

4 Serviços Integrados e Diferenciados

As diversas aplicações utilizadas na Internet possuem diferentes requisitos de QoS. Para permitir que a pilha de protocolos TCP/IP tenha a capacidade de atender as diferentes necessidades destas aplicações, os serviços integrados [1] e serviços diferenciados [2] surgiram como uma proposta apresentada pelo IETF.

O IntServ utiliza o protocolo de sinalização fim-a-fim RSVP (Resource ReSerVation Protocol) [4], e mecanismos de alocação de largura de faixa nos roteadores que se encontram ao longo do caminho percorrido por um determinado tráfego. A integração desses dois mecanismos possibilita a utilização de serviços orientados à conexão na Internet, e a qualidade de serviço para um determinado tráfego é garantida. Entretanto esta arquitetura apresenta problemas de escalabilidade.

Com o objetivo de resolver os problemas de escalabilidade existentes no IntServ, foi proposto no IETF a arquitetura DiffServ. O mecanismo básico de funcionamento do DiffServ é o agrupamento de fluxos provenientes das fontes interligadas aos roteadores de borda (roteadores de entrada de um domínio DiffServ) em classes de serviço pré-determinadas. Nesta arquitetura os pacotes com necessidades de tráfego semelhantes são agrupados através de um byte já existente no cabeçalho de cada pacote IP, o campo ToS (*Type of Service*) na versão 4 do protocolo IP e Classe de Tráfego na versão 6.0 do IP, denominado DSCP (*Differentiated Services Code Point*) [3] na arquitetura DiffServ. Esse campo possibilita a marcação de pacote em níveis de prioridade ou níveis de serviços diferenciados de acordo com os requisitos do tráfego. Isso possibilita minimizar o processamento gerado nos roteadores da rede, pois as informações são tratadas como um agregado de fluxos e não como um fluxo individual, o que ocorre na rede IntServ.

A seção a seguir apresenta um detalhamento das duas arquiteturas propostas pelo IETF. Inicialmente, são apresentados os principais componentes e seus princípios de funcionamento. Posteriormente, uma comparação é realizada entre as duas arquiteturas.

4.1. Serviços Integrados (IntServ)

A arquitetura Serviços Integrados foi desenvolvida no IETF entre os anos de 1995 -1997. Para que os fluxos possam ser tratados de forma diferenciada a arquitetura IntServ utiliza o protocolo de sinalização RSVP, para fazer a reserva dos recursos necessários para um determinado fluxo, e de controle de admissão para determinar quais fluxos poderão ser admitidos com a garantia solicitada. Em adição ao tradicional serviço de melhor esforço, a arquitetura IntServ propõe duas novas classes: serviço garantido e serviço de carga controlada. Essas classes são brevemente descritas a seguir.

- Serviço Garantido [5]: tem como objetivo garantir que os pacotes chegarão ao destinatário com tempo de entrega garantido. Para isso fornece limites rígidos (matematicamente prováveis) em relação a atrasos de enfileiramento. Além disso, oferece um nível assegurado de largura de faixa e uma proteção contra perda de pacotes nas filas, que obedecerem ao perfil de tráfego contratado. Este tipo de serviço é projetado para aplicações que necessitam receber garantias fixas de que os pacotes chegarão ao destino com um atraso máximo pré-determinado, como aplicações de áudio e vídeo em tempo real. Esse serviço não oferece garantia mínima da variação de atraso. Ele simplesmente garante um atraso máximo gerado pelas filas.
- Serviço de Carga Controlada [6]: tem como objetivo fornecer suporte a uma faixa larga de aplicações que são extremamente sensíveis a condições de congestionamento, mas não fornecem garantias quantitativas em relação ao desempenho. O comportamento fim-a-fim oferecido por este serviço é semelhante ao comportamento visto por aplicações que estão recebendo o serviço de “melhor esforço” em uma rede apenas “levemente” carregada. Um fluxo atendido pelo Serviço de Carga Controlada espera receber da rede um atraso de enfileiramento próximo a zero e perda de pacote devido a congestionamento também próximo a zero. Este tipo de serviço é apropriado para aplicações

em tempo real adaptativas como as que estão sendo atualmente desenvolvidas para Internet. O desempenho destas aplicações é sensivelmente degradado por condições de congestionamento.

4.1.1.

