiramento justo de pacotes ou fair queuing. Este algoritmo foi primeiramente proposto por Nagle em 1987 em que os roteadores mantinham filas distintas para diferentes fontes e estas filas eram

servidas por um escalonador na forma de *round-robin*. A ordem da fila de saída é realizada através do último bit de um pacote que está em uma das filas, seja ele de que tamanho for, que espera ser servido pelo escalonador para chegar ao *link* de saída.

Para se ter um exemplo deste algoritmo podemos ver na figura 2.

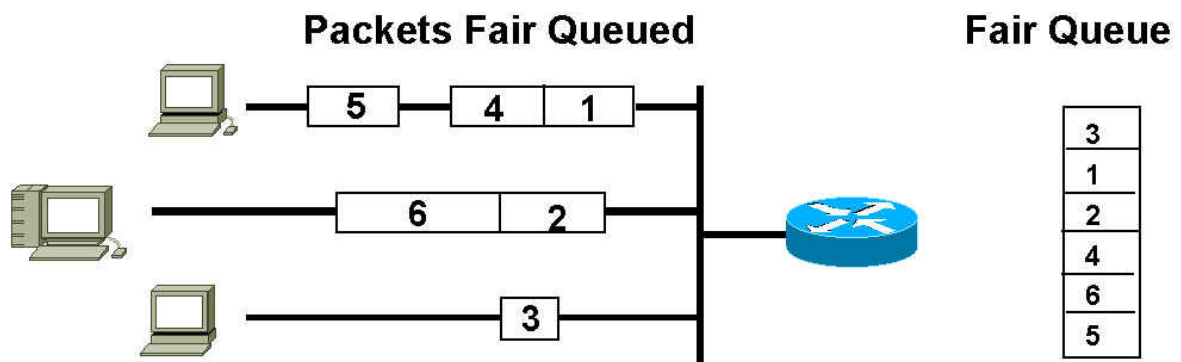


Figura 2: Fair Queueing

Neste algoritmo não há distinção entre tráfego FTP, HTTP, Mail, VoIP, entre outros, e também não faz a distinção entre tamanhos de pacotes.

Serão comentados aqui 6 variantes dessa fila.

3.2.1 Weighted Fair Queueing

Esta política assegura um tempo de resposta satisfatório às aplicações críticas e interativas. Ela escalona tráfego prioritário e também compartilha o restante da banda com os outros tráfegos de menor prioridade.

Esta política classifica o tráfego baseado em diferentes fluxos baseado no cabeçalho dos pacotes. Tais características são baseadas em endereços MAC,

redes de origem e destino, portas de origem e destino, protocolos, Frame Relay Data-Link Connection Identifier – DLCI, e Tipos de Serviços (ToS).

Existem 2 tipos de categorias de fluxos: os fluxos que demandam alta largura de banda e os que demandam baixa largura de banda. Os fluxos que demandam baixa largura de banda deverão ter prioridade, e os que demandar altas taxa de tráfego, deverão compartilhar o serviço de transmissão proporcionalmente de acordo com os seus pesos.

Esta é a política adotada quando se deseja prover um tempo de resposta consistente para redes com baixo e alto tráfego sem a necessidade de aumentar a largura de banda.

Este tipo de fila não suporta tunelamento ou encriptação de dados porque estas características modificam o conteúdo do pacote que é necessário para a sua classificação.

Esta política trabalha com precedência IP. Precedência IP, também conhecida como CoS, é uma classe de serviços que usa 3 bits do cabeçalho do quadro ethernet quando usado com o protocolo 802.1Q, criando 8 classes (0 a 7). Com os 3 bits para o campo Precedência IP têm-se as seguintes configurações e funções:

- 0 - *rotine*: estabelece a precedência rotina;
- 1 - *priority*: estabelece a precedência prioridade;
- 2 - *immediate*: estabelece a precedência imediata;
- 3 - *flash*: estabelece a precedência *flash*;
- 4 - *flash-override*: estabelece a precedência *flash-override*;
- 5 - *critical*: estabelece a precedência crítica;
- 6 - *internet*: estabelece a precedência *internetwork control*;
- 7 - *network*: estabelece a precedência controle de rede

Quanto maior o nível de classificação do pacote, maior será a prioridade no tratamento e alocação de recursos da rede. Os níveis 6 e 7 são reservados para as aplicações de controle e gerência da rede, ou seja, não é possível habilitar um pacote com 6 ou 7 e nem modificar um pacote já marcado. Todos os pacotes são normalmente marcados com o nível zero. Os 3 bits mais significantes do Tipo de Serviço no cabeçalho IP são correlacionados aos valores binários 32, 64 e 128, e são estes 3 bits usados para a precedência IP.

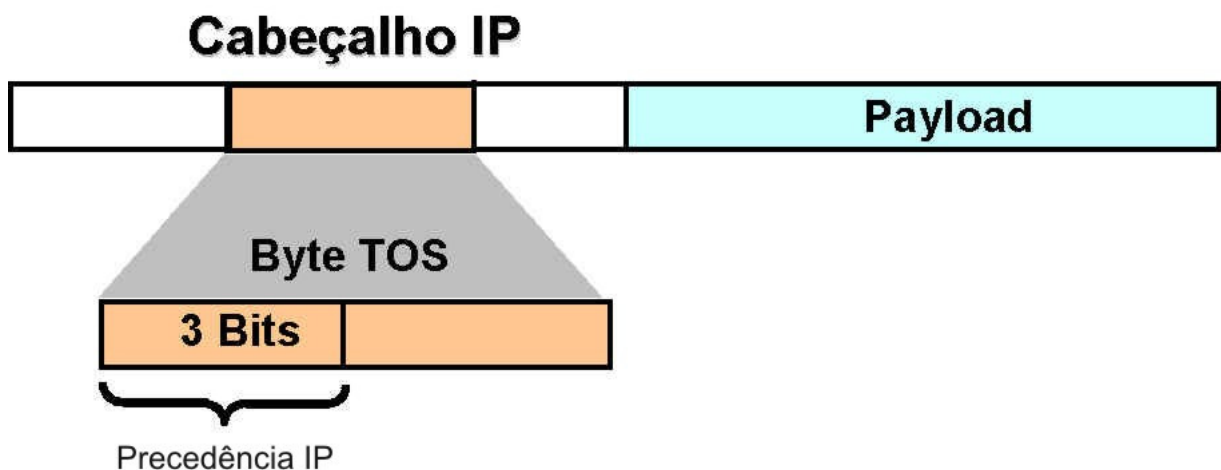


Figura 3 – Precedência IP

Em roteadores Cisco, o campo precedência IP pode ser modificado através de listas de acesso (*access list*) ou mapas de rotas – (*route maps*).

