MARCAÇÃO DE TRÁFEGO PARA JUSTIÇA EM FLUXOS AGREGADOS NO SERVIÇO ASSEGURADO

por

Igor Briglia Habib de Almeida Alves

UFRJ

Tese submetida para a obtenção do título de
Mestre em Ciências em Engenharia de Sistemas e Computação
ao Programa de Pós-Graduação de Engenharia de Sistemas e Computação
da COPPE/UFRJ

MARCAÇÃO DE TRÁFEGO PARA JUSTIÇA EM FLUXOS AGREGADOS NO SERVIÇO ASSEGURADO

Igor Briglia Habib de Almeida Alves

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

Prof. Luis Felipe Magalhães de Moraes, Ph.D.

Prof. José Ferreira de Rezende, Dr.

Prof. Aloysio de Castro Pinto Pedroza, Dr.

Dr. Alexandre Leib Grojsgold, Ph.D.

RIO DE JANEIRO, RJ - BRASIL

OUTUBRO DE 2001

BRIGLIA HABIB DE A. ALVES, IGOR

Marcação de Tráfego para Justiça em Fluxos Agregados no Serviço Assegurado [Rio de Janeiro] 2001

XIV, 169 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 2001)

Tese - Universidade Federal do Rio de Janeiro, COPPE

1. Serviços Diferenciados
2. Justiça no Serviço Assegurado
3. Condicionadores de Tráfego

I. COPPE/UFRJ II. Título (Série)

Dedicatória

Em especial aos meu pais Carlos Olímpio e Solange, pelo incentivo e apoio dados para mais este passo na minha formação, bem como pelo esforço em viabilizar todas as etapas anteriores.

Com gratidão, ao ensino público gratuito de nível superior, que apesar da falta de incentivo e constantes cortes de verbas por parte do governo federal, consegue manter um nível de excelência difícil de ser alcançado pela iniciativa privada, a qual visa apenas o lucro.

Com amor, à minha namorada Luciene.

Com carinho, à minha sobrinha Luana, que veio alegrar a vida da minha família a partir do último ano.

In memorium, ao vovô Abílio Habib e ao tio Waldemar Rodrigues da Silva.

Agradecimentos

Ao professor José Ferreira de Rezende pela oportunidade oferecida de realizar um trabalho na minha área de interesse, pelo suporte no uso das ferramentas de simulação e pela companhia e incentivo nas apresentações em congressos.

Ao professor Luis Felipe Magalhães de Moraes, pelo suporte em termos de infra-estrutura de alta qualidade para o desenvolvimento da tese, pelo incentivo dado em desenvolver um trabalho de parceira entre dois programas da COPPE e por todas as oportunidades abertas *em prol* do meu desenvolvimento técnico e científico.

A ambos pela compreensão com relação a todas as dificuldades que passei e até criei durante o desenvolvimento deste trabalho.

Ao professor Aloysio de Castro Pinto Pedrosa e ao doutor Alexandre Leib Grojsgold, por terem aceitado o convite em participar da banca examinadora.

Aos meus pais, Carlos Olímpio e Solange, e a toda a minha família, em especial à tia Lene e à vovó Nalita pelo suporte na fase final da tese; e à minha namorada Luciene, que talvez tenha sido a única pessoa que torceu mais do que eu pelo sucesso deste trabalho.

Ao colega João Paulo Gonsiro Nacao, sempre disposto a ajudar no que fosse preciso e à colega Aline Carneiro Viana, pela revisão de parte da documentação.

Às secretárias do PESC, Cláudia e Solange, pela eficiência no atendimento e boa vontade para "quebrar vários galhos".

Aos colegas do PESC, do laboratório RAVEL e do GTA pela troca de conhecimentos, pela presença nas apresentações em congressos e seminários sobre o trabalho, e também pelos momentos de lazer que tornaram estes anos ainda mais agradáveis.

À CAPES, pelo apoio financeiro dado para a realização deste trabalho.

Finalmente, a Nelson Ramos Ribeiro e Paulo Cabral Filho, pelas dispensas das minhas obrigações de bolsista do LNCC em todas as ocasiões necessárias, cientes de que em determinados momentos é difícil conciliar duas grandes responsabilidades e objetivando apenas o meu enriquecimento profissional.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

MARCAÇÃO DE TRÁFEGO PARA JUSTIÇA EM FLUXOS AGREGADOS NO SERVIÇO ASSEGURADO

Igor Briglia Habib de Almeida Alves

Outubro/2001

Orientadores: Luis Felipe Magalhães de Moraes
José Ferreira de Rezende
Programa: Engenharia de Sistemas e Computação

Esta tese estuda o problema da falta de justiça no tratamento dado aos fluxos que compõem o tráfego assegurado quando estes compartilham recursos da rede de um provedor de Serviços Diferenciados (*Differentiated Services* - DiffServ).

Seu objetivo é propor um condicionador de tráfego eficiente em termos de justiça entre fluxos de um mesmo tráfego agregado, considerando o compartilhamento das larguras de faixa assegurada e excedente.

Como ponto de partida, uma solução denominada FM (*Fair Marker*) é apresentada. Estudos através de simulações são feitos de forma a entender o ajuste de seus parâmetros e avaliar o seu desempenho. De acordo com os resultados obtidos, o FM garante um alto grau de justiça no compartilhamento da largura de faixa assegurada, a depender do ajuste adequado de seus parâmetros. Porém, o FM se mostra incapaz de prover justiça no compartilhamento da largura de faixa excedente, especialmente quando o fluxo agregado é formado pela mistura de fontes de tráfego TCP e UDP.

Para suprir esta deficiência, este trabalho propõe a utilização de uma extensão do FM, denominada TCFM (Three Color Fair Marker), cujo desempenho também é avaliado através de simulações. Os resultados mostram que o TCFM consegue suprir as deficiências do FM em vários cenários. No entanto, devido à dinâmica do protocolo TCP, ambos os marcadores têm o seu desempenho degradado à medida que o número de fluxos aumenta.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

TRAFFIC MARKING FOR FAIRNESS IN AGGREGATED FLOWS IN THE ASSURED SERVICE

Igor Briglia Habib de Almeida Alves

October/2001

Advisors: Luis Felipe Magalhães de Moraes

José Ferreira de Rezende

Department: Computing and Systems Engineering

This thesis studies the problem of the lack of fairness in the treatment given to the flows that compose the assured traffic when these flows share resources of the network of a Differentiated Services (DiffServ) provider.

Its objective is to propose an efficient traffic conditioner in terms of fairness between flows of the same aggregated traffic, considering the sharing of the assured and excess bandwidths.

As a starting point, a proposed solution called FM (*Fair Marker*) is presented. Studies through simulations are made in order to understand the adjustment of its parameters and to evaluate its performance. According to the results, the FM guarantees a high degree of fairness in the assured bandwidth sharing, depending on the adequate smoothing of its parameters. However, the FM shows itself incapable to provide fairness in the excess bandwidth sharing, especially when the aggregated traffic is composed by the mixture of TCP and UDP flows.

To overcome this deficiency, this work proposes the use of an extension of FM, called TCFM (Three Color Fair Marker), whose performance is also evaluated through simulations. The results show that the TCFM overcomes the deficiencies of the FM in several scenarios. However, because of the TCP dynamics, both markers have their performance degraded as the number of flows increases.

Palavras-chave

Internet

Qualidade de Serviço

Serviços Diferenciados

Condicionador de Tráfego

Marcador de Tráfego

PHB-AF

Serviço Assegurado

Fluxo Agregado

Justiça

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 A Necessidade de Qualidade de Serviço na Internet

Quando se imagina o futuro mais imediato das telecomunicações, chega-se inevitavelmente a um cenário de uma grande infra-estrutura global e integrada de redes, capaz de transportar todo tipo de informação existente entre dois ou mais pontos quaisquer, geograficamente distribuídos. Para alcançar esta realidade, a solução escolhida deve acompanhar o aumento da quantidade de tráfego gerada pelo também crescente número de usuários, e ser capaz de lidar com as mais variadas características de tráfego.

Dentre as redes existentes hoje, a Internet surge como a maior candidata para ser o núcleo desta infra-estrutura por dois motivos principais [1]. Em primeiro lugar, já possui milhões de usuários conectados, entre pessoas, empresas e organizações. Esta alta capilaridade atrai empresas em função do enorme potencial para a venda de produtos e serviços. Algumas destas empresas dependem totalmente da Internet para sobreviver, como nos casos de provedores de acesso e lojas virtuais. Em segundo lugar, por ser uma rede baseada em comutação de pacotes, a Internet é consideravelmente flexível a mudanças visando adaptação aos requisitos específicos de aplicações existentes e emergentes.

