

### 3 ALGORITMOS DE ENFILEIRAMENTO

Uma das maneiras que os elementos de redes (tipicamente roteadores) possuem para controlar as sobrecargas de pacotes nos buffers desses dispositivos de rede é através do emprego de um **algoritmo de enfileiramento** [14] para ordenar este tráfego de pacotes e, então, determinar algum método de priorização de seu encaminhamento através de uma interface de saída (fila de saída). Na engenharia de rede, enfileiramento é a ação de armazenar pacotes ou células em um local onde eles permaneçam até serem processados. Normalmente, o enfileiramento ocorre dentro de roteadores quando os pacotes são recebidos pelo processador da interface de entrada do dispositivo, fila de entrada, e ocorre também antes da transmissão dos pacotes para uma outra interface (fila de saída) do mesmo dispositivo.

O gerenciamento de filas depende basicamente do algoritmo de enfileiramento dos pacotes e do tamanho máximo da fila. As escolhas do algoritmo e do tamanho máximo da fila podem, a primeira vista, ser relativamente simples, porém esta escolha pode ser extremamente difícil em função do padrão de comportamento do tráfego na rede ser aparentemente aleatório. Caso atribua-se um tamanho de fila elevado, poder-se-ia introduzir um grande atraso e grande jitter nos pacotes das aplicações, e conseqüentemente interromper temporariamente essas aplicações e os protocolos de transporte fim a fim. Por outro lado, se a fila é muito pequena, pode-se enfrentar o problema de tentar enviar dados para a rede de forma mais rápida do que ela pode aceitar, resultando em excesso de perdas de pacotes.

Nas subseções seguintes são descritas algumas técnicas para tratar o enfileiramento nas interfaces de saídas dos dispositivos de redes, pois estas são as estratégias predominantes para o padrão de tráfego de dados da rede IP e de certa forma auxilia no provimento de QoS para os serviços de diferentes classes. Cada um dos algoritmos foi projetado para resolver problemas específicos de tráfego na rede e desta forma tem um efeito particular no desempenho desta.

### 3.1. Enfileiramento FIRST IN FIRST OUT (FIFO)

O algoritmo de enfileiramento **FIFO** [14] pode ser considerado como o primeiro passo no controle de tráfego de dados e é atualmente o algoritmo de enfileiramento padrão da Internet. Este algoritmo de enfileiramento permite o armazenamento de pacotes de dados quando a rede está congestionada e o envio deles na ordem de chegada quando a rede não estiver mais sobrecarregada. Isto é, a medida em que os pacotes entram na fila da interface de entrada, eles são colocados na fila da interface de saída na mesma ordem que foram recebidos. A figura 7 abaixo ilustra o funcionamento do enfileiramento FIFO.

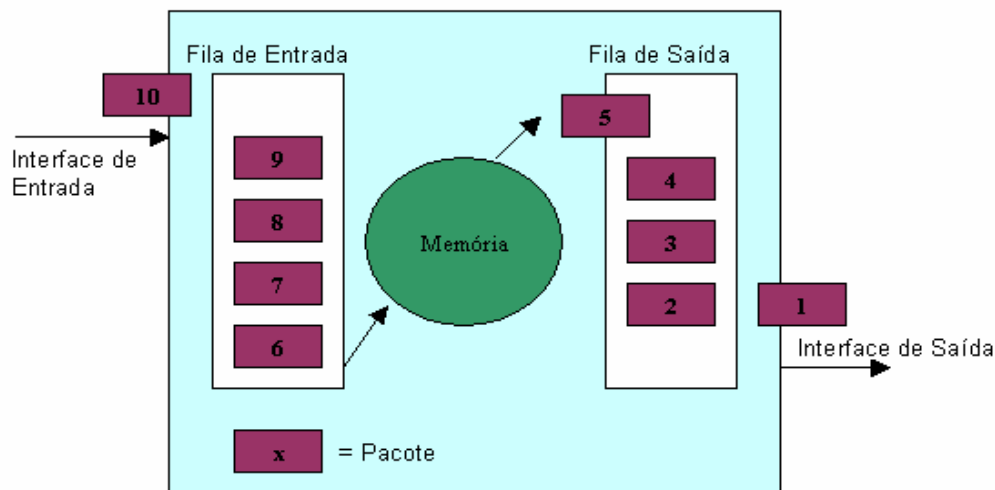


Figura 7 - Enfileiramento First In First Out.