Controle de Admissão e Protocolo de Reserva de Recurso

Quando uma nova fonte solicitar um determinado nível de serviço para um fluxo de pacotes, o controle de admissão verifica as especificações do fluxo e determina se o serviço desejado pode ser atendido dada a situação atual dos recursos disponíveis. Se houver recursos disponíveis o fluxo é admitido, senão é negado.

O RSVP (*Resource ReserVation Protocol*) [4] é um protocolo desenvolvido para realizar reserva de recursos em uma rede de serviços integrados. Esse protocolo é utilizado pelos terminais para solicitar à rede níveis específicos de QoS para as aplicações. O protocolo RSVP é também utilizado por roteadores para entregar requisições de controle da Qualidade de Serviço para todos os nós ao longo do caminho por onde os dados irão fluir, além de manter o estado da conexão para o serviço que foi requisitado. As requisições RSVP geralmente terão como resultado a reserva de recursos realizada em todos os roteadores presentes ao longo da conexão.

O mecanismo de reserva do RSVP pode ser explicado considerando a situação onde uma fonte e um receptor estão tentando fazer uma reserva para o tráfego que irá fluir entre eles. Existem duas condições iniciais para que a reserva seja realizada. Primeiramente, a fonte irá especificar os requisitos de QoS que o novo fluxo necessita. Em segundo lugar, é necessário determinar o caminho exato que os pacotes irão seguir entre a fonte e o receptor, pois é o receptor que irá enviar um pacote de volta reservando os recursos. A figura 4.1 ilustra o princípio de funcionamento do processo de reserva.

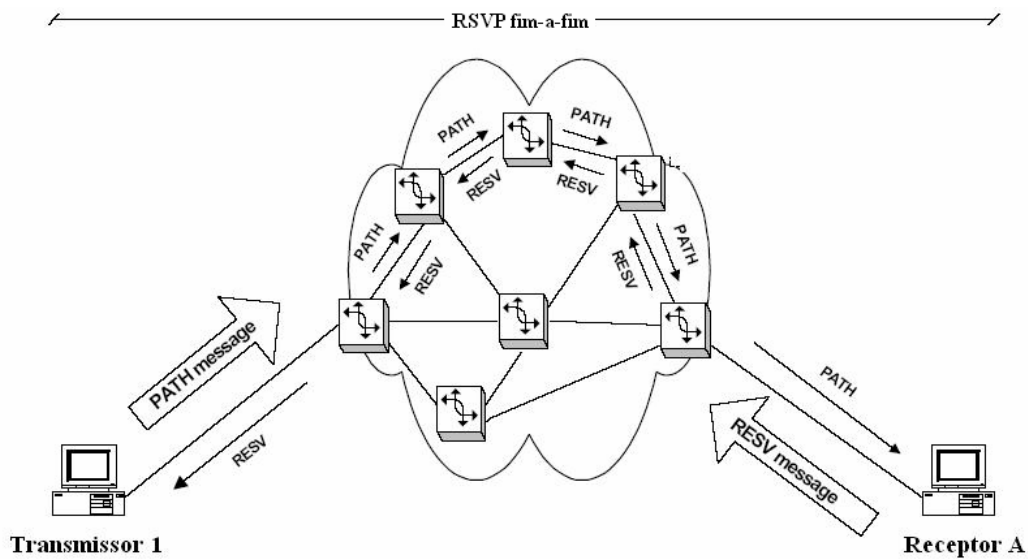


Figura 4.1: Mensagens PATH e RESV

No processo de reserva, ilustrado na figura 4.1, o transmissor envia uma mensagem PATH para o receptor, especificando as características do tráfego. Após receber a mensagem PATH, o receptor responde com uma mensagem RESV, descrevendo os recursos necessários para este receptor. Cada roteador ao longo do caminho verifica o pedido de reserva e tenta alocar os recursos necessários. Se o pedido de reserva puder ser atendido, a mensagem RESV é passada para o próximo roteador. Senão uma mensagem de erro é enviada para o receptor que fez o pedido e o processo de sinalização é finalizado. Se a requisição for aceita por todos os roteadores, a largura de faixa e o espaço necessário em *buffer* são alocados para o fluxo em cada roteador envolvido.

4.1.2. Classificação e agendamento de pacotes

Após realizar a reserva de recursos para transmissão do fluxo especificado pela fonte, os roteadores devem entregar os pacotes respeitando os requisitos acordados. Para que isso possa ser feito, duas tarefas importantes devem ser realizadas: a classificação e o agendamento dos pacotes. A classificação consiste em associar cada pacote com os fluxos reservados. O agendamento é o processo de gerência dos pacotes nas filas internas dos roteadores.