No entanto, a Internet foi idealizada originalmente para transferência de arquivos entre computadores e acesso a estações remotas. Sendo assim, estes requisitos não muito restritivos levaram ao desenho de uma rede simples e caracterizada por um único nível de serviço, definido através das seguintes propriedades: não confiável, pois não há garantia ou confirmação de entrega de um pacote; não orientado

a conexão, já que cada pacote é tratado de forma independente e pode seguir um caminho distinto dos demais; e de melhor esforço (*best-effort*), na medida em que as perdas só ocorrem por esgotamento de recursos ou falhas em equipamentos, e não por iniciativa da própria rede [2]. Além disso, assume-se que todos os pacotes têm a mesma importância. Cada nó simplesmente armazena os pacotes na ordem de chegada e os encaminha baseando-se no endereço destino presente no cabeçalho do protocolo IP. Devido à ausência de qualquer mecanismo de diferenciação no tratamento dos pacotes, todos possuem a mesma chance de serem descartados. Embora este nível de serviço tenha sido suficiente para satisfazer às principais aplicações utilizadas inicialmente, este quadro muda a cada dia principalmente em função de dois fatores.

O primeiro fator é o crescimento da popularidade da Internet. Devido à sua expansão, principalmente no contexto comercial, surgem empresas interessadas em um nível de serviço superior ao de melhor esforço. Estas empresas visam expandir os seus mercados e se manterem competitivas, melhorando o nível de serviço oferecido no momento e (ou) viabilizando outros novos. Além disso, quando dispostas a pagar mais, esperam como retorno um melhor nível de serviço em relação aos demais usuários, demandando um tratamento diferenciado por parte da rede.

O segundo fator corresponde ao surgimento de novas aplicações. Com o grande desenvolvimento tecnológico na área de transmissão de dados através de fibras óticas, várias aplicações com requisitos específicos de desempenho se tornaram objetos de pesquisa e estão previstas para serem utilizadas na Internet, além de voz e vídeo já bastante difundidos. Tele-imersão, laboratórios virtuais, bibliotecas virtuais e ensino a distância são algumas das aplicações avançadas em desenvolvimento no projeto Internet 2, liderado pela UCAID (*University Corporation for Advanced Internet Development*). Estas aplicações demandam requisitos de desempenho por parte da rede que não podem ser satisfeitos pelo serviço de melhor esforço. Dependendo do tipo de aplicação, pode ser necessário impor limites de retardo, variação de retardo, largura de faixa e taxa de perda. Atualmente, a Internet não tem como oferecer tais garantias a nível global. Indo mais além, estas aplicações competiriam em igualdade de condições com aquelas que não necessitam destas garantias, levando ao uso ineficiente dos recursos disponíveis.

Portanto, vítima de seu próprio sucesso, a Internet encontra-se forçada a sofrer mudanças estruturais para que possa se firmar como solução global de telecomunicações. Isto significa transformá-la em uma rede com suporte a Qualidade de Serviço

Tabela 1.1: Aplicações versus sensibilidade aos parâmetros de QoS.

Tipo de tráfego	Vazão	Perdas	Latência	*Jitter*
Voz	Muito baixa	Média	Alta	Alta
Comércio eletrônico	Baixa	Alta	Alta	Baixa
Transações	Baixa	Alta	Alta	Baixa
Correio eletrônico	Baixa	Alta	Baixa	Baixa
Acesso remoto (Telnet)	Baixa	Alta	Média	Baixa
Navegação web casual	Baixa	Média	Média	Baixa
Navegação web crítica	Média	Alta	Alta	Baixa
Transferência de arquivos	Alta	Média	Baixa	Baixa
Videoconferência	Alta	Média	Alta	Alta
Multicast	Alta	Alta	Alta	Alta

(*Quality of Service* - QoS). FERGUSON e HUSTON [3] definiram QoS através de quatro parâmetros: o retardo (ou latência), tempo gasto pelos dados para ir da fonte ao destino; o *jitter* ou variação do retardo; a vazão, que corresponde à taxa efetiva de transferência de dados; e a integridade que diz respeito à capacidade da rede em entregar corretamente os dados. DUTTA-ROY [4] definiu QoS como um conjunto de métricas, destacando as quatro métricas anteriores juntamente com a disponibilidade, fração do tempo em que a rede está operando em condições satisfatórias. Na verdade, QoS pode ser definida de forma mais ampla como o agregado de todas as métricas de desempenho relevantes na definição do nível de serviço acordado entre cliente e provedor.

Diferentes aplicações requerem diferentes níveis de QoS, pois podem ser sensíveis a diferentes parâmetros, conforme ilustra a tabela 1.1 [4]. Por exemplo, uma longa transferência de arquivo precisa de alta largura de faixa e integridade, mas não é sensível ao retardo e à sua variação. Já uma videoconferência também precisa de alta largura de faixa, mas é altamente sensível ao retardo e à sua variação, além de admitir um certo nível de perdas. Sendo assim, prover QoS significa em termos práticos controlar estes parâmetros para cada aplicação, a fim de otimizar os recursos da rede e oferecer diferentes níveis de serviço.

Apesar de toda a argumentação apresentada, a necessidade de QoS na Internet não é um consenso geral. Alguns crêem que a técnica de Multiplexação por Divisão em Comprimento de Onda (*Wavelength Division Multiplexing* - WDM) levará a um

cenário de largura de faixa extremamente abundante [5]. Isto significaria ter QoS intrinsecamente fornecida pela rede, eliminando a necessidade de mecanismos adicionais para construí-la. Apesar de não ser possível prever o futuro, alguns argumentos fortes vão de encontro a esta linha de pensamento. Em primeiro lugar, novas aplicações tendem a ser criadas com requisitos de QoS mais restritivos e principalmente com maior demanda de tráfego gerado. Inclusive a própria fartura de largura de faixa serve como incentivo ao desenvolvimento destas aplicações. Em segundo lugar, o perfil do usuário segue a disponibilidade de recursos. Como exemplo podemos citar a evolução do volume médio das mensagens de correio eletrônico na Internet. Inicialmente, pequenas mensagens de texto. Atualmente, texto com imagens e vídeos anexados. Com o aumento da largura de faixa, por que não enviar filmes inteiros em uma mensagem? Em terceiro lugar, retardos continuarão sendo imprevisíveis devido à possibilidade de surtos de alta carga na rede. Para determinados usuários, isto pode ser intolerável. Finalmente, fartura de largura de faixa atualmente não é uma realidade e muito menos uma certeza, justificando por um bom tempo a adoção de mecanismo de QoS na Internet.

Sendo assim, ao longo dos últimos anos, o IETF (*Internet Engineering Task Force*) propôs vários modelos e mecanismos para suprir as demandas de QoS. As principais destas propostas serão descritas sucintamente a seguir.

1.2 Propostas para Obtenção de QoS na Internet

Dentre as propostas desenvolvidas pelo IETF para a implementação de QoS na Internet destacam-se: MPLS (*MultiProtocol Label Switching*) [6, 7], Roteamento Baseado em Restrições (*Constraint-Based Routing* - CBR) [8], Engenharia de Tráfego (*Traffic Engineering* - TE) [9], Serviços Integrados (*Integrated Services* - IntServ) [10] e Serviços Diferenciados (*Differentiated Services* - DiffServ) [11, 12]. Embora todas estas propostas venham a ser abordadas aqui em um mesmo nível, as três primeiras consistem mais em otimizadores de desempenho em redes, enquanto que as duas últimas definem mudanças estruturais na arquitetura da Internet.

1.2.1 MPLS

O principal objetivo do grupo de trabalho do IETF em MPLS (*MultiProtocol Label Switching*) é o de padronizar uma tecnologia que integre os paradigmas do encaminhamento por comutação de rótulos (*label switching*) com o do roteamento

na camada de rede. Esta tecnologia visa melhorar o desempenho do roteamento, torná-lo mais escalável e prover mais flexibilidade nos seus serviços, fazendo com que possam ser adicionados sem alterar a forma de encaminhamento [6, 7].