Quando a rede IP opera sem sobrecarga, as filas são necessárias somente para assegurar que tráfegos de dados em rajadas de curta duração não causem descarte de pacotes. Nessa situação, o enfileiramento FIFO é altamente eficiente, pois na medida em que o tamanho da fila permanece pequeno, a média de retardo dos pacotes na fila é apenas uma fração ínfima do retardo de transmissão fim a fim. Entretanto, quando a carga da rede aumenta, o tráfego em rajada causa significativo atraso de enfileiramento em relação ao retardo de transmissão total e, quando a fila está totalmente cheia, todos os pacotes subsequentes são descartados. Se a fila opera deste modo por longo período, a qualidade do serviço inevitavelmente se degradará. Portanto, pode-se apontar como principal deficiência desse algoritmo de enfileiramento, o fato dele usar a ordem de chegada dos pacotes para determinar: a prioridade destes e a locação de espaço no buffer.

Do que foi exposto aqui fica evidente a necessidade de algoritmos de enfileiramento mais sofisticados para prover qualidade de serviço aos pacotes de dados que exigem principalmente baixa taxa de perda e baixo retardo de transmissão fim a fim.

### 3.2. Enfileiramento por prioridade

O algoritmo de enfileiramento por prioridade (**Priority Queuing - PQ**) [14] foi projetado para dar maior prioridade de enfileiramento aos tráfegos de dados que exigem certa urgência de processamento (sensíveis aos retardos de enfileiramentos). Este algoritmo é baseado no conceito que certos tipos de tráfegos podem ser identificados e colocados à frente na fila de saída e desta forma transmitidos na frente de outros tráfegos.

A forma como o tráfego de dados pode ser classificado na interface de rede de um dispositivo de rede é bastante flexível. Esta classificação [14] pode ser feita de acordo com o protocolo de rede (Internet Protocol, Internetwork Packet Exchange, Apple Talk) que está sendo utilizado no pacote de dados, no tamanho destes pacotes (quantidades de bytes), no endereço IP de origem e de destino e etc. Cada pacote pode ser classificado em um dos quatro níveis de prioridades, alto, médio, normal e baixo. A figura 8 abaixo mostra os pacotes de dados de quatro níveis de prioridades sendo recebidos e classificados na interface de entrada de um dispositivo de rede. Em seguida esses pacotes são enfileirados na fila de saída quanto aos seus níveis de prioridades.

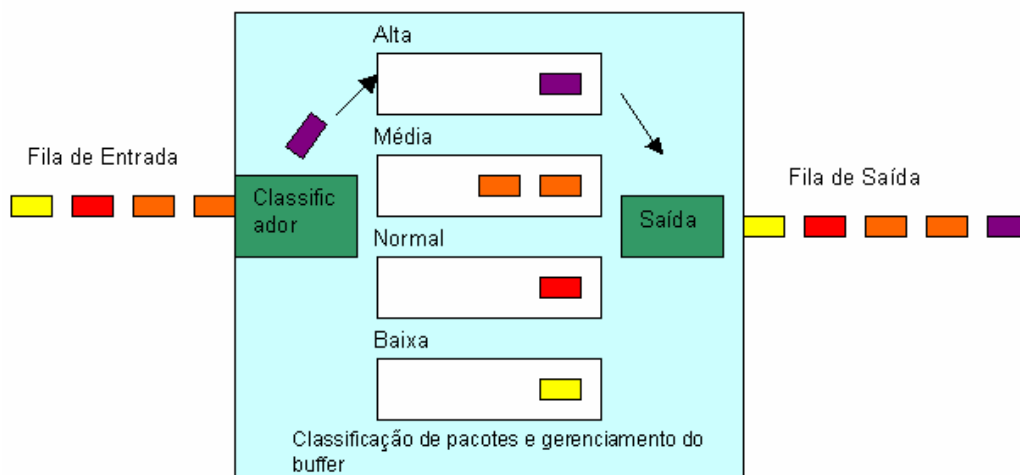


Figura 8 - Enfileiramento por prioridade.

Embora bastante flexível quanto aos parâmetros de classificação, este algoritmo de enfileiramento apresenta vulnerabilidade. Se o volume de tráfego de dados de alta prioridade for usualmente alto, o tráfego normal e de baixa prioridade podem esperar por um tempo maior no buffer para serem enfileirados na fila de saída. Isto pode levar a algumas aplicações a não funcionarem adequadamente por causa do retardo indesejável introduzido ou porque protocolos de roteamentos mais sensíveis ao retardo sofram time-out por não receberem confirmação dentro de um período predeterminado.