A arquitetura *IntServ/RSVP* representa uma mudança fundamental no modelo original da Internet. Entretanto, esta arquitetura é pouco escalável por

que cada roteador deve armazenar informações de estado dos fluxos individuais. A arquitetura de serviços diferenciados (*DiffServ*) ganhou destaque como proposta alternativa por endereçar o problema de Qualidade de Serviços para os protocolos TCP/IP e não apresentar esta limitação.

4.2. Serviços Diferenciados

A arquitetura DiffServ tem se apresentado como uma solução escalável que poderá fornecer suporte ao crescimento da Internet em relação a tráfego e a oferta de diferentes tipos de serviços. Em uma rede DiffServ os usuários e organizações interessados em receber um serviço melhor do que o de “melhor esforço”, devem negociar um Acordo de Serviço (Service Level Agreement–SLA) com o provedor de serviço Internet. Este negocia um SLA com outros provedores de serviço Internet. O SLA é um contrato de serviço que especifica o nível de serviço que um cliente da rede deve receber em termos de parâmetros de QoS. Este contrato também especifica o modo de condicionamento de um tráfego quando este entra em uma rede DiffServ.

A seção a seguir apresenta os elementos que compõem uma região capaz de prover serviços diferenciados.

4.2.1. Elementos de uma arquitetura DiffServ

Um domínio de Serviços Diferenciados é composto por um conjunto de roteadores contíguos compatíveis com a proposta de serviços diferenciados que compartilham uma mesma política de provisão de serviços. O conjunto de domínios DiffServ formam uma região DiffServ. A figura 4.2 mostra os elementos que compõe uma região DiffServ. Os roteadores de um domínio DiffServ podem ser divididos em duas categorias: roteadores de borda e roteadores de núcleo. Os roteadores de borda interconectam um domínio DiffServ a outros domínios ou a redes de acesso do usuário final, enquanto os roteadores de núcleo conectam-se a outros roteadores localizados no interior de um domínio DiffServ e aos roteadores de borda que pertencem ao mesmo domínio DiffServ. Os roteadores de borda podem ser classificados em roteadores de ingresso ou egresso, os roteadores de ingresso são responsáveis pelo condicionamento do tráfego que entra em um domínio DiffServ, e os roteadores de egresso são responsáveis por

Os roteadores de borda têm como função principal medir e controlar os fluxos individuais através da utilização de mecanismos para classificar, medir, marcar e moldar/descartar pacotes. Os roteadores de núcleo encaminham os pacotes de acordo com as especificações definidas nos roteadores de borda.

Nesta arquitetura os pacotes com necessidades de tráfego semelhantes são agrupados através de um byte já existente no cabeçalho de cada pacote IP, campo ToS na versão 4 do protocolo IP e Classe de Tráfego na versão 6.0 do IP. Denominado DSCP (*Differentiated Services Code Point*) [3] na arquitetura DiffServ. Esse campo possibilita a marcação de pacote em níveis de prioridade ou níveis de serviços diferenciados de acordo com a necessidade do tráfego. Isso possibilita minimizar o processamento gerado nos roteadores da rede, pois as informações são tratadas como um agregado de fluxos e não como um fluxo individual.

A figura 4.4 ilustra o uso deste campo, os bits de 0 a 5 representam o DSCP e os bits 6 e 7 não são utilizados. A figura 4.5 ilustra o cabeçalho IPv4 e IPv6.