Cada pacote MPLS possui um cabeçalho. Na maioria dos casos, este cabeçalho é encapsulado entre as camada de enlace e de rede, contendo: um rótulo, um campo experimental conhecido como classe de serviço (*Class of Service* - CoS), um indicador de pilha de rótulos (*label stack*) e um campo TTL (*Time-To-Live*). Nos demais casos, o cabeçalho MPLS é codificado em cabeçalhos já existentes, como nos identificadores de caminho e circuito virtuais (VPI/VCI) em redes ATM (*Asynchronous Transfer Mode*).

O principal fundamento do paradigma MPLS é a separação do roteamento em duas funções. A primeira função reúne os pacotes que serão encaminhados da mesma maneira em grupos denominados Classes de Equivalência de Encaminhamento (*Forwarding Equivalence Classes* - FECs). A segunda função realiza a escolha de um valor para o rótulo MPLS de cada classe, determinando o roteamento.

Esta visão explora certas deficiências do roteamento na Internet, onde cabeçalhos IP são examinados em cada ponto de trânsito (multiplexador, roteador ou comutador)[1]. Isto consome tempo e contribui para o retardo total na medida em que o cabeçalho IP contém muito mais informação do que o necessário para determinar o próximo nó. Além disso, em cada roteador, tabelas de roteamento também são examinadas para a escolha do próximo passo em direção ao destino. Esta tarefa também pode ser considerada ineficiente quando se leva em conta a razoável estabilidade das tabelas de roteamento e a sobreposição de rotas de fluxos de tráfego distintos. Ambas fazem com que vários pacotes quase sempre obtenham o mesmo resultado no processo de roteamento. Portanto, no encaminhamento convencional do protocolo IP, um roteador considera dois pacotes como pertencentes à mesma FEC se existe algum prefixo nos endereços destino de ambos que provoca o mesmo resultado na busca do próximo nó na tabela de roteamento. Contudo, o processo se repete a cada nó atravessado, onde os pacotes são reexaminados e reatribuídos a uma FEC.

Em MPLS, a divisão do roteamento em duas funções distintas torna-o mais flexível, pois todos os pacotes pertencentes a uma mesma FEC podem receber o mesmo rótulo caso tenham as mesmas restrições. Alternativamente, rótulos distintos

[1]Embora esteja sendo abordado com enfoque no contexto da Internet, MPLS suporta vários protocolos da camada de rede, justificando o nome *MultiProtocol*.

podem ser usados para diferenciar fluxos de tráfego dentro de uma mesma classe de encaminhamento. Mas a principal vantagem é que um pacote é atribuído a uma FEC apenas uma vez, no ponto de entrada de um domínio MPLS (conjunto contínuo de nós que operam MPLS e são administrados segundo as mesmas políticas). Isto reduz o custo por nó em relação ao roteamento convencional. Cada roteador MPLS, denominado Roteador de Comutação por Rótulo (*Label-Switching Router* - LSR), examina apenas o rótulo (e possivelmente o campo CoS) para encaminhar um pacote.

A figura 1.1 ilustra o encaminhamento de um pacote através de um domínio MPLS. No nó MPLS de ingresso, o pacote é classificado e roteado com base no cabeçalho IP e em informações locais sobre roteamento mantidas no LSR. O cabeçalho MPLS é então inserido. Internamente, LSRs usam o rótulo como índice de busca em uma tabela de encaminhamento. Este processo é mais rápido do que a busca em tabelas de roteamento, pela simplicidade e facilidade de integração ao *hardware*. Em seguida, o rótulo de entrada é substituído pelo de saída e o pacote é comutado para o próximo LSR. Isto se repete até o nó MPLS de egresso, onde o cabeçalho é removido antes do pacote deixar o domínio MPLS. O processo é bastante similar à comutação VPI/VCI em redes ATM. A comutação hierárquica é permitida através do empilhamento de rótulos, facilitando a implementação de túneis.

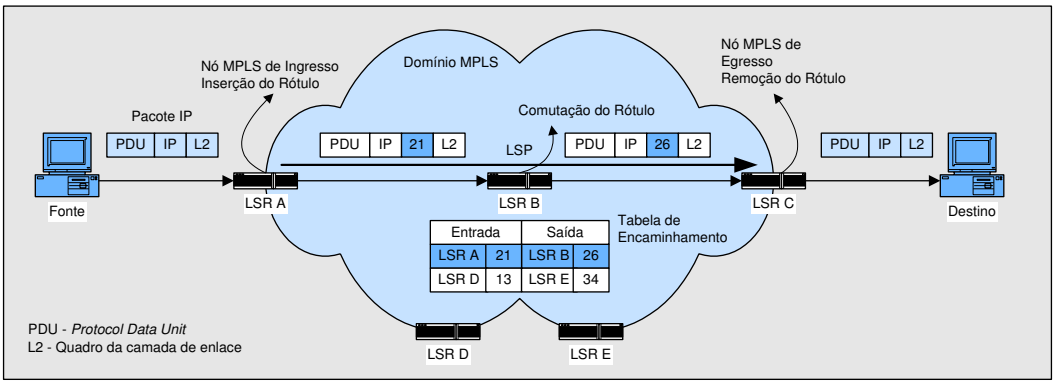


Figura 1.1: MPLS.

Os caminhos entre LSRs de ingresso e egresso são chamados de Caminhos Comutados por Rótulo (*Label-Switched Paths* - LSPs). A seleção das rotas pode ser feita nó-a-nó (*hop-by-hop*), onde cada roteador escolhe o próximo passo; ou de forma explícita, onde um único nó estabelece todo o caminho.

MPLS pode realizar a atribuição de rótulos de três maneiras. Na primeira, os rótulos são atribuídos previamente de acordo com as informações dos protocolos

de roteamento convencionais. Na segunda, informações de controle de protocolos de reserva de recursos disparam a criação dos caminhos e de rótulos para fluxos individuais ou agregados. Finalmente, rótulos podem ser atribuídos e distribuídos dinamicamente na chegada do tráfego. Neste último caso, o tempo de configuração pode ser significante para o retardo.

Portanto, MPLS provê potencial para a comutação de todo o tráfego que atravessa uma rede, onde o nível de granulosidade da atribuição dos rótulos é flexível e depende da escolha de uma das abordagens anteriores. Rótulos podem ser atribuídos por prefixos de endereços ou agrupamentos destes, podendo representar também rotas explícitas. Para níveis de capilaridade maiores, rótulos podem ser atribuídos por usuários, estações e até microfluxos[2].

MPLS necessita de um mecanismo para a distribuição dos rótulos a fim de construir LSPs. A arquitetura não assume nenhum protocolo em particular e existem várias abordagens que podem ser escolhidas em função dos requerimentos impostos para a criação dos LSPs. Se os caminhos se relacionam com as rotas convencionais, a distribuição dos rótulos pode ser embutida nos protocolos de roteamento [13]. Quando os caminhos estão reservados para pacotes de fluxos específicos, protocolos de reserva como o RSVP (*Resource ReserVation Protocol*) [14] executam esta tarefa. Além disso, novos protocolos como o LDP (*Label Distribution Protocol*) estão em desenvolvimento para distribuição genérica de rótulos [15] e suporte a rotas explícitas [16].

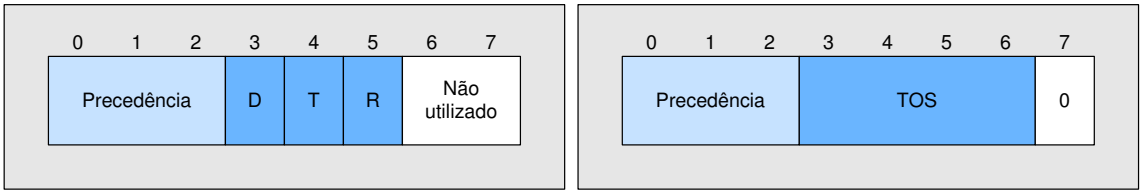
Em termos de suporte para QoS, o rótulo MPLS e opcionalmente o campo CoS permitem associar diferentes classes de serviço a diferentes LSPs. A separação fundamental entre classes de encaminhamento e atribuição de rótulos garante a flexibilidade para a provisão de níveis distintos de QoS em diferentes níveis de granulosidade. Conforme será visto na subseção 1.2.3, MPLS também provê suporte para engenharia de tráfego através da implementação de roteamento baseado em restrições. Finalmente, em redes de provedores de acesso à Internet (*Internet Service Providers* - ISPs), a diferenciação dos serviços implicará em cobranças como função do nível de QoS desfrutado pelo usuário. Nestes ambientes, MPLS pode ser muito útil na tarifação, pois sua arquitetura permite a atribuição de rótulos tanto a nível de usuário como por estação.