### 3.3. Enfileiramento baseado em classes

O algoritmo de enfileiramento baseado em classes (**Class Based Queuing - CBQ**) [14] foi projetado para permitir que várias aplicações, com especificações de largura de banda mínima ou de exigências de retardos compartilhem a rede. Este algoritmo é uma variação do enfileiramento por prioridades, pois o buffer pode ser dividido em várias filas com diferentes níveis de preferência (classes). Entretanto, o algoritmo de enfileiramento CBQ permite definir a quantidade de bytes de dados que deve ser escoada de cada fila do buffer para a fila da interface de saída do dispositivo de rede.

Utilizando esse algoritmo de enfileiramento é possível prover uma dada largura de banda fixa em um ponto de rede congestionado para um específico tráfego de dados e deixar a banda remanescente para o tráfego restante. Isto é feito determinando uma quantidade específica de bytes da fila do buffer para cada classe de pacotes e depois servindo a fila da interface de saída utilizando o algoritmo de escalonamento **Round Robin**. Este algoritmo de escalonamento permite que cada fila do buffer seja acessada numa seqüência pré-definida e uma dada quantidade de bytes seja extraída dessa fila. O resultado do acesso a essas filas do buffer numa seqüência pré-definida cria diferentes níveis de prioridades entre estas. Logo a escolha da fila do buffer a ser acessada basea-se na classe que esta fila pertence. A figura 9 ilustra o funcionamento desse algoritmo de enfileiramento.

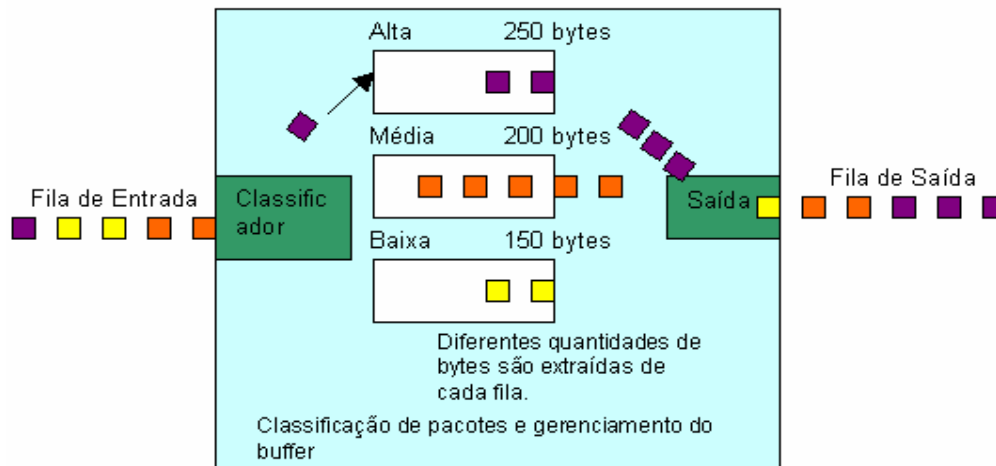


Figura 9 - Enfileiramento CBQ.

O algoritmo de enfileiramento CBQ tem sido considerado um método razoável para implementação de tecnologias que fornecem compartilhamento de canais de comunicação para classes de serviços. Entretanto, apresenta sobrecarga computacional na reordenação e no escalonamento de pacotes quando trabalha com canais de alta taxa de transmissão. Portanto, esse mecanismo é apropriado somente para canais de baixa taxa de transmissão, o que limita sua utilização.

### 3.4. Enfileiramento justo ponderado

O algoritmo de enfileiramento justo ponderado (**Weighted Fair Queuing - WFQ**) [14] permite que cada fila do buffer tenha uma mesma quantidade de bytes extraída e direcionada para a fila da interface de saída do dispositivo de rede. Isto torna mais justo o escalonamento desses pacotes de dados e, conseqüentemente, equidade no fornecimento dos diferentes serviços. Além disso, o WFQ assegura que as filas dos buffers não sofram por falta de largura de banda, pois a cada ciclo de execução desse algoritmo uma mesma quantidade de bytes é extraída de cada uma dessas filas.