O roteador pode detectar pacotes com maiores prioridades marcados com precedência IP, desta maneira os pacotes podem ser escalonados mais rapidamente.

Daí vem o nome *weighted* pois os pacotes têm um “peso” que é um número calculado da precedência IP para um pacote num tráfego. O “peso” é calculado da seguinte forma: $\text{peso} = (32384 / (\text{precedência IP} + 1))$ este cálculo é determinante para se saber quando o pacote será servido.

3.2.2 WFQ – Flow-based WFQ

Este é o modo default de quase todas as interfaces seriais configuradas para operar a velocidades até E1 (2.048Mbps).

A característica deste algoritmo é classificar o tráfego de pacotes de acordo com o seu fluxo, onde fluxos que demandam uma menor largura de banda, possuem uma maior prioridade sobre fluxos que demandam uma maior largura de banda, de acordo com a tabela 1.

Tabela 1: Identificação do Fluxo WFQ

Protocolo	Campos de Identificação de fluxo WFQ
TCP/IP	Protocolo IP Endereço IP Origem Endereço IP Destino Porta Origem Porta Destino Campo Tipo de Serviço (ToS)
Appletalk	Rede origem, nó e socket Rede destino, nó e socket Tipo de protocolo
IPX	Rede origem, host e socket Rede destino, host e socket Tipo de protocolo nível 2
DECnet	Endereço origem Endereço destino
Frame Relay	Valor do DLCI
Transparent Bridging	Origem e destino de endereço MAC
CLNS	Endereço Origem NSAP Endereço Destino NSAP
Banyan VINES	Rede Origem e Host Rede Destino e host Tipo de protocolo nível 2
Apollo	Rede Origem, host e socket Rede Destino, host e socket Tipo de protocolo nível 2
Todos os outros	Protocolos de controle (um por fila)

Para um melhor entendimento podemos citar dois tráfegos distintos como Telnet e FTP. Telnet demanda menos largura de banda que FTP, logo, teria maior prioridade. No caso de ocorrência simultânea de dois tráfegos FTP, estes dividirão a largura de banda restante igualmente.

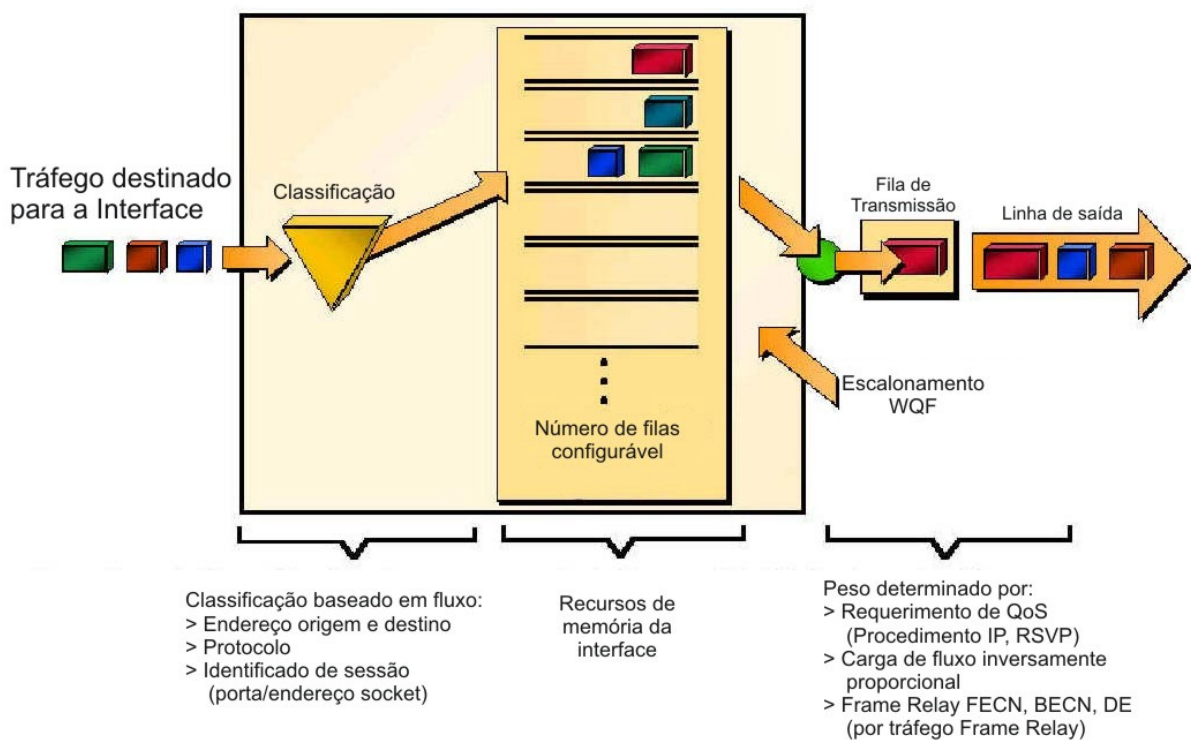


Figura 4: Weighted Fair Queueing

Para se configurar WFQ baseado em fluxo usa-se o comando:

```
Router(config-if)# fair-queue [congestive-discard-threshold [dynamic-queues [reservable-queues]]]
```

Opções:

- *congestive-discard-threshold*: (opcional) número de mensagens permitidas em cada fila, o padrão é 64 mensagens, e um novo limite deve ser uma potência de 2 entre 16 e 4096. Atingido este limite, novos pacotes são descartados.
- *dynamic-queues*: (opcional) número dinâmico de filas usado para o tráfego de melhor esforço. Os valores são 16, 32, 64, 128, 256, 512, 1024, 2048 e 4096.
- *reservable-queues*: (opcional) número reservável de filas para tráfego reservado, possui uma abrangência de 0 a 1000, o padrão é 0. Filas reserváveis são usadas para interfaces configuradas para uso com RSVP (Resource Reservation Protocol).