[2]Um microfluxo corresponde a um fluxo de pacotes entre aplicações, identificado pelo endereço fonte, endereço destino, porta fonte, porta destino e identificador do protocolo.

1.2.2 Roteamento Baseado em Restrições

Nos algoritmos que implementam o roteamento IP convencional, o resultado é função apenas do endereço destino e do conteúdo da tabela de roteamento. A partir destes dois parâmetros, o enlace de saída escolhido é aquele que corresponde ao menor custo de acordo com as métricas utilizadas. Como o endereço destino é fixo, a rota para um determinado fluxo de tráfego definido por um par de endereços fonte e destino será sempre a mesma, a não ser que ocorram mudanças na tabela de roteamento. Estas alterações ocorrem quando a topologia da rede muda ou em função de falhas. Flutuações de carga também podem influenciar quando o protocolo de roteamento utiliza métricas variáveis no tempo, tais como o retardo ou tamanho de filas. Outra possibilidade é a técnica de balanceamento de carga presente em alguns protocolos, a qual compartilha o tráfego entre rotas de mesmo custo. Por conseguinte, a escolha da rota não está relacionada a nenhum requisito de QoS.

Uma primeira possibilidade no sentido de utilizar outras variáveis no roteamento seria através do uso do octeto TOS (*Type Of Service*) do cabeçalho do protocolo IP (figura 1.2a). O octeto se inicia com três bits de precedência que determinam a importância relativa do pacote. Em seguida os bits D, T e R especificam o tipo de transporte que o datagrama deseja. O bit D significa baixo retardo (*Delay*), o bit T alta vazão (*Throughput*) e o bit R alta confiabilidade (*Reliability*). Os dois bits restantes não são utilizados. Porém, esta proposta não é implementada de forma abrangente nos roteadores. Além disso, embora os bits de precedência possam prover diferentes níveis de prioridade (ainda que de forma simples), os bits D, T e R têm sua função bastante limitada. Isto ocorre porque o nível de serviço de melhor esforço não pode garantir nenhum dos parâmetros solicitados. Logo, o uso destes campos serviria no máximo como dicas para protocolos de roteamento na escolha das métricas a serem usadas para cada datagrama [2].



(a) Cabeçalho do protocolo IP. (b) Proposta de roteamento TOS.

Figura 1.2: Campos do octeto TOS.

Outra proposta, conhecida como Roteamento TOS (*TOS Routing*) [17], redefine o octeto TOS conforme mostra a figura 1.2b. O campo de precedência mantém o mesmo propósito. Em seguida um campo de quatro bits, também chamado de TOS, indica os compromissos entre atraso, vazão, confibialidade e custo financeiro. O bit menos significativo não é utilizado e deve ser zero.

Na sua especificação, o roteamento TOS considera que apenas uma diretiva pode ser requisitada. A tabela 1.2 contém os valores possíveis e seus significados. Além disso, não há obrigatoriedade de que os roteadores sejam sensíveis aos valores destes campos. A única exigência é de que pelos menos o tratamento convencional seja aplicado. Esta característica e as semelhanças com o caso anterior inibem a aplicação deste modelo como forma de provisão de QoS.

Tabela 1.2: Campo TOS.

Valor	Significado
1000	Minimizar atraso
0100	Maximizar vazão
0010	Maximizar confiabilidade
0001	Minimizar custo financeiro
0000	Serviço normal

Um passo mais largo nesta direção foram as propostas de Roteamento Baseado em Qos (*QoS-Based Routing*) [18, 19]. Este tipo de roteamento permite, dinamicamente, a determinação de uma rota que tenha uma boa chance de acomodar o nível de QoS requisitado. Para isto, a seleção de uma rota é baseada em uma ou mais métricas para cada enlace. Métricas comuns são custo, número de nós, largura de faixa, confiabilidade, atraso e *jitter*.

Opcionalmente, apesar de não incluir um mecanismo para reserva de recursos, um protocolo de reserva como o RSVP pode ser usado para disparar cálculos de roteamento baseado em QoS para que as necessidades de um determinado fluxo sejam supridas. Desta forma, pacotes pertencentes a um mesmo par de endereços origem e destino podem ser encaminhados por rotas distintas de acordo com os requisitos de QoS de cada fluxo.

Outra evolução mais recente nesta área é o Roteamento Baseado em Restrições (*Constraint-Based Routing* - CBR) [8]. Este termo tem sido usado no contexto de MPLS como alternativa para a construção de LSPs com requerimentos de recursos

e desempenho, onde se leva em conta a disponibilidade de recursos da rede e (ou) políticas administrativas no cálculos das rotas. Porém, o conceito de roteamento baseado em restrições é ainda mais abrangente, incluindo roteamento para fluxos IP de diversos níveis de granulosidade, fluxos agregados ou circuitos virtuais, sujeitos a requerimentos de QoS. Além disso, dependendo da aplicação, um esquema de CBR pode levar em conta mudanças relativamente mais lentas ou mais frequentes no estado da rede. Por todas estas características, o roteamento baseado em restrições é uma generalização do conceito de roteamento baseado em QoS.

Portanto, ao determinar uma rota, podem ser considerados além da topologia da rede, requerimentos de um fluxo de tráfego específico, disponibilidade de recursos nos enlaces, e possivelmente outras políticas definidas por administradores de rede. Estas políticas podem por exemplo restringir o uso de determinados enlaces a pacotes oriundos de um conjunto limitado de endereços. Outra opção é a escolha de rota mais longas e com menos carga em detrimento ao caminho mais curto determinado pelos algoritmos de roteamento. A figura 1.3 exemplifica esta possibilidade quando o caminho A-B-C é escolhido (apesar do caminho A-C ser mais curto) por poder oferecer uma largura de faixa de 40 Mbit/s requisitada por um fluxo de tráfego.

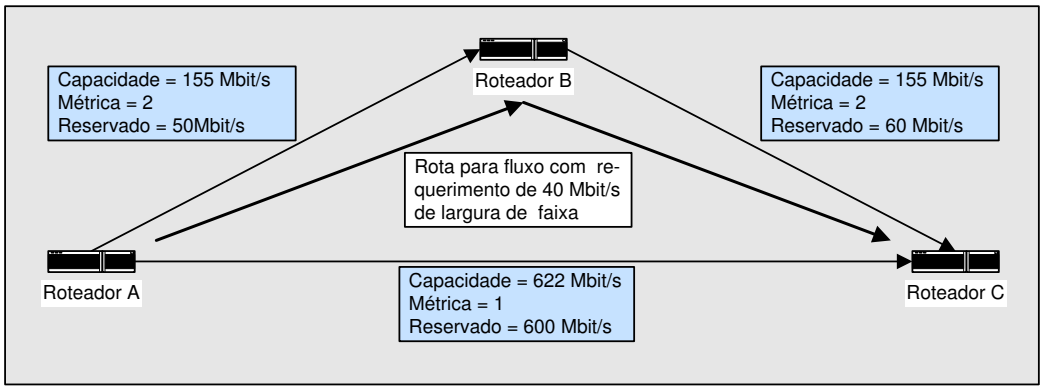


Figura 1.3: Roteamento baseado em restrições.

Essencialmente, o roteamento baseado em restrições cria um nível de decisão superior aos esquemas tradicionais de roteamento. Logo, difere de esquemas de roteamento baseado em QoS, os quais correspondem à extensões de protocolos específicos de estado dos enlaces (*link state*) [20, 21]. A adição de um nível de decisão superior permite que mudanças sejam feitas independentemente do esquema de roteamento utilizado. Além disso, a definição de uma boa interface entre os dois níveis elimina a necessidade do desenvolvimento de aspectos específicos de QoS para cada

algoritmo de roteamento a ser adotado. Finalmente, o roteamento baseado em restrições serve de suporte para a implementação de engenharia de tráfego conforme será visto na subseção 1.2.3 a seguir.