O termo ponderado empregado no nome deste algoritmo provém do fato desse utilizar “pesos” para determinar a ordem de enfileiramento dos pacotes na transmissão. Para a execução deste ordenamento é preciso que cada pacote seja diferenciado quanto seu nível de prioridade. Portanto, o WFQ analisa o campo Prioridade (Precedence) do cabeçalho do pacote IP a fim de atribuir um determinado peso a esse pacote. Esse campo contém uma prioridade que varia de

0 (normal) a 7 (pacote de controle de rede). Quanto maior o nível (valor) de prioridade menor é o valor do peso atribuído a esse pacote. Assim, os pacotes de dados com baixos pesos são escalonados primeiro, ou seja, garantem ao fluxo desses pacotes um tempo de resposta mais imediato, tornando este algoritmo mais adaptativo às mudanças nas condições de tráfego da rede. O algoritmo WFQ é um mecanismo eficiente na medida em que esse pode utilizar toda largura de banda disponível para transmitir o tráfego de baixa prioridade se não existir tráfego de mais alta prioridade. A figura 10 abaixo ilustra o funcionamento do algoritmo WFQ.

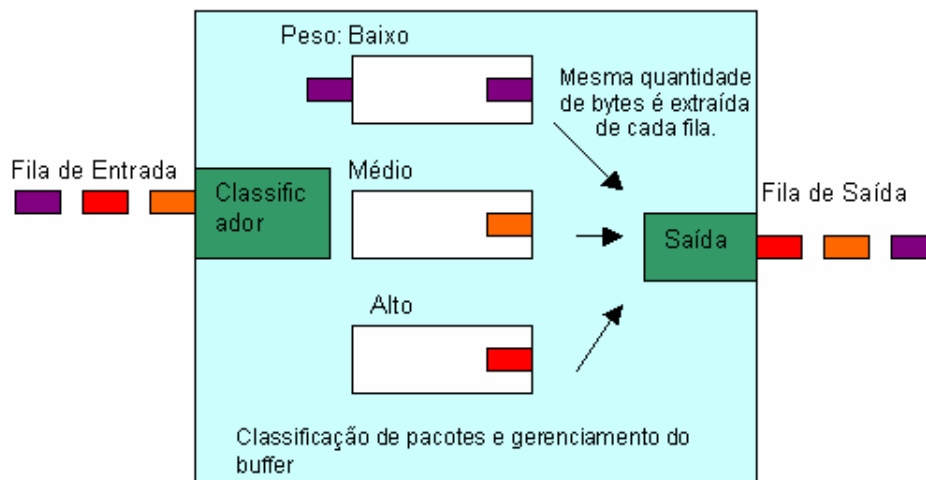


Figura 10 - Enfileiramento WFQ.

### 3.5. Algoritmo Random Early Detection (RED)

O princípio básico de funcionamento do mecanismo de controle de congestionamento de **Deteção Antecipada Aleatória (RED)** [15] consiste em não esperar até que o buffer de um Gateway ou Roteador fique cheio para detectar a sobrecarga de seu buffer, mas a detecção desse congestionamento é feita antes que essa sobrecarga ocorra.

O gerenciamento do buffer do roteador empregando o mecanismo de controle de congestionamento RED busca atingir os seguintes objetivos:

1. Acomodar pequenas rajadas de pacotes de dados que podem ser sensíveis aos retardos. Entretanto não permite que o tamanho médio da fila do buffer do roteador cresça em demasia. Para isso é usado um filtro passa baixa que monitora periodicamente o tamanho médio dessa fila. O

objetivo do monitoramento dessa fila é detectar de forma antecipada a ocorrência de possíveis congestionamentos no buffer deste dispositivo de rede.

2. Evitar a sincronização na rede: no tipo de fila **FIFO**, várias conexões podem receber ao mesmo tempo sinais que indicam a ocorrência de congestionamento na rede. Tal sinalização pode conduzir a ocorrência de oscilações nas taxas de transmissão de dados da rede, pois parte da rede que recebeu esse sinal tentará reduzir sua taxa de transmissão na tentativa de se adequar a uma nova taxa de transmissão, gerando assim oscilações na taxa de transmissão da rede.
3. O controle do tamanho médio da fila do buffer conduz ao controle do retardo de enfileiramento dos pacotes e conseqüentemente da taxa de transmissão desses pelo roteador.

Esses objetivos são alcançados através do monitoramento do tamanho médio da fila ( $avg$ ) do buffer do roteador que emprega o RED como mecanismo de gerenciamento do buffer. O valor do tamanho médio dessa fila é então comparado com dois limiares, o limiar do número mínimo de pacotes ( $min_{th}$ ) e o limiar do número máximo de pacotes ( $max_{th}$ ) no buffer desse roteador. Quando o tamanho médio dessa fila é menor do que o limiar do número mínimo de pacotes, nenhum pacote será descartado ou marcado com um bit no seu cabeçalho. Quando o tamanho médio da fila é maior do que o limiar do número máximo de pacotes todos, os pacotes que chegam no instante em que esse tamanho médio foi determinado serão descartados ou marcados.