3.2.3 DWFQ – Distributed WFQ

Esta é uma versão especial da WFQ que opera em interfaces VIP (Versatile Interface Processor). Roteadores equipados com Interfaces VIP podem descarregar os processos de WFQ nestas interfaces e, desta forma, descarregar o seu processamento do roteador.

Esta versão roda em roteadores Cisco séries 7000 e 7500 e a implementação desta política de filas na Interface VIP difere de todas as outras plataformas.

A política para descarte de pacotes é tratada da seguinte maneira: esta tem uma contagem do número de pacotes em cada fila e uma contagem para todas as filas. Se o total de número de pacotes é menor do que o limite agregado, as filas podem armazenar mais pacotes do que o limite individual de cada fila. Quando o total de número de pacotes atinge o limite agregado, a interface começa a forçar os limites individuais da fila, de forma que se algum pacote chegar neste momento em que o limite excedeu o da fila individual, o pacote é descartado.

Todo o tráfego não-IP é tratado por uma única fila.

DWFQ tem algumas restrições consideráveis tais como não poder ser configurado em subinterfaces, não é suportado por encapsulamento ATM AAL5-MUX e AAL5-NLPID, não é suportado em Fast Ether Channel, interfaces de tunelamento ou interfaces virtuais como MLP.

Os pacotes são classificados por fluxo. Pacotes com mesmo endereço IP origem e destino, portas TCP ou UDP origem e destino, e protocolo, pertencem ao mesmo fluxo. DWFQ aloca uma fatia igual da largura de banda para cada fluxo.

Como dito anteriormente, DWFQ roda em uma interface VIP, como por exemplo uma HSSI (High-Speed Serial Interface). Os comandos usados nesta interface são:

- *fair-queue aggregate-limit*: define o número máximo de pacotes em todas as filas juntas.
- *fair-queue individual-limit*: define o número máximo de pacotes em filas individuais.
- *fair-queue limit*: define o limite máximo de uma fila para uma classe DWFQ específica.
- *fair-queue qos-group*: classifica os pacotes levando em conta o número do grupo QoS interno.
- *fair-queue tos*: classifica os pacotes levando em conta o campo ToS.

3.2.4 CBWFQ – Class-based WFQ

Esta é uma extensão da WFQ para prover funcionalidade e suporte às classes de tráfego definidas pelo usuário. Estas classes de tráfego são definidas

baseadas segundo critérios como protocolo, ACL e interface de entrada. Qualquer pacote que satisfaça o critério definido constitui o tráfego para aquela classe.

Cada classe tem uma fila FIFO associada a ela e todo tráfego pertencente àquela classe é direcionado à fila apropriada.

Para se definir uma classe deve-se assinalar uma largura de banda, um peso e um limite máximo de pacotes. A largura de banda é assegurada mesmo em períodos de congestionamento.

Caso a fila atinja seu limite máximo de pacotes a chegada de novos pacotes força, o efeito do *tail drop* ou *packet drop*, dependendo de como foi configurada esta classe.

Este policiamento deverá ser usado juntamente com o WRED, que será descrito mais a frente. É importante notar que caso se pretenda usar WRED com *packet drop* ao invés de *tail drop*, deve-se assegurar que o WRED não esteja configurado na mesma interface onde foi aplicada esta política.

O efeito padrão para CBWFQ é o *tail-drop* a não ser que se configure explicitamente para descartar pacotes como um meio de evitar o congestionamento.

Entende-se por *tail-drop* um algoritmo de gerência de fila em que o roteador decide por descartar pacotes quando a fila chega ao seu máximo de capacidade, continuar a descartar novos pacotes até que a fila tenha espaço para aceitar novos pacotes. A perda de pacotes força o remetente a entrar em *slow-start* que é um mecanismo de controle de congestionamento para se reduzir o fluxo e não enviar mais dados do que a rede pode suportar. Um dos grandes problemas do *tail drop* é que ele pode levar à sincronização global de TCP.

O efeito *packet drop* é um algoritmo mais elaborado como RED (Random Early Detection) ou WRED (Weighted Random Early Detection) que são algoritmos para se evitar congestionamento.

O algoritmo RED monitora o tamanho médio da fila, quando a fila aumenta, a probabilidade de se descartar um pacote também aumenta. Se o roteador estiver habilitado para trabalhar com ECN (Explicit Congestion Notification), poderá marcá-los para descarte mas não necessariamente o fará pois isso depende de probabilidades estatísticas de acordo com o algoritmo.

WRED é uma extensão do algoritmo RED, onde em uma fila os pacotes podem ter vários níveis de prioridades para descarte. O nível de cada fila está associado à CoS ou DSCP. Este processo de marcar o pacote para descarte ou descartá-lo chama-se gerência ativa de fila.

Para configurar CBWFQ em uma interface deve-se primeiro definir um mapa de classe e depois um mapa de policiamento. Para definir o nome do mapa de classe usa-se o comando:

```
Router(config)# class-map <nome do mapa de classe>
```

Depois deve-se executar um dos comandos:

```
Router(config-cmap)# match access-group {access-group | <nome do grupo de acesso>}
```

- define o nome da ACL (*Access Control List*) e serve para verificar se um pacote pertence à classe.

```
Router(config-cmap)# match input-interface <nome da interface>
```

- define o nome da interface de entrada usada como critério de avaliação para checar se o pacote pertence à classe.

Router(config-cmap)# match protocol <protocol>

- define o protocolo usado como critério de avaliação para checar se o pacote pertence à classe.

Router(config-cmap)# match mpls experimental <número>

- define o valor do campo EXP a ser usado como critério de avaliação para checar se o pacote pertence à classe.

Para definir o mapa de policiamento usa-se o comando *police-map* que difere de configuração se é aplicada *tail-drop* ou *WRED*.

Os comandos a serem usados com *tail-drop* são (devem seguir a ordem):

Router(config)# policy-map <nome do mapa de policiamento>

- define o nome do mapa de policiamento a ser criado ou modificado.

Router(config-pmap)# class <nome da classe>

- define o nome da classe a ser criada e incluída no serviço de policiamento.