1.2.3 Engenharia de Tráfego

A Engenharia de Tráfego corresponde ao controle de como os fluxos de tráfego atravessam uma rede a fim de otimizar a utilização dos recursos e evitar congestionamentos, melhorando o desempenho da rede [9]. Congestionamentos podem ser causados pelo esgotamento de recursos ou por distribuição ineficiente de tráfego. No primeiro caso, roteadores e enlaces ficam sobrecarregados e a solução passa por prover mais recursos e substituir equipamentos. No segundo caso, algumas partes da rede estão sobrecarregadas enquanto outras estão subutilizadas. Esta distribuição desigual de tráfego é causada pelos protocolos de roteamento dinâmicos utilizados atualmente, tais como RIP (*Routing Information Protocol*), OSPF (*Open Shortest Path First*) e IS-IS (*Intermediate System-to-Intermediate System*). Protocolos IGP (*Interior Gateway Protocol*) como estes utilizam o caminho mais curto para encaminhar pacotes. Por conseguinte, roteadores e enlaces ao longo do menor caminho entre dois nós podem apresentar congestionamentos quando as suas capacidades são excedidas. O fato de caminhos mais curtos de diferentes fluxos de tráfego poderem se sobrepor em alguns trechos também contribui para isso. Como resultado, o caminho mais curto entre dois pontos pode ficar congestionado enquanto caminhos alternativos mais longos estão ociosos.

Estes problemas são enfrentados atualmente por ISPs, motivando o uso da engenharia de tráfego para otimizar recursos, aumentar o desempenho de suas redes e aumentar a renda sem a necessidade de grandes investimentos na infra-estrutura. Porém, a engenharia de tráfego fica difícil de ser implementada através de protocolos IGP por vários motivos. Em primeiro lugar, mesmo com alguns protocolos incorporando a opção ECMP (*Equal-Cost MultiPath*) [22], a qual permite dividir a carga igualmente entre vários caminhos mais curtos, a fração para cada rota não pode ser alterada. Portanto, alguns caminhos podem terminar carregando muito mais tráfego que os demais porque também suportam tráfego de outras fontes. Em segundo lugar, não há balanceamento de carga entre caminhos de custos diferentes em protocolos IGP. Em terceiro lugar, a modificação de métricas em um protocolo IGP para desviar o tráfego convenientemente tende a produzir efeitos colaterais, pois desvios indesejáveis também podem ocorrer. Finalmente, para redes simples talvez

Tabela 1.3: Atributos dos LSPs.

Nome do atributo	Significado
Largura de faixa	Valor mínimo para criação do LSP
Atributo do caminho	Determina se a rota é especificada manualmente ou dinamicamente
Prioridade de configuração	Decide qual LSP fica com um recurso disputado por vários LSPs
Prioridade de preempção	Decide quando um LSP já criado pode ceder um recurso para um novo LSP
Afinidade (cor)	Propriedade administrativa
Capacidade de adaptação	Controla a mudança para rotas mais adequadas
Confiabilidade	Decide por mudança de rota em caso de falhas

seja possível para os administradores implementar alguma engenharia de tráfego configurando manualmente os custos dos enlaces, mas para redes maiores e mais complexas outros mecanismos acabam sendo necessários.

Por estas razões, o uso de técnicas que suportem a automatização da engenharia de tráfego são estimuladas. Entre estas técnicas estão MPLS [23, 24], o roteamento baseado em restrições [16] e o uso de IGPs de estado de enlace (*link state*) com modificações apropriadas [25].

MPLS provê suporte à engenharia de tráfego em conjunto com o roteamento baseado em restrições. Contudo, para que o roteamento compute os LSPs sujeitos às restrições, um protocolo IGP de estado de enlace modificado deve ser usado para propagar os atributos dos enlaces, além das informações normais inerentes ao roteamento convencional. A tabela 1.3 lista atributos que podem ser associados a um LSP, de forma a controlar eficientemente os enlaces que o compõem [23].

Com a utilização destas três técnicas em conjunto (figura 1.4), dois dos problemas discutidos anteriormente são resolvidos através da engenharia de tráfego. O esgotamento de recursos é solucionado fixando a máxima largura de faixa reservável para cada enlace como a sua própria capacidade e configurando os requerimentos de largura de faixa para cada LSP. Deste modo, o roteamento baseado em restrições evita colocar mais LSPs do que o suportável em cada enlace. A figura 1.5 mostra um exemplo onde o CBR automaticamente coloca o LSP B-E num caminho mais longo para evitar um congestionamento no enlace C-E.

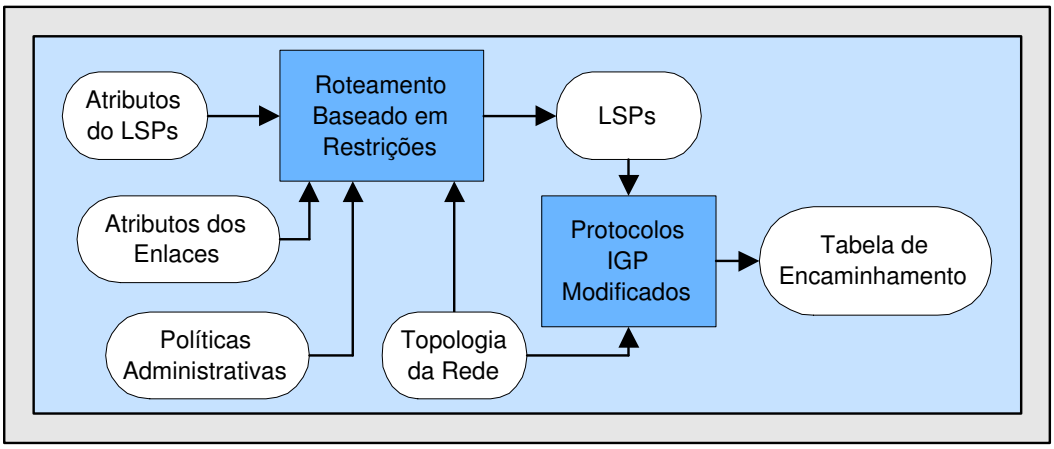


Figura 1.4: Engenharia de tráfego com MPLS, CBR e IGP modificados.

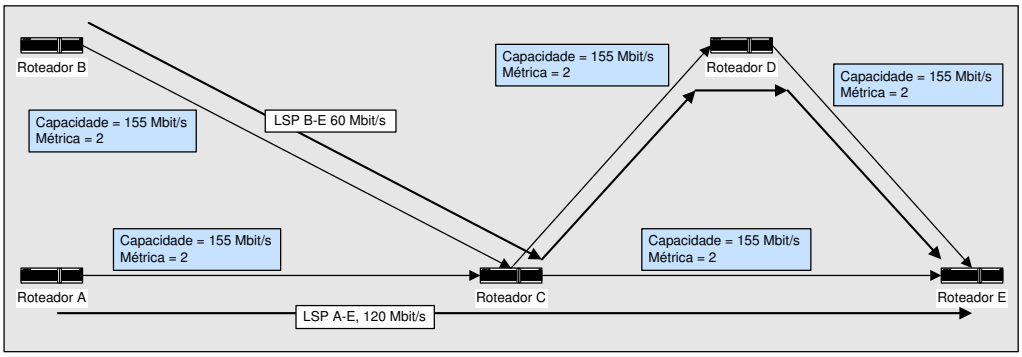


Figura 1.5: Prevenção de congestionamento com MPLS e CBR.

A distribuição ineficiente de tráfego é resolvida através a criação de novos LSPs quando o tráfego de uma fonte excede a capacidade do caminho de menor custo. A distribuição de carga é especificada conforme desejado, podendo ser derivada automaticamente da especificação de largura de faixa para o fluxo de tráfego. A figura 1.6 exemplifica este caso. Se o tráfego de A para C chegar à 160 Mbit/s, então um novo LSP de 30 Mbit/s é criado de A para C passando por B. Esta solução não seria possível com o roteamento de menor caminho nem com a opção ECMP dos protocolos IGP.

Outras vantagens são: especificação de rotas explícitas para LSPs possibilitando o controle preciso da trajetória do tráfego, obtenção de matrizes de tráfego através de estatísticas por LSP e definição de LSPs reservas para prover uma degradação elegante do nível de serviço em caso de falhas.