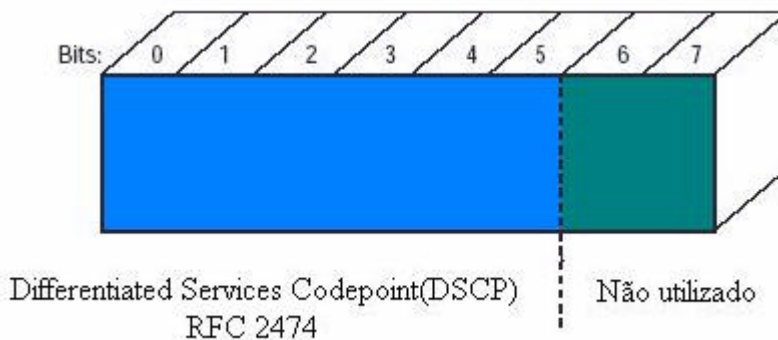


Figura 4.4: Campo DSCP

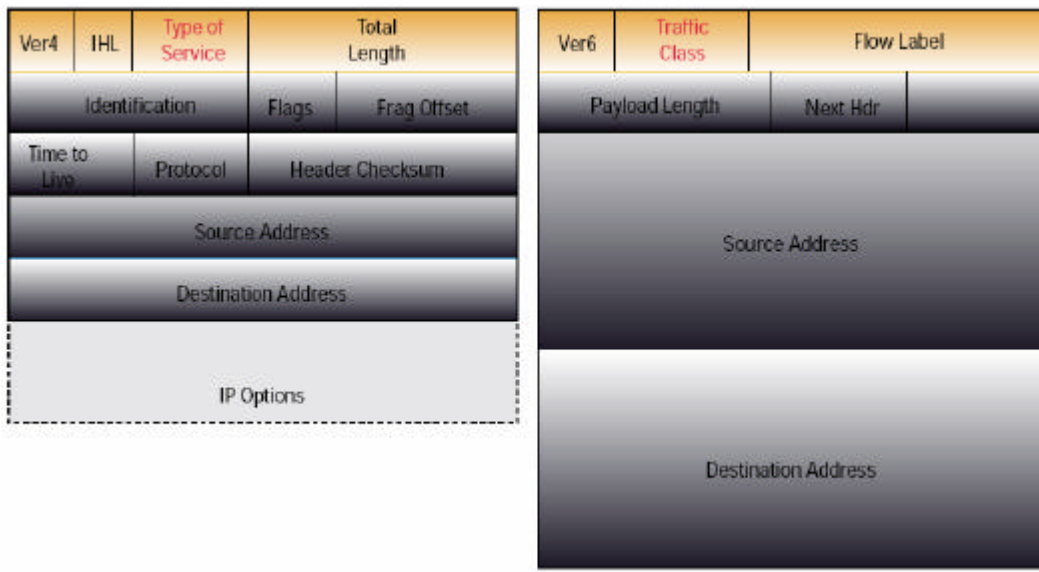


Figura 4.5: Cabeçalho IPv4 e IPv6

A marcação de pacotes pode ocorrer no próprio cliente ou nos roteadores de borda de uma rede DiffServ.

Diferentemente do protocolo RSVP, os pacotes são identificados pelos roteadores na chegada. Assumindo que os pacotes tenham sido marcados, os roteadores de núcleo são responsáveis pelo encaminhamento destes pacotes. Para definir como os pacotes serão tratados no núcleo da rede, o grupo de trabalho DiffServ do IETF padronizou um conjunto de comportamento dos roteadores para serem aplicados aos pacotes marcados. Estes comportamentos são conhecidos como “comportamento agregado por nó (*PHB – Per Hop Behavior*)”. O tratamento que cada pacote receberá vai depender do campo DSCP.

Atualmente, encontram-se padronizados dois PHBs: Encaminhamento Expresso (*Expedited Forwarding-EF*) [7] e o Encaminhamento Assegurado (*Assured Forwarding-AF*) [8]. O mais simples dos PHBs é o encaminhamento expresso (EF).

4.2.2. Encaminhamento Expresso

Os pacotes marcados para receberem um tratamento EF são encaminhados pelo roteador com valores mínimos de atraso e perda. Existem várias estratégias possíveis para implementação de um comportamento EF. Uma possibilidade é fornecer aos pacotes EF estrita prioridade sobre todos os outros pacotes. Outra possibilidade é usar o mecanismo de gerenciamento de

fila Weighted Fair Queue (WFQ) [44] entre os pacotes EF e os outros pacotes, com o peso dos pacotes EF suficientemente maior, para que todos os pacotes EF possam ser processados rapidamente. A vantagem do segundo método é permitir que pacotes não-EF possam conseguir algum acesso ao link, mesmo quando a quantidade de tráfego EF for excessiva. Este tipo de serviço pode ser visto pelos usuários como uma Linha Privada Virtual.

4.2.3. Encaminhamento Assegurado

O PHB AF é um mecanismo utilizado por um domínio DiffServ para fornecer diferentes níveis de garantia de encaminhamento dos pacotes IP provenientes de um cliente do domínio DiffServ (pode ser a própria fonte ou um roteador).