Router(config-pmap-c)# bandwidth {bandwidth-kbps | percent <porcentagem>}

- define a quantidade da largura de banda em kbps ou a porcentagem da largura de banda disponível a ser designada para a classe.

Router(config-pmap-c)# queue-limit <número de pacotes>

- define o número máximo de pacotes que pode ser enfileirado para a classe.

Já os comandos a serem usados com *WRED* são (devem seguir a ordem):

Router(config)# policy-map <nome do mapa de policiamento>

Router(config-pmap)# class <nome da classe>

```

Router(config-pmap-c)# bandwidth {bandwidth-kbps | percent <porcentagem>}
Router(config-pmap-c)# random-detect: habilita WRED
Router(config-pmap-c)# random-detect exponential-weighting-constant <expoente>
ou
Router(config-pmap-c)# random-detect precedence <precedência> <limite mínimo>
<limite máximo> <limite para marcar pacotes>

```

- primeiro comando define o valor do peso exponencial usado para calcular o tamanho médio da fila.
- segundo comando configura os parâmetros do *WRED* com uma precedência IP específica.

3.2.5 DCBWFQ – Distributed Class-Based WFQ

Esta política é uma extensão da WFQ para prover suporte ao tráfego definido pelo usuário na VIP. Estes tráfegos definidos pelo usuário são configurados na Modular QoS CLI (Modular QoS Command Line Interface). A Modular QoS CLI é um utilitário que tem como principais funções definir uma classe de tráfego, avaliar este tráfego, criar um serviço de policiamento e associá-lo a uma classe de tráfego, e também vincular o serviço de policiamento à uma interface

Na Modular QoS CLI deverão ser especificados um limite máximo de pacotes e um serviço de policiamento. Todo tráfego pertencente a esta classe terá assegurado a sua largura de banda. O tráfego excedente sofrerá um *tail drop* ou *packet drop*, dependendo de como foi configurada a política de policiamento.

DCBWFQ permite que seja configurada a largura de banda alocada para uma classe de tráfego. Pode-se configurar até 64 classes de tráfego por interface.

Após configuradas as classes, e no caso de haver largura de banda ainda disponível, esta disponibilidade é dividida proporcionalmente entre todas as classes

configuradas, diferentemente da WFQ que divide a largura de banda disponível igualmente a todos os fluxos de tráfego.

Todo o tráfego que não corresponder a uma classe pré-definida é classificado como parte do tráfego default. A largura de banda alocada para o tráfego default é igual à largura do link menos a largura de banda de todas as classes pré-definidas. Será sempre reservado, pelo menos, um por cento do link para o tráfego default.

Os comandos usados por esta política são mais usados para modificar ou monitorar uma classe de tráfego existente, por serem comandos definidos pelo usuário e usados na Modular QoS CLI com dito anteriormente.

Para modificar a largura de banda para uma classe de tráfego existente devem ser dados, na ordem, os comandos:

```
Router(config)# policy-map <mapa de policiamento>
```

- define o nome do mapa de policiamento a ser criado ou modificado.

```
Router(config-pmap)# class <nome da classe>
```

- define o nome da classe de tráfego cuja largura de banda deseja-se alterar.

```
Router(config-pmap-c)# bandwidth <largura de banda em kbps>
```

- define o novo valor a ser reservado para a classe de tráfego

Para modificar o limite da fila para uma classe de tráfego existente devem ser dados, na ordem, os comandos:

```
Router(config)# policy-map <nome do mapa de policiamento>
```

```
Router(config-pmap)# class <nome da classe>
```

```
Router(config-pmap-c)# queue-limit <número de pacotes>
```

- define o novo valor máximo de pacotes *que podem* ser enfileirados para a classe de tráfego. O valor padrão e máximo de pacotes é 64.

Para monitoramento usam-se os seguintes comandos:

Router# show policy-map

- exibe todos os policiamentos de tráfego configurados.

Router# show policy-map <nome do mapa de policiamento>

- exibe o policiamento de tráfego específico.

Router# show policy-map <interface>

- exibe estatísticas e configurações de entrada e saída da interface definida.

Router# show policy-map interface <interface> input

- exibe estatísticas e configurações de policiamento de entrada da interface definida.

Router# show policy-map interface <interface> output

- exibe estatísticas e configurações de policiamento de saída da interface definida.

Router# show policy-map [interface [<interface> [input | output] [class <nome da classe>]]]]

- exibe estatísticas e configurações para o nome da classe configurada no policiamento.

3.2.6 LLQ – Low Latency Queueing

A LLQ é uma CBWFQ que tem uma característica singular por ter uma fila prioritária. Desta forma pode-se dar prioridade a tráfegos de tempo real. CBWFQ permite definir classes de tráfego e depois ajustar características para aquela classe, como por exemplo, designar um mínimo de largura de banda para uma classe. A LLQ irá prover uma prioridade específica para CBWFQ, reduzindo o *jitter* em uma conversação.

Apesar de ser possível colocar mais de um tipo de tráfego de tempo real dentro de uma fila prioritária, a Cisco recomenda que se coloque apenas tráfego de voz pois ele é um tráfego bem comportado e mais sensível a *jitter*. Colocando-se tráfego de vídeo nesta fila poderia ocorrer variação no *delay*.

A largura de banda alocada para a fila prioritária é configurada manualmente, garantindo assim uma largura de banda para a classe.

Em caso de congestionamento, o método usado para descarte de pacotes, quando a largura de banda para a fila prioritária é excedida, é o policiamento WRED. Devido ao tráfego de voz ser um tráfego UDP, ele não se adapta a este policiamento, pois ele não é orientado a conexão e, além disso, a perda de um pacote UDP no VOIP não afetaria significativamente a transmissão.

Isto é muito mais parecido a um limitador CAR (*Committed Access Rate* ou taxa de acesso acordada) que é uma taxa de transferência garantida por algum critério como interface de entrada, grupo de QoS, precedência IP ou lista de acesso. Uma exceção é quando esta limitação de tráfego é utilizada em condições de congestionamento, visto que, quando não há congestionamento, o tráfego prioritário pode utilizar taxa de tráfego acima do configurado.

Este método utiliza o algoritmo de balde de tokens, descartando o excesso de tráfego.