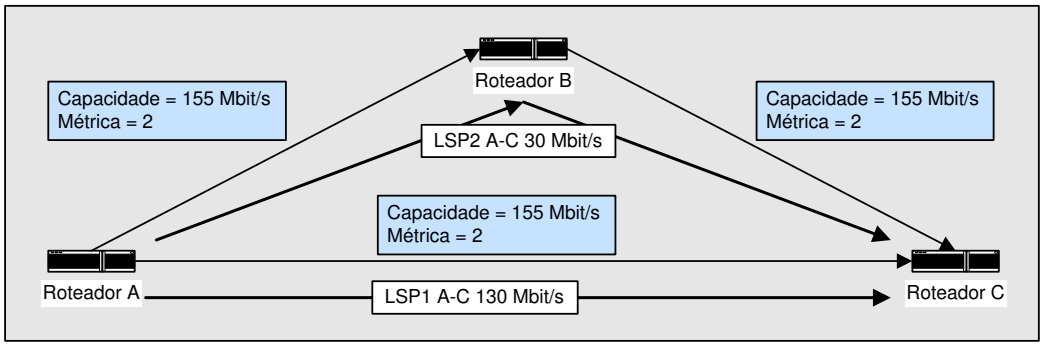


Figura 1.6: Balanceamento de carga com MPLS e CBR.

1.2.4 Serviços Integrados

A proposta de Serviços Integrados (*Integrated Services* - IntServ) [10, 26] determina que para prover QoS para fluxos de tráfego, os roteadores devem ser capazes de reservar recursos e, consequentemente, manter estados para cada um destes fluxos localmente.

Em sua arquitetura, três tipos de serviço são propostas de acordo com os requerimentos de retardo das aplicações. O Serviço Garantido (*Guaranteed Service*) [27] provê limites determinísticos de retardo enquanto que o Serviço de Carga Controlada (*Controlled-Load Service*) [28] provê limites probabilísticos. Ambos baseiam-se em exigências quantitativas de nível de serviço e requerem sinalização e controle de admissão. O terceiro tipo é o conhecido Serviço de Melhor Esforço, o qual é particionado em três categorias: rajada interativa (*interactive burst*) que abrange por exemplo tráfego HTTP (WWW), X e NFS; transferência em lotes interativa (*interactive bulk transfer*) como no caso de tráfego FTP; e transferência em lotes assíncrona (*asynchronous bulk transfer*) como no caso de tráfego de correio eletrônico.

O modelo de referência da arquitetura IntServ possui três componentes principais (figura 1.7): controle de admissão, o qual verifica se a rede pode suportar a solicitação de um novo serviço um função de seus requerimentos, evitando a degradação de outros; mecanismos de encaminhamento de pacotes, os quais realizam operações de classificação, suavização (*shaping*), policiamento (*policing*) e escalonamento (*scheduling*); e um protocolo de sinalização para a reserva de recursos. Embora a arquitetura não esteja amarrada a nenhuma solução específica, RSVP [29, 30] muitas vezes é referido como o protocolo de sinalização para IntServ [31, 32].

RSVP foi concebido como um protocolo de sinalização que permitisse que as

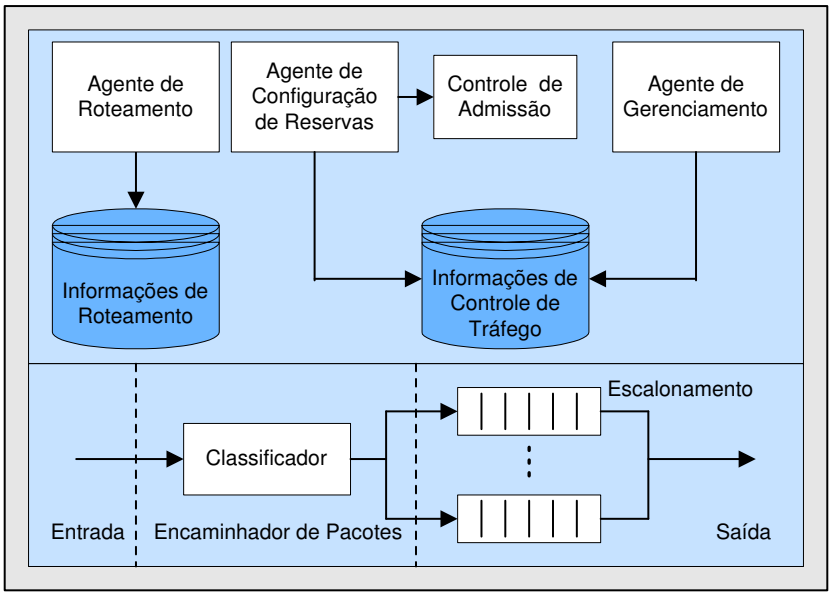


Figura 1.7: Modelo de referência da arquitetura IntServ em roteadores.

aplicações reservassem recursos. Na sinalização (figura 1.8), a fonte envia uma mensagem de caminho (*path message*) para o próximo nó de acordo com o protocolo de roteamento. Ao receber uma mensagem de caminho, o destino responde com uma mensagem de reserva (*reservation message*), solicitando os recursos para o fluxo de tráfego. Qualquer roteador intermediário pode rejeitar ou aceitar o pedido de uma mensagem de reserva. Se a mensagem for rejeitada, o roteador notifica o destino e o processo de reserva é terminado. Se a mensagem for aceita, largura de faixa e espaço em *buffer* são reservados para o fluxo e informações sobre o estado desta reserva são armazenados no roteador. Estas reservas são mantidas em estados voláteis (*soft states*) que precisam ser renovados periodicamente para permanecerem ativos. Para isto, mensagens de caminho são periodicamente enviadas da fonte para o destino, o que permite também o restabelecimento automático de uma reserva em casos de mudança da rota. Outra funcionalidade presente é a definição de estilos de reserva (*reservation styles*), onde filtros presentes nos roteadores e associados às reservas controlam quais pacotes podem utilizar determinados recursos.

A maior vantagem da proposta IntServ é a de prover classes de serviço que possam acomodar os requisitos de QoS de diferentes aplicações. Enquanto o serviço garantido é apropriado para aplicações críticas e intolerantes, o serviço de carga controlada suporta algumas aplicações críticas porém mas maleáveis. As demais são suportadas pelas categorias do serviço de melhor esforço.

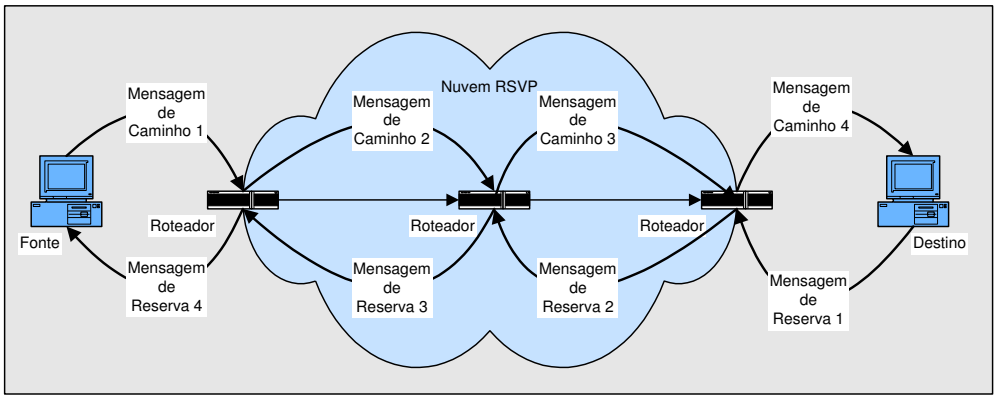


Figura 1.8: Sinalização RSVP.

Contudo, a aceitação dos serviços integrados por parte dos provedores de serviço tem sido limitada principalmente por problemas de propriedade escalar. Estes problemas surgem porque é necessário que os roteadores mantenham o controle dos estados de todos os fluxos de tráfego que estão sendo encaminhados por eles. Em enlaces com capacidades da ordem de gigabits e terabits, com milhões de fluxos simultaneamente ativos, a arquitetura IntServ impõe obstáculos em termos de implementação (devido à explosão de estados), gerência e contabilidade. Por estes motivos, atualmente o uso de RSVP e serviços integrados é mais recomendado para redes menores e confinadas [33].

1.2.5 Serviços Diferenciados

Uma das motivações para o surgimento da proposta de Serviços Diferenciados (*Differentiated Services* - DiffServ) [11, 12] foi a dificuldade de implementação dos serviços integrados e RSVP. Como uma alternativa de provisão de QoS, a arquitetura DiffServ objetiva prover serviços diferenciados de forma simples e escalável. Para isto, DiffServ baseia-se na premissa de que, em última instância, os fluxos de dados de aplicações diferentes podem ser classificados em um conjunto mais genérico e reduzido de categorias de tráfego. Seguindo este princípio, os fluxos de dados são discriminados e tratados de acordo com as suas classes de tráfego.