Em um PHB AF podem ser definidas 4 classes. Dentro de cada classe AF, um pacote IP pode ser marcado com três níveis diferentes de precedência de descarte. Em situações de congestionamento, a precedência de descarte de um pacote determina a sua importância relativa dentro da classe AF. Um nó DiffServ congestionado descarta preferencialmente pacotes com um valor de precedência de descarte maior e evita descartar pacotes com um valor de precedência de descarte menor. Cada classe aloca determinada quantidade de recursos, espaço no *buffer* e largura de faixa, para encaminhamento dos pacotes em cada nó DiffServ.

Em um nó DiffServ, as garantias de QoS para encaminhamento do pacote dependem de três fatores: (1) a quantidade de recursos alocados para a classe AF na qual o pacote se encontra, (2) a carga atual da classe AF e por último em caso de congestionamento com a classe (3) qual é o nível de prioridade de descarte do pacote.

O objetivo desse tipo de serviço é fornecer uma melhor qualidade de transmissão para um determinado tráfego quando existirem momentos de congestionamento. A implementação de um PHB AF procura minimizar congestionamentos de longa duração, porém permitindo congestionamentos de curta duração resultantes de rajadas de tráfego.

O PHB AF detecta e reage a congestionamentos de longa duração dentro de cada classe através do descarte de pacotes. Além disso, pacotes que causam congestionamento de curta duração são aceitos na fila. O

gerenciamento das filas em um PHB AF é realizado através de algoritmos de gerenciamento ativo de filas.

4.2.4. Gerenciamento Ativo de Filas

A implementação de um PHB AF requer a existência de um mecanismo de gerenciamento ativo de filas que será capaz de minimizar congestionamentos longos, enquanto permite congestionamentos curtos para atender tráfego em rajada.

4.2.4.1. RED

O mecanismo mais comumente utilizado atualmente capaz de suportar o requerimento acima é o gerenciamento ativo de fila RED (*Random Early Detection*) [9]. Este algoritmo detecta congestionamentos incipientes através da estimativa do tamanho médio da fila a cada chegada de um novo pacote. O tamanho médio da fila é calculado utilizando um filtro passa-baixa. Dois parâmetros são utilizados para determinar a probabilidade de descarte do pacote recém chegado à fila: min_th e max_th . Enquanto o tamanho médio da fila permanecer abaixo de min_th , todos os pacotes que chegam ao roteador são aceitos na fila. Se este valor médio estiver entre min_th e max_th , os novos pacotes serão marcados para descarte com uma probabilidade p_b calculada dinamicamente, sendo que p_b é uma função do tamanho médio da fila avg_queue .

$$p_b = \frac{P_{max} * (avg_queue - min_th)}{max_th - min_th}$$

P_{max} é o valor máximo da probabilidade de marcação dentro do intervalo $[min_th, max_th]$. A probabilidade de descarte p_b aumenta com o tamanho médio da fila avg_queue e com o número de pacotes aceitos no *buffer* desde o descarte do último pacote. O resultado de uma probabilidade de descarte alta é a detecção e limitação do congestionamento através do descarte de pacotes de forma antecipada. Se avg_queue exceder max_th , todos os pacotes novos serão

marcados para descarte com probabilidade 1, até que a ocupação da fila seja reduzida a um valor menor do que max_th . Devido à probabilidade do roteador que implementa RED descartar um pacote ser proporcional à parcela de ocupação na fila do roteador, quanto mais pacotes um fluxo enviar, maior será a probabilidade de um de seus pacotes serem descartados. A probabilidade de descarte correspondente a max_th é determinada pelo parâmetro RED P_{max} . A probabilidade de descarte p_b no algoritmo RED é ilustrada na figura 4.6 em função dos parâmetros descritos acima.

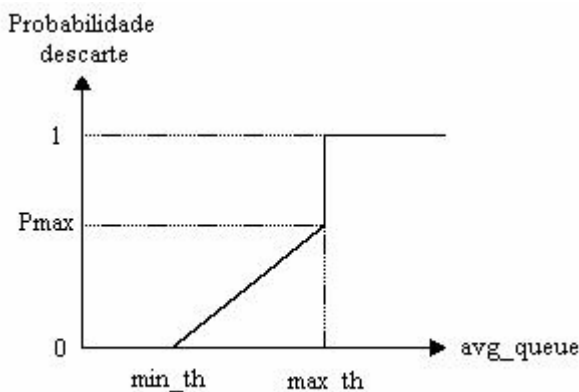


Figura 4.6: Parâmetros do algoritmo RED

Detalhes do algoritmo RED e de seus parâmetros podem ser encontrados em [9].