O tráfego prioritário fica restrito à largura de banda alocada para que se possa evitar que outros tráfegos não prioritários fiquem sem ser servidos. Com a medição, as classes de tráfego são policiadas e taxadas individualmente.

Os comandos usados para configurar uma LLQ são:

```
Router(config-pmap-c)# priority <largura de banda>
```

- reserva uma fila prioritária específica para esta classe de tráfego

```
Router(config-if)# max-reserved-bandwidth <porcentagem>
```

- muda a largura de banda máxima que pode ser configurável para CBWFQ, LLQ e IP RTP. O valor padrão percentual é 75.

```
Router# show queue <tipo da interface><número da interface>
```

- exibe as configurações e estatísticas de enfileiramento de uma interface em particular.

```
Router# debug priority
```

- exibe a saída da fila prioritária, útil para saber se os pacotes estão sendo dropados.

```
Router# show policy-map interface <nome da interface>
```

- exibe a configuração *de todas as* classes configuradas para todo policiamento de tráfego de uma determinada interface, informa inclusive se pacotes são dropados na fila prioritária.

3.2.6.1 **Distributed Low Latency Queueing**

Esta política é usada em roteadores Cisco 7500 nas interfaces VIP e tem como característica a capacidade de definir um comportamento de baixa latência para uma classe de tráfego. Esta política também introduz a capacidade de limitar a taxa de transmissão de um equipamento em um circuito. Antes da introdução da DLLQ, o máximo de transmissão não era um parâmetro configurado pelo usuário. Partículas poderiam, então, se acumular no circuito de transmissão sem limitação, ocasionando altas latências indesejadas.

Esta política veio a permitir que o usuário definisse o número de partículas que pode existir num anel de transmissão, baixando, substancialmente a latência dos pacotes transmitidos naquele anel.

Para se ter uma idéia mais generalizada sobre o assunto, a Cisco criou uma tabela para melhor compreensão destas filas.

Os comandos podem ser usados para reservar uma parte fixa ou percentual de uma largura de banda a uma classe de tráfego.

Os comandos a seguir são usados nesta política de fila para reservar uma parte fixa da largura de banda disponível e devem ser dados na ordem:

Router(config)# policy-map <nome do policiamento>

- define o nome do policiamento a ser configurado e habilita o modo de configuração de mapa de policiamento.

Router(config-pmap)#class <nome da classe>

- define o nome da classe já existente no serviço de policiamento e habilita o modo de configuração de classe de mapa de policiamento.

Router(config-pmap-c)# priority kpbs [bytes]

- reserva uma certa quantidade de bytes da largura de banda disponível para o tráfego CWFQ para a fila prioritária.

Os comandos a seguir são usados nesta política de fila para reservar uma parte percentual da largura de banda disponível e devem ser dados na ordem:

Router(config)# policy-map <nome do policiamento>

Router(config-pmap)#class <nome da classe>:

Router(config-pmap-c)# priority percent <porcentagem>

- reserva uma porcentagem do tráfego disponível a uma determinada fila prioritária para o tráfego CBWFQ.

Tabela 2 Comparação entre as WFQ

WFQ	DWFQ
WFQ baseado em fluxo: <ul style="list-style-type: none"> • Por peso: quando pacotes estão classificados, por ex.: (RSVP) • Enfileiramento Justo: quando pacotes não estão classificados, por ex.: tráfego de melhor esforço 	DWFQ baseado em fluxo: <ul style="list-style-type: none"> • Enfileiramento justo e sem peso DWFQ baseado em classe: <ul style="list-style-type: none"> • Por peso • QoS baseado em grupo • Baseado em ToS(Tipo de Serviço)
Roda em qualquer plataforma IOS da Cisco	Roda somente nas VIP (Versatile Interface processor)
CBWFQ	DCBWFQ
WFQ baseado em classe <ul style="list-style-type: none"> • Por peso • Alocação de banda pode ser especificada para uma determinada classe de tráfego 	WFQ distribuído baseado em classe <ul style="list-style-type: none"> • Por peso • Alocação de banda pode ser especificada para uma determinada classe de tráfego
Roda em qualquer plataforma IOS da Cisco	Roda somente nas VIP (Versatile Interface Processor)

3.3 CQ - CUSTOM QUEUEING

A Custom Queueing permite definir uma certa quantidade de bytes a se enviar em uma fila toda vez que esta fila é atendida. Este processo permite compartilhar o canal entre as aplicações de uma maneira personalizada e com os mínimos recursos definidos. Pode-se também definir o número máximo de pacotes em cada fila.

As filas são servidas em um escalonamento Round Robin. Se alguma das filas estiver vazia o escalonador irá enviar dados da próxima fila que tiver pronta para transmitir.

Quando se habilita CQ em uma interface o sistema cria 17 filas. A fila de número 0 é uma fila de sistema, e as filas de 1 a 16 são configuráveis por bytes ou pacotes. Uma fila não pode ser configurada por byte e outra fila por pacote, todas as filas devem ser do mesmo tipo. A fila 0 é uma fila que será servida e esvaziada antes de todas as outras filas. Esta fila é usada para pacotes de alta prioridade como pacotes de sinalização e *keepalive*, e nenhum outro tráfego poderá se utilizar desta fila.

Quando uma fila é processada, ela é servida até que o número de bytes enviados exceda o limite estabelecido ou a fila seja esvaziada.

A Custom Queueing assegura que nenhuma aplicação ou grupo de aplicações obtenha mais do que a porção preestabelecida quando a linha está sob congestionamento. Esta política é uma política estática como a Priority Queueing, e não se adapta dinamicamente em casos de baixo ou alto tráfego.

Deve-se determinar o byte limite de cada fila e este processo requer alguma atenção pois pode gerar confusões indesejadas . Supondo que se especifique para uma fila o limite de 100 bytes, mas o protocolo tem um tamanho máximo de 1024

bytes, ao começar a transmissão o limite de 100 bytes é atingido mas todo o pacote é transmitido. Daí o que se transmitiu foram 1024 bytes e não 100 bytes.

É importante saber calcular o byte limite para se obter o melhor desempenho, pois se o byte limite for ajustado para um valor muito baixo, poderá gerar resultados como o anteriormente citado e, se for ajustado para um valor muito alto, poderá gerar oscilações na distribuição de tráfego no link.