Entre domínios, níveis de serviço unidirecionais são acordados entre provedor e cliente (usuário ou outro provedor) para o tráfego que entra na rede do provedor. Internamente, o provedor do serviço é totalmente responsável pelo aprovisionamento dos recursos e pelas políticas de serviço.

Nas fronteiras de um domínio DiffServ (DS), o tráfego é condicionado para as-

segurar conformidade ao perfil de tráfego correspondente ao nível de serviço contratado, e que geralmente irá se basear no tráfego agregado. Para tratar os pacotes de forma diferenciada, são utilizados bits de um campo do cabeçalho IP, denominado campo DS [11]. Antes da entrada em um domínio DS, este campo é marcado com um valor que especificará o tipo de tratamento fornecido no encaminhamento deste pacote.

Devido a estas características, a complexidade ligada à classificação e ao mapeamento dos fluxos nas classes de tráfego é deslocada do núcleo para as bordas da rede de serviços diferenciados, simplificando a arquitetura do seu núcleo. Além disso, diferentes níveis de serviço são oferecidos ao tráfego agregado[3] e não a cada fluxo, o que torna a proposta DiffServ escalável. Estes dois pontos constituem as principais vantagens desta alternativa.

Contudo, a proposta de serviços diferenciados também apresenta alguns obstáculos ao seu desenvolvimento em larga escala. Conforme dito anteriormente, a complexidade concentrada nas bordas da rede através de aprovisionamento e configuração, contribui para a simplificação do núcleo. Porém, prover níveis distintos de serviço ao mesmo tempo não é uma tarefa simples em função das possíveis interações adversas entre classes diferentes, ou até mesmo dentro de cada classe. Mecanismos que minimizem estes problemas têm sido alvo de pesquisa atual, incluindo este trabalho (seção 1.3). Além disso, DiffServ baseia-se em acordos locais nas fronteiras entre provedores e clientes. Consequentemente, serviços fim-a-fim só podem ser obtidos através da conciliação dos contratos entre as fronteiras de cada domínio no caminho entre fonte e destino. A construção deste serviços fim-a-fim não é trivial, constituindo-se também em objeto de pesquisa atual.

Comparando de forma simples as propostas IntServ e DiffServ, pode-se dizer que soluções para QoS que utilizam armazenamento de estados (*stateful solutions*) como IntServ, provêem serviços mais flexíveis e fornecem maiores níveis de garantia em relação a soluções de QoS sem armazenamento de estados (*stateless solutions*). Contudo, as primeiras são menos escaláveis e robustas que as últimas. A arquitetura SCORE (*Scalable Core*) [34] é um exemplo de proposta intermediária entre IntServ e DiffServ, e que tenta reunir as vantagens de ambas.

Várias propostas do IETF para a provisão de QoS na Internet foram descritas, cada uma com seus objetivos, vantagens e limitações. Como integrar estas propostas

[3]Composto por vários fluxos de menor nível de agregação ("subfluxos") ou microfluxos. O mesmo que fluxo agregado.

de forma a construir uma infra-estrutura global que implemente QoS fim-a-fim é um ponto ainda em discussão [1, 4, 5, 35]. Os provedores de serviço também necessitam de mecanismos que regulem quais usuários, aplicações e estações devem ter acesso a quais serviços e em que condições. Esta infra-estrutura política, compreendendo um conjunto de protocolos, modelos de informação e serviços é necessária para transformar intenções administrativas em um tratamento diferenciado para os pacotes na rede [36]. O protocolo COPS (*Common Open Policy Service*) [37] é uma proposta do IETF para a provisão de QoS. Outro aspecto importante é a questão contratual expressa nos chamados Contratos de Nível de Serviço (*Service Level Agreements* - SLAs) [38], determinando os serviços a serem disponibilizados. Além disso, o sucesso da implantação de QoS depende de SLAs que maximizem os ganhos de ambas as partes, tanto provedor como usuário.

1.3 Objetivos e Organização do Texto

Dentre as propostas discutidas anteriormente, os serviços diferenciados constituem uma alternativa viável para a provisão de níveis de serviço discriminados e de alguma forma superiores ao de melhor esforço. Como consequência, uma larga variedade de estudos vem sendo feitos nesta área. Dentre eles, podem ser destacados trabalhos sobre modelos matemáticos [39, 40], tarifação [41, 42], adequação para transporte de voz, vídeo e tráfego *multicast* [43, 44, 45], propostas e avaliações de novos serviços [46, 47, 48], além de outros trabalhos citados no decorrer do texto.

Conforme já foi dito, a proposta Diffserv baseia-se em um conjunto de mecanismos que tratam pacotes de forma diferenciada em função da marcação dos bits do campo DS. Em termos de padronização, dois tratamentos ou Comportamentos Por Enlace (*Per-Hop Behaviors* - PHBs) foram especificados: o Encaminhamento Expresso (PHB-EF) [49] e o Encaminhamento Assegurado (PHB-AF) [50]. O PHB-AF serve de base para a implementação do Serviço Assegurado [51], o qual permite que um provedor ofereça diferentes níveis de garantia de encaminhamento de tráfego a seus clientes. Os pacotes são encaminhados com probabilidade alta se a taxa do tráfego de cada usuário não exceder o valor contratado. O tráfego excedente é encaminhado com probabilidades menores. O estudo deste serviço tem sido bastante incentivado pela falta de garantia no encaminhamento de tráfego na Internet.

Para garantir o nível de serviço desejado através da marcação do campo DS dos pacotes, o condicionamento de tráfego é executado na entrada de um domínio DS. Vários estudos propuseram marcadores para o serviço assegurado [52, 53, 54, 55] e

avaliaram os seus desempenhos [56, 57, 58, 59, 60, 61]. Como consequência, foram identificados vários problemas de justiça entre os fluxos que compõem o tráfego assegurado quando estes compartilham recursos da rede de um provedor. Mais especificamente, a falta de justiça entre fluxos que estão associados a um mesmo perfil de tráfego ainda é um problema pouco explorado.

O objetivo deste trabalho é propor um condicionador de tráfego eficiente em termos de justiça entre fluxos de um mesmo tráfego agregado, considerando o compartilhamento das larguras de faixa assegurada e excedente. Como ponto de partida, uma solução denominada FM (*Fair Marker*) [62] é apresentada. Estudos através de simulações são feitos de forma a entender o ajuste de seus parâmetros e avaliar o seu desempenho. De acordo com os resultados obtidos nas simulações, o FM garante um alto grau de justiça no compartilhamento da largura de faixa assegurada, a depender do ajuste adequado de seus parâmetros. Porém, o FM se mostra incapaz de prover justiça no compartilhamento da largura de faixa excedente, especialmente quando o fluxo agregado é formado pela mistura de fontes de tráfego TCP e UDP. Para suprir esta deficiência, este trabalho propõe a utilização de uma extensão do FM, denominada TCFM (*Three Color Fair Marker*) [63], cujo desempenho também é avaliado através de simulações.

O trabalho está organizado da seguinte maneira. O capítulo 2 aborda a arquitetura Diffserv, seus principais elementos e serviços propostos. O capítulo 3 descreve os mecanismos principais utilizados na implementação do serviço assegurado, incluindo marcadores de tráfego e gerenciamento ativo de filas. No capítulo 4, o problema da justiça é discutido, estruturado e suas causas analisadas. Em seguida, as principais estratégias para combater o problema da justiça entre fluxos de um mesmo agregados são comparadas. No final do capítulo, os marcadores FM e TCFM são descritos em detalhes. O capítulo 5 apresenta os modelos e cenários utilizados nas simulações e a análise dos resultados obtidos. Finalmente, no capítulo 6 são apresentadas as conclusões deste trabalho, assim como sugestões para trabalhos futuros.

O trabalho apresenta ainda cinco apêndices. O apêndice A contém os conceitos ligados ao controle de congestionamento do protocolo TCP, fundamentais para o entendimento do texto e dos resultados obtidos. Além disso, suas principais implementações são descritas e comparadas. O apêndice B contém os algoritmos de gerenciamento ativo de filas utilizados no trabalho. O apêndice C apresenta resultados adicionais referentes ao capítulo 5. O apêndice D contém um glossário e o apêndice E uma listas de endereços eletrônicos relacionados ao trabalho.