Se o roteador que emprega o RED como mecanismo de gerenciamento de buffer descarta os pacotes quando o tamanho médio de sua fila atinge o limiar do número máximo de pacotes, este roteador garante que o tamanho médio dessa fila nunca exceda o limiar do número máximo de pacotes. Caso esse roteador ao invés de descartar os pacotes atribua um bit no cabeçalho desse pacote, então todos os dispositivos geradores de pacotes devem cooperar com este roteador para que o tamanho médio da fila não exceda significativamente o limiar do número máximo de pacotes. Quando o tamanho médio da fila do buffer está entre os dois limiares, a probabilidade ( $p_a$ ) com que um pacote que acaba de chegar no buffer venha ser marcado ou descartado é uma função do tamanho médio dessa fila. Portanto, o

mecanismo RED possui dois algoritmos. Um algoritmo para calcular o tamanho médio da fila do buffer para determinar o grau de rajada que irá ser permitida na fila do buffer desse roteador que utiliza esse mecanismo de gerenciamento de buffer. E um outro algoritmo para calcular a probabilidade ( $p_a$ ) com que os pacotes serão marcados ou descartados dado um determinado nível de congestionamento.

### 3.5.1. Detalhes do algoritmo RED

O cálculo do tamanho médio da fila do buffer do roteador que emprega o RED é determinado usando um filtro passa baixa, que nada mais é do que a função **Exponential Weighted Moving Average (EWMA)**.

$$avg_{(t)} = (1 - w_q) avg_{(t-1)} + w_q q \quad (1)$$

onde:

$avg$  é o tamanho médio da fila do buffer.

$w_q$  é o peso.

$q$  é o valor atual da fila (em pacotes).

O parâmetro peso ( $w_q$ ) é a constante de tempo do filtro passa baixa. Caso o valor para esse parâmetro seja muito baixo, o cálculo do valor do tamanho médio da fila irá refletir o valor atual da fila ( $q$ ) com um retardo muito grande. Ou seja, o tamanho médio da fila não refletirá as mudanças repentinas no tamanho atual da fila. Na referência [15] os autores Sally Floyd e Van Jacobson indicam que o valor do peso ( $w_q$ ) deve ser maior ou igual a 0,001 para assegurar um adequado cálculo do tamanho médio da fila.

Durante o período em que a fila desse buffer está vazia (no período ocioso da rajada de dados) o algoritmo RED estima o número ( $m$ ) de pequenos pacotes que poderia ser transmitido pelo roteador durante esse período. Após o período ocioso o roteador calcula o tamanho médio da fila como se ( $m$ ) pacotes tivessem chegado a essa fila durante o período ocioso.

Como o tamanho médio da fila ( $avg$ ) varia entre  $min_{th}$  e  $max_{th}$ , a probabilidade de descarte ou de marcação de pacotes varia linearmente entre 0 ao maior valor dessa probabilidade ( $max_p$ ). Portanto, a probabilidade de descarte ou



de marcação de pacotes quando o tamanho médio da fila ( $avg$ ) está entre os limiares  $min_{th}$  e  $max_{th}$  é dada por:

$$P_a = \max_p \frac{(avg - min_{th})}{(max_{th} - min_{th})} \quad (2)$$

Para evitar a sincronização global da rede é necessário que a diferença entre  $max_{th}$  e  $min_{th}$  seja grande o suficiente para que o tamanho médio da fila ( $avg$ ) não oscile freqüentemente até  $max_{th}$ . Caso essa diferença seja muito pequena o tamanho médio da fila irá variar até  $max_{th}$  em uma freqüência maior, logo com maior freqüência todos os pacotes serão descartado com probabilidade máxima de descarte ( $max_p$ ). Por recomendação que pode ser vista na referência [15] a relação entre estes dois limiares é que o  $max_{th}$  seja no mínimo duas vezes o  $min_{th}$ .

De forma bem simplificada é ilustrado o funcionamento do algoritmo RED no pequeno script abaixo.

```

for (cada novo pacote que chega a fila) {
  Calcule o tamanho médio da fila (avg);
  if ( $min_{th} \leq avg < max_{th}$ ) {
    Calcule a probabilidade  $p_a$ ;
    Com probabilidade  $p_a$  marque ou descarte o pacote;
  }
  else if ( $avg \leq max_{th}$ ) {
    Marque ou descarte o pacote;
  }
}

```