A Cisco recomenda que alguns passos sejam seguidos para se calcular o byte limite:

1) Dividir a percentagem de largura de banda que se deseja alocar para cada fila pelo tamanho do pacote em bytes. Como exemplo, suponha um pacote de um protocolo A de 1086 bytes, de um protocolo B de 291 bytes e de um protocolo C de 831 bytes. Deseja-se alocar 20% para o protocolo A, 60% para o protocolo B e 20% para o protocolo C.

As médias ficariam: 20/1086 ou 0.01842, 60/291 ou 0.20619 e 20/831 ou 0.02407

2) Acertar os valores para que o menor valor seja 1, isto é, dividir os valores pelo de menor valor:

$$\begin{aligned} 0.01842 / 0.01842 &= 1 \\ 0.20619 / 0.01842 &= 11.2 \\ 0.02407 / 0.01842 &= 1.3 \end{aligned}$$

Em casos de números fracionados, eles deverão ser arredondados para cima, ficando então os valores: 1, 12 e 2.

Este arredondamento para cima deve-se ao fato de que poderão ser enviados alguns bytes a mais, como o próprio algoritmos já iria fazer.

3) Multiplicar o número obtido pelo número de bytes do pacote.

$$\begin{aligned} 1 \times 1086 &= 1086 \\ 12 \times 291 &= 3492 \\ 2 \times 831 &= 1662 \end{aligned}$$

Estes valores serão os bytes limites de cada fila.

4) Como, inicialmente, era a intenção de se distribuir a banda em 20% para a primeira fila, 60% para a segunda fila e 20% para a terceira fila, devemos dividir cada um dos valores pela soma a fim de se ter uma idéia da alocação de banda de cada fila.

$1086 + 3492 + 1662 = 6240$ é o total da largura de banda.

$1086 / 6240 = 17.4\%$ para a primeira fila

$3492 / 6240 = 56\%$ para a segunda fila

$1662 / 6240 = 26.6\%$ para a terceira fila

Se estes valores não estiverem satisfatórios, aconselha-se multiplicar os valores 1, 11.2 e 1.3 por um valor. Este valor pode ser 2, 3, 4 ou qualquer número que melhor se aproximar da percentagem desejada. Exemplo.

$1 \times 2 = 2$ arredondando ficará 2

$11.2 \times 2 = 22.4$ arredondando ficará 23

$1.3 \times 2 = 2.6$ arredondando ficará 3

Repetindo-se o processo 3 em diante.

$2 \times 1086 = 2172$

$23 \times 291 = 6693$

$3 \times 831 = 2493$

Somando-se os valores $2172 + 6693 + 2493 = 11358$

Depois divide-se os valores novos pelo novo valor total.

$2172 / 11358 = 0.191$ ou 19%

$6693 / 11358 = 0.589$ ou 58%

$2493 / 11358 = 0.219$ ou 21%

Estes valores estão mais próximos da percentagem desejada, mas deve-se notar que o valor do byte limite estará 2 vezes maior, e, se o fator de multiplicação for muito elevado, poderá ocorrer perda de performance com o aumento de *jitter* em pacotes RTP como VOIP.

A Cisco informa que é interessante usar a CQ em pontos de congestionamento para garantir largura de banda a tráfego prioritário.

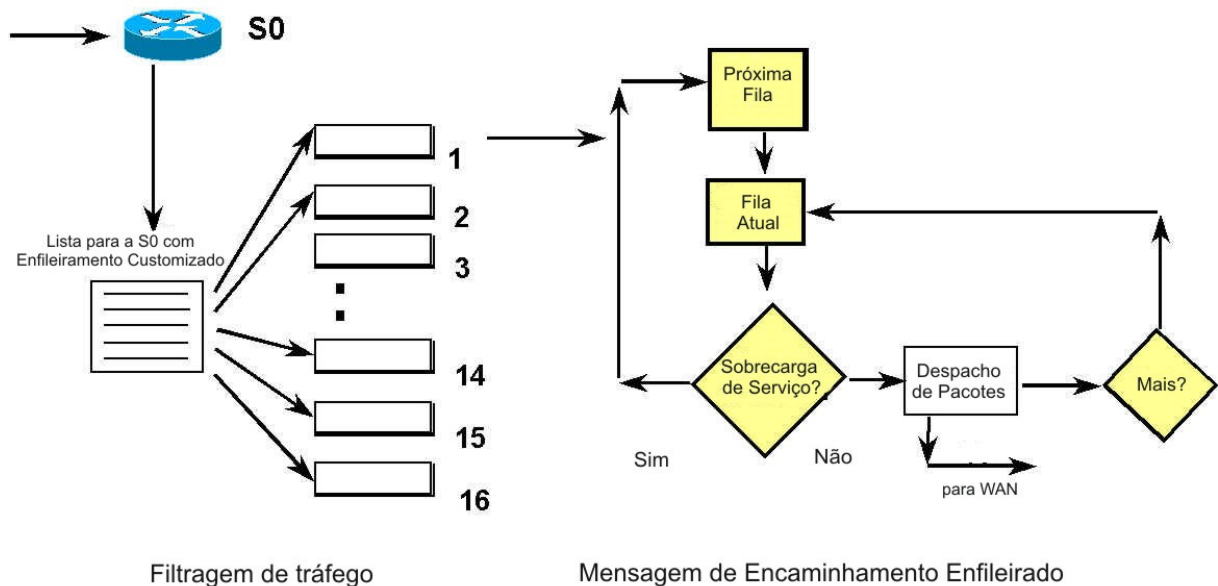


Figura 5: Custom Queueing

Os comandos a seguir são usados para a configuração da Custom Queueing e devem seguir a ordem:

```
Router(config)# interface <tipo da interface><número da interface>
```

- especifica a interface e entra no modo de configuração

```
Router(config-if)# custom-queue-list <lista>
```

- designa uma lista CQ para a interface. O argumento da lista pode ser qualquer número entre 1 e 16.