Uma generalização do algoritmo RED, o algoritmo RIO (*RED IN-OUT*) [10], tem sido muito utilizado como gerenciador ativo de fila.

4.2.4.2. RIO (RED IN-OUT)

Como mencionado anteriormente, para cada classe AF, os roteadores de borda de um domínio DiffServ monitoram e marcam pacotes de fluxos individuais ou de agregações de fluxos de acordo com os diferentes níveis de precedência de descarte suportados pelo domínio. O esquema RIO (RED IN-OUT) foi inicialmente proposto para fornecer diferenciação de serviço em dois níveis. Pacotes de um fluxo que estejam em conformidade com o perfil de serviço contratado são marcados como IN (*in profile*) e os pacotes que não estiverem em conformidade são marcados como OUT (*out of profile*). Essencialmente, são utilizados dois algoritmos RED: um é usado para controlar os pacotes IN, e o

outro para os pacotes OUT. O último é mais agressivo do que o primeiro, porque os pacotes fora do perfil devem ser os primeiros a serem descartados em caso de congestionamento, permitindo dessa forma um tratamento diferenciado entre as classes IN e OUT.

O mecanismo RIO contabiliza o tamanho médio da fila para pacotes IN (avg_queue_in) e também o tamanho médio da fila considerando-se tanto pacotes IN como pacotes OUT (avg_queue_total). A probabilidade de descarte de um pacote IN é função de avg_queue_in , enquanto a probabilidade de descarte de um pacote OUT é função de avg_queue_total . Como ilustrado na figura 4.7 existem três parâmetros para cada RED.

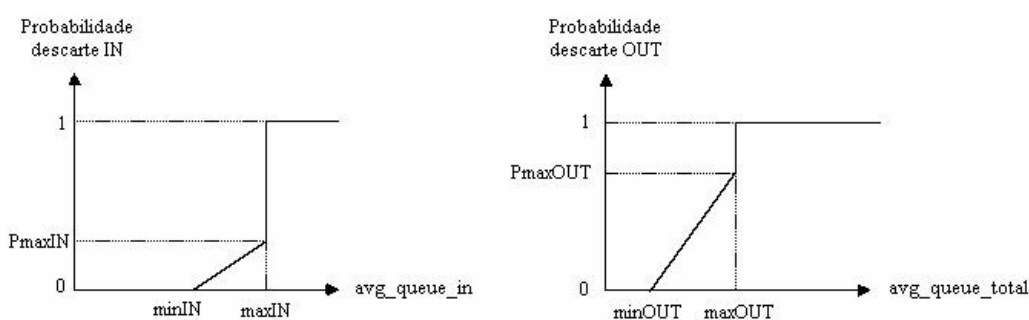


Figura 4.7: Parâmetros do algoritmo RIO

Embora cada RED que compõe o mecanismo RIO seja configurado com seus próprios parâmetros, o cálculo do tamanho médio da fila avg_queue para o RED que armazena os pacotes OUT é baseado na ocupação total da fila.

4.2.4.3. FRED

Lin e Morris [39] observaram que o algoritmo RED impõe as mesmas taxas de perdas aos diferentes fluxos, independentemente do espaço ocupado por cada um deles na fila. Esta política causa desigualdades no compartilhamento da largura de faixa por fluxos de diferentes características, devido aos seguintes fatores:

- Quando um novo pacote de uma determinada conexão é aceito na fila, observa-se um aumento do tamanho da fila e da probabilidade de descarte para todas as conexões. Portanto, um tráfego não responsivo

pode impor uma alta taxa de descarte para todas as conexões, prejudicando os fluxos adaptativos.

- O descarte aleatório pode fazer com que o RED elimine um pacote de um fluxo que possui poucos pacotes na fila.
- Conexões TCP com janelas maiores são mais tolerantes aos descartes porque são menos suscetíveis a *timeouts*. Como o algoritmo RED escolhe um pacote para ser descartado de forma aleatória e este pacote pode ser de uma conexão que possui janela pequena, tal conexão será muito mais penalizada do que seria uma outra conexão com uma janela maior.