Deve-se definir também o tamanho máximo da fila que pode ser por pacotes, bytes ou média em bytes:

```
Router(config)# queue-list <número da lista> queue <número da fila> limit <limite>
```

- define o número máximo de pacotes que são permitidos em cada CQ. O argumento *<limite>* define o número de pacotes que podem ser enfileirados de uma única vez, este número varia de 0 a 32767.

Router(config)# queue-list <número da lista> queue <número da fila> byte-count <número do byte-count>

- define o número médio de bytes encaminhados por fila. O argumento *<número do byte-count>* define o número médio de bytes que o sistema permite ser encaminhado por uma fila.

Pode-se definir quais pacotes podem entrar no roteador baseado no protocolo usando-se os comandos:

Router(config)# queue-list <número da lista> protocol <tipo de protocolo> <número da fila> <palavra-chave da fila> <valor da palavra chave>

- define prioridades de enfileiramento baseado no protocolo. O argumento *<palavra chave da fila>* fornece opções adicionais como *byte-count*, serviços TCP e número da porta, e listas de acesso como AppleTalk, IP, IPX, VINES, ou XNS.

Router(config)# queue-list <número da lista> interface <tipo da interface> <número da interface.><número da fila>

- estabelece uma CQ baseado em pacotes que entram por uma interface específica.

Router(config)# queue-list <número da lista> default <número da fila>

- pacotes que não satisfizerem nenhuma das outras filas serão definidos para a fila padrão.

Router# show queue <tipo da interface> <número da interface>

- exibe o conteúdo dos pacotes dentro de uma fila ou circuito virtual.

Router# show queueing custom

- exibe o status das listas CQ.

Router# show interfaces <tipo da interface> <número da interface>

- exibe o status de saída atual das CQ quando habilitadas.

3.4 PRIORITY QUEUEING

É a política em que se quer dar prioridade a certos tráfegos. Pode-se definir uma série de filtros baseado em características de pacotes.

São definidas 4 filas de prioridades, cada uma com prioridade sobre a outra. A fila de maior prioridade é servida até que esteja vazia. Depois a segunda fila de maior prioridade é servida até ser esvaziada, e assim por diante. Se algum pacote de maior prioridade chegar, ele será certamente servido. Para cada pacote a ser enviado na linha, as 4 filas são verificadas e o pacote no topo da fila de maior prioridade é selecionado para a transmissão.

Define-se um tamanho máximo para cada fila e quando o limite é excedido os pacotes em excesso são descartados.

Uma lista de prioridades é um conjunto de regras que descreve como os pacotes serão classificados em cada fila. Esta lista poderá descrever também uma prioridade padrão ou os limites do tamanho de cada fila.

Os pacotes são classificados segundo os critérios:

- Tipo de Protocolo ou subprotocolo
- Interface de entrada

- Tamanho do pacote
- Fragmentos
- Lista de Acesso

Pacotes de gerência *keepalive* oriundos de servidores são sempre assinalados com a fila de maior prioridade. Qualquer outro tráfego de controle deve ser classificado/configurado manualmente.

A Priority Queueing provê o tempo de resposta mais rápido de todas as políticas apresentadas.

Deve-se usar este método com bastante cautela, pois tráfegos de baixa prioridade terão, quase sempre, seus acessos negados em relação aos de alta prioridade. Para se evitar este tipo de falta de acesso recomenda-se usar moldagem de tráfego CAR para limitar o tráfego de maior prioridade.

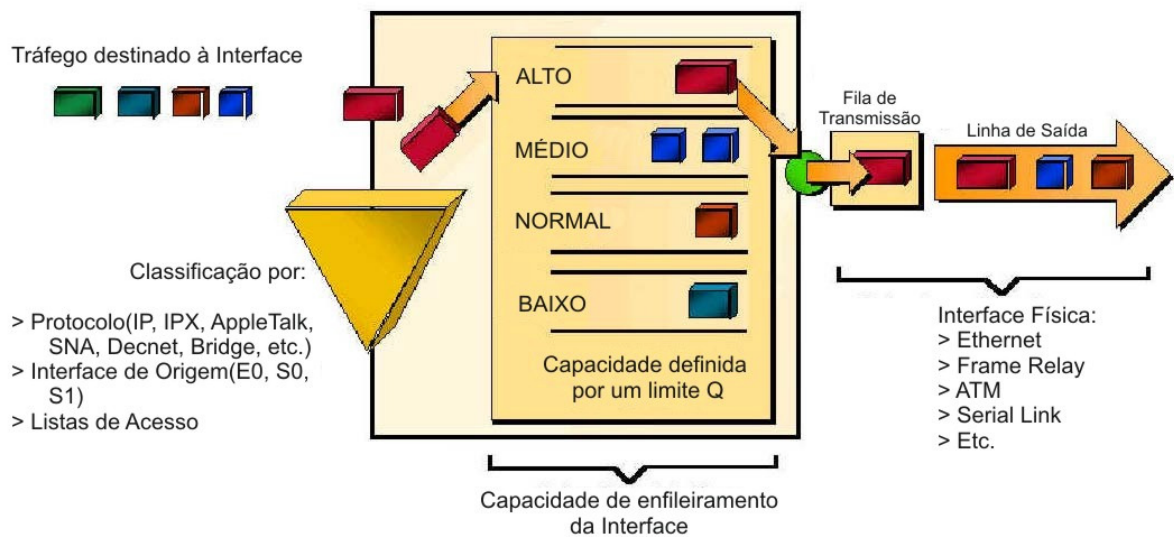


Figura 6: Priority Queueing

A PQ produz maior overhead, é aceitável em enlaces de baixa velocidade, mas deixa de ser aceitável em enlaces de alta performance, como ethernet, pois o sistema demora mais tempo para comutar os pacotes pois eles têm de ser classificados pelo processador.

Priority Queueing usa uma configuração estática e não se adapta dinamicamente mediante as condições de tráfego, além de também não ser suportado em nenhum tunelamento.

Para configura PQ nos roteadores usam-se os seguintes comandos que devem seguir a ordem:

Router(config)# priority-list <número da lista> protocol <nome do protocolo> {high | medium | normal | low} <palavra-chave da fila> <valor da palavra chave>

- define as prioridades de enfileiramento baseado no tipo de protocolo.

Router(config)# priority-list <número da lista> interface <tipo da interface> <número da interface> {high | medium | normal | low}

- define as prioridades de enfileiramento para pacotes que entram em uma determinada interface.

Router(config)# priority-list <número da lista> default {high | medium | normal | low}

- define a fila prioritária para os pacotes que não correspondem a nenhuma regra na lista prioritária.

O comando a seguir especifica o tamanho máximo das filas prioritárias:

Router(config)# priority-list <número da lista> queue-limit [<limite alto> [limite medio [<limite normal> [<limite baixo>]]]

- define o número máximo de pacotes permitido em cada uma das filas prioritárias. Caso não seja definido um tamanho, é adotado, então, o tamanho padrão das filas que são, limite alto = 20, limite médio = 40, limite normal = 60 e limite baixo = 80.

Os comandos a seguir atribuem uma lista de prioridade a uma interface e devem seguir a ordem:

Router(config)# interface <tipo da interface> <número da interface>

- define a interface e entra no modo de configuração da interface.

Router(config-if)# priority-group <número da lista>

- define a lista de prioridade à interface.

Os comandos a seguir servem para monitorar o tráfego de entrada e saída das interfaces:

Router# show queue <tipo da interface> <número da interface>

- mostra o conteúdo dos pacotes dentro de uma fila em uma interface particular.

Router# show queueing priority

- exibe os status das listas de prioridade de enfileiramento.

A Tabela 3 mostra uma comparação entre os 4 principais métodos de enfileiramento da Cisco.

Tabela 3 – Comparação entre as 4 principais filas

	WFQ Baseado em fluxo	CBWFQ/DCBWFQ
Número de filas	Número de filas configurável (256 filas de usuários por default)	Uma fila por classe; até 64 classes
Tipo de serviço	<ul style="list-style-type: none"> • Assegura justiça entre todos os fluxos de tráfegos baseados por peso • PQ estrito está disponível através do uso das características de prioridade IP RTP ou prioridade <i>Frame Relay</i> 	<ul style="list-style-type: none"> • Fornece largura de banda garantida por classe para toda classe de tráfego definida pelo usuário • Fornece suporte para WFQ baseado em fluxo para toda classe de tráfego não definida pelo usuário
Configuração	Não requer configuração	Requer configuração
	CQ	PQ
Número de filas	16 filas de usuário	4 filas de usuário
Tipo de serviço	<ul style="list-style-type: none"> • Serviço <i>round-robin</i> 	<ul style="list-style-type: none"> • Filas de alta prioridade são servidas primeiro • Prioridade absoluta: assegura características de enfileiramento prioritário a tráfego crítico ou de mais alta prioridade na interface Frame Relay PVC
Configuração	Requer configuração	Requer configuração

4 CONCLUSÃO

Atualmente fala-se muito em QoS, esta é a palavra do momento e todas as empresas procuram prover, de alguma forma, um meio de diferenciar tráfegos na rede, principalmente VOIP. Esta palavra foi encontrada em todos os fabricantes de roteadores, visto que todos estão conscientes da importância deste tipo de tráfego em especial. E, de fato, este é o tipo de tráfego que é o mais sensível de todos pois é intolerante a atrasos demasiados e a *jitter*.

É notório que algumas empresas são mais especialistas em fabricação de switches do que de roteadores, e mesmo assim não deixam de aplicar QoS em portas Ethernet Gigabit.

Quanto às políticas de filas apresentadas pela Cisco, o que se percebe é a presença de duas políticas principais: a Fair Queueing e a Priority Queueing, elas se desdobram em algumas variantes.

Vale a pena, também, lembrar as palavras de Leonardo Balliache: que só se pode aplicar alguma Qualidade de Serviço quando se tem o controle do seu domínio ou, pelo menos, tem-se algum serviço diferenciado nos links que não estão sob o controle do administrador. Não adiantará nada monitorar e estudar qual tráfego será prioritário ou mais sensível a atrasos se, em algum momento, não se tem como gerenciar um link.

5 REFERÊNCIAS

5.1 PESQUISA EM LIVROS

- PAQUET, CATHERINE *Construindo Redes Cisco de Acesso Remoto*
- KENSHAV, S. *An Engeneering Aproach to Computer Networking*
- VEGESNA, Srinivas *IP Quality of Service, Cisco Press 2001*
- CHAO, H. JONATHAN and LIU, BIN *High Performance Switches and Routers*
- FLANNAGAN, Michael E. *Administering Cisco QoS for IP Networks*
- JOSEPH, Vinod and CHAPMAN, Bred *Deploying QoS for Cisco IP and Next Generation Networks*

5.2 INTERNET

- <http://www.ecse.rpi.edu/Homepages/shivkuma/teaching/sp2001/readings/fq.pdf>
Acesso em 10/03/2010
- <http://opalsoft.net/qos/WhyQoS.htm> Acesso em 08/03/2010
- http://www.ciscosistemas.org/en/US/docs/ios/12_0t/12_0t5/feature/guide/cbwfq.html Acesso em 08/03/2010
- http://www.cisco.com/en/US/docs/ios/11_2/feature/guide/wred_gs.html Acesso em 12/03/2010
- <http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120xe/120xe5/mqc/mcli.htm> Acesso em 07/03/2010
- <http://www.cisco.com/en/US/docs/internetworking/technology/handbook/QoS.html>
Acesso em 07/03/2010
- http://www.cisco.com/en/US/docs/ios/11_1/feature/guide/DWFQ.html Acesso em 07/03/2010
- http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/fqos_c.html
Acesso em 07/03/2010
- http://en.wikipedia.org/wiki/Random_early_detection Acesso em 10/03/2010
- http://en.wikipedia.org/wiki/Priority_Queue Acesso em 10/03/2010

- http://www.rnp.br/newsgen/0005/qos_voip1.html#ng-classe Acesso em 08/03/2010

5.3 RFCs

- RFC 2309 Recommendations on Queue Management and Congestion Avoidance in the Internet
- RFC 2474 Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers
- RFC 2475 An Architecture for Differentiated Services
- RFC 2597 Assured Forwarding PHB Group
- RFC 2697 A Single Rate Three Color Marker
- RFC 3140 Per Hop Behavior Identification Codes
- RFC 3246 An Expedited Forwarding PHB (Per-Hop Behavior)
- RFC 3260 New Terminology and Clarifications for Diffserv