A partir dessas conclusões Lin e Morris [39], propuseram um algoritmo denominado FRED (*Flow Random Early Drop*), onde a taxa de perda para cada fluxo é proporcional à sua ocupação na fila.

Resumidamente, o algoritmo FRED funciona basicamente da mesma forma que o RED, mas com as seguintes modificações: o FRED adiciona os seguintes parâmetros e variáveis: os parâmetros min_q e max_q , que são respectivamente, os números mínimo e máximo de pacotes que cada fluxo pode ter na fila; a variável global $avgcq$, que é uma estimativa da média de pacotes no *buffer* por fluxo; $qlen_i$ é uma variável mantido por fluxo que indica o número de pacotes na fila para determinado fluxo i ; e finalmente, $strike_i$ é uma variável por fluxo que indica o número de vezes que um fluxo i tenta ter mais do que max_q pacotes na fila.

O algoritmo FRED permite que cada conexão armazene min_q pacotes no *buffer* sem perdas. Um número de pacotes maior do que este valor está sujeito a ser descartado de maneira aleatória da mesma forma como acontece com o algoritmo RED. Com isso, o FRED protege fluxos considerados frágeis ou fluxos que possuem janelas pequenas, fazendo com que pacotes de conexões que possuem menos do que min_q pacotes armazenados no *buffer* sejam sempre aceitos enquanto o tamanho médio do *buffer* for menor do que max_{th} . O algoritmo FRED restringe os descartes aleatórios para os fluxos que possuem mais pacotes do que $avgcq$ ou min_q , o que for maior.

Para impedir que fluxos não-adaptativos mantenham o monopólio dos recursos da rede, o FRED não permite que um fluxo armazene mais do que max_q pacotes na fila, contabilizando em $strike_i$ as tentativas de um fluxo i exceder este limite. Finalmente, os fluxos com altos valores para $strike_i$ ficam proibidos de armazenar mais do que $avgcq$ pacotes na fila. Com isso, o FRED permite que

fluxos adaptativos enviem rajadas de tráfego, ao mesmo tempo que proíbe que fluxos não adaptativos monopolizem o espaço na fila.

Ao contrário do RED, o FRED calcula o tamanho médio da fila não apenas na chegada de um pacote como também na sua saída. A perda desta variação poderia levar a um erro de cálculo no tamanho médio da fila e acarretar uma baixa taxa de utilização da banda devido a descartes desnecessários.

Em suma, o algoritmo FRED garante uma maior justiça do que o algoritmo RED. Protege as conexões com janelas pequenas e consegue lidar com fluxos não adaptativos. O preço destas vantagens é uma maior complexidade e um custo computacional considerável.

4.3. Comparação entre IntServ e DiffServ

A arquitetura IntServ/RSVP proporciona uma melhora significativa em relação aos serviços de “melhor esforço” presentes na Internet, mas problemas de escalabilidade dificultam a sua implementação na Internet. Para solucionar os problemas existentes na arquitetura IntServ/RSVP foi proposta a arquitetura DiffServ. Os principais problemas relacionados a arquitetura IntServ são citados abaixo:

- Cada roteador deve armazenar informações de estado que crescem proporcionalmente ao número de fluxo. Quando um novo fluxo solicita reserva, informações de estado PATH e RESV são criadas em cada roteador presente ao longo do caminho, exigindo dos roteadores alta capacidade de armazenamento e processamento.
- Os roteadores devem classificar, policiar e enfileirar cada um destes fluxos. Além disso, decisões de controle de admissão devem ser realizadas quando um fluxo solicita reserva, o que pode causar uma carga excessiva nos roteadores.

Na arquitetura DiffServ podemos destacar as seguintes características:

- O conjunto de informações de estado é proporcional ao número de classes e não ao número de fluxos o que torna os Serviços Diferenciados mais escaláveis.
- As operações de classificação, marcação, policiamento e moldagem são realizadas nos roteadores de borda, onde os fluxos com características comuns são agregados. Com isso a complexidade está nos roteadores de borda, simplificando as tarefas dos roteadores de núcleo.

Devido as vantagens apresentadas pela arquitetura DiffServ, ela foi escolhida como base a ser utilizada neste trabalho, onde o objetivo é investigar o desempenho de vários mecanismos de marcação de pacotes propostos atualmente. A próxima seção apresenta em detalhes os componentes responsáveis pela classificação e pelo condicionamento de tráfego em uma rede DiffServ.