Capítulo 2

Serviços Diferenciados

Este capítulo aborda os Serviços Diferenciados (DiffServ) [11, 12], uma das propostas do IETF para QoS na Internet, conforme visto no capítulo 1. Entre outros aspectos, serão apresentados os principais componentes de sua arquitetura, as formas de tratamento diferenciado dos pacotes na rede e os principais serviços propostos. O capítulo termina com as considerações mais atuais sobre o assunto.

2.1 Fundamentos da Proposta

DiffServ busca viabilizar a criação de níveis distintos de serviço de forma escalável. Isto se deve à ausência de armazenamento de estados por fluxo e de sinalização por nó. A idéia fundamental é definir um conjunto pequeno de mecanismos que possam ser implementados nos nós da rede e que suportem uma variedade de serviços, fim-a-fim ou intra-domínio. Os requerimentos de desempenho que definem os serviços podem ser quantitativos ou estatísticos em cima de parâmetros como por exemplo vazão, retardo, *jitter* e perdas; ou podem ser especificados em forma de prioridades relativas de acesso aos recursos da rede.

O princípio de funcionamento da proposta DiffServ pode ser resumido da seguinte forma. Nas bordas da rede, os bits de um campo do cabeçalho IP denominado campo DS são marcados de modo a atribuir-lhe um valor específico. Esta marcação está condicionada aos requerimentos e regras de cada serviço. No interior da rede, os mesmos bits são utilizados para determinar o tratamento que será dado aos pacotes em termos de encaminhamento.

A arquitetura DiffServ é composta de um conjunto de elementos funcionais implementados em cada nó, incluindo: formas de encaminhamento dos pacotes em

função da marcação, denominadas Comportamentos Por Enlace (*Per-Hop Behaviors* - PHBs), funções de classificação de pacotes e funções de condicionamento de tráfego. A proposta se torna escalável na medida em que as funções mais complexas de classificação e condicionamento são implementadas apenas nas bordas, onde a concentração de tráfego é menor. Além disso, os PHBs são aplicados ao tráfego agregado e não a cada fluxo, eliminando a necessidade de manutenção de estados para fluxos individuais no núcleo da rede.

De uma forma mais abrangente, três elementos são utilizados para implementar os serviços diferenciados, sendo importante distinguí-los:

- o serviço a ser entregue ao tráfego agregado;
- os elementos da arquitetura DiffServ expostos acima (classificação de pacotes, condicionamento de tráfego e PHBs), utilizados para implementar os serviços;
- o valor do campo DS (*DS codepoint* ou DSCP) usado para marcar os pacotes e selecionar o PHB.

Vale notar que a definição e a provisão dos serviços estão separados das definições das formas de encaminhar os pacotes internamente na rede. Duas implicações importantes surgem deste desacoplamento. Em primeiro lugar, um espaço é criado para o desenvolvimento de uma variedade de serviços em cima dos PHBs, tornando a proposta mais flexível. Em segundo lugar, os PHBs mais básicos podem continuar os mesmos enquanto que os serviços podem ser criados e reformulados continuamente, garantindo a evolução da proposta ao longo dos anos. Por este motivo, a especificação dos serviços não faz parte do escopo de trabalho do IETF na área de DiffServ.

2.2 Definição do Campo DS

O campo DS tem a função de indicar, para cada pacote, qual o comportamento por nó que deve ser atribuído a ele. Este campo substitui os octetos TOS (Type Of Service) no IPv4 e classe de tráfego no IPv6 [11]. Os seis primeiros bits, utilizados para a marcação de tráfego, são chamados de *codepoint* ou DSCP. Os outros dois bits não são utilizados atualmente e estão reservados para uso futuro. Porém, foi proposto [64] que o campo DS corresponda apenas ao seis bits menos significativos dos octetos, e que o termo *codepoint* corresponda ao valor do campo DS. Esta

última definição tem sido adotada na literatura e o mesmo será feito neste texto[1]. A figura 2.1 mostra a localização do campo DS nos cabeçalhos dos protocolos IPv4 e IPv6.

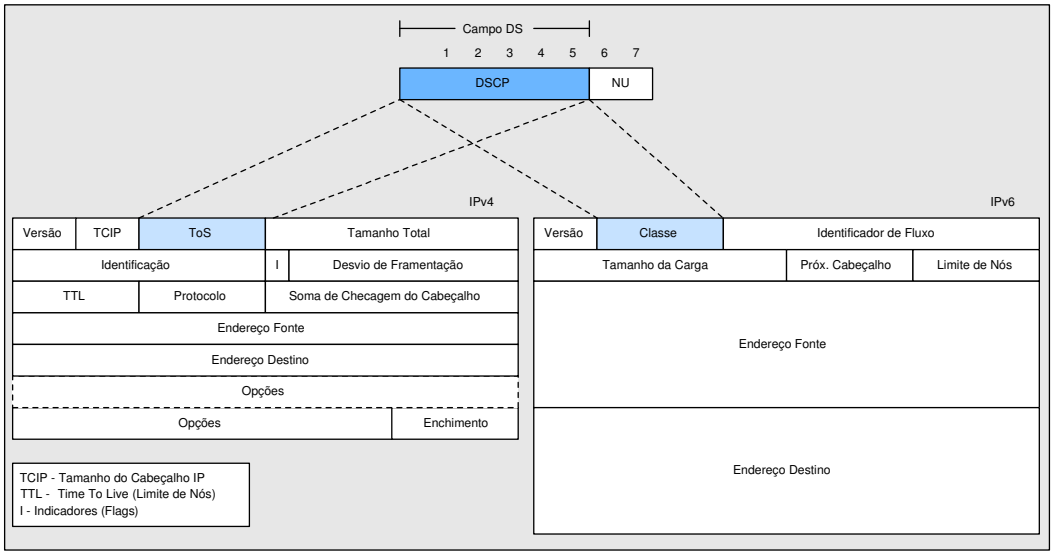


Figura 2.1: Campo DS no IPv4 e no IPv6.

Os seis bits do campo DS definem um espaço de *codepoints* de sessenta e quatro valores possíveis. Este espaço é dividido em três regiões de acordo com a tabela 2.1. A maior região abriga os valores sujeitos à padronização enquanto que as demais estão reservadas para experimentos e uso local.

Tabela 2.1: Regiões do espaço de *codepoints*.

Valores	Núm.	Política de uso
`XXXXX0`	32	Sujeita à ação dos padrões
`XXXX11`	16	Experimentos e uso local
`XXXX01`	16	Experimentos e uso local; sujeita à ação dos padrões caso necessário

Quando um pacote tem um *codepoint* atribuído ao campo DS, ele passa fazer parte de um tráfego agregado composto por todos os pacotes de mesmo valor de *co-*

[1] GROSSMAN [64] propôs correções nas definições de alguns termos do contexto de DiffServ, assim como introduziu novos termos. Este texto adota estas modificações por entender que procedem e são consenso dentro do meio acadêmico, embora não oficializadas até o momento.

depoint. O mesmo PHB deve ser aplicado a toda esta porção de tráfego, denominada Agregado de Comportamento (*Behavior Agregate* - BA).

2.3 Arquitetura DiffServ

A arquitetura DiffServ foi proposta [12] com vários objetivos, dentre eles o de ser escalável e poder evoluir continuamente. É assimétrica proporcionando a diferenciação dos serviços em apenas uma direção do tráfego. A seguir, seus principais conceitos serão descritos.

2.3.1 Domínios, Regiões e Nós DS

Um nó DS é aquele em conformidade com os documentos que definem DiffServ [11, 12], sendo capaz de suportar os serviços e PHBs definidos. Por sua vez, um domínio DS corresponde a um conjunto contíguo de nós DS que operam sob uma mesma política de provisão de serviços, possuem o mesmo conjunto de PHBs implementados em cada nó, e possuem uma fronteira bem definida. Um domínio DS pode consistir de uma ou mais redes sob mesma administração, ISPs ou intranets organizacionais.

Um domínio DS possui nós DS de fronteira e nós DS interiores. Nós DS de fronteira interconectam um domínio DS a um outro domínio DS ou a um domínio não-DS. Também exercem funções de condicionamento de tráfego obedecendo um Contrato de Condicionamento de Tráfego (*Traffic Conditioning Agreement* - TCA), parte de um SLA que trata especificamente das regras de classificação de pacotes e condicionamento de tráfego[2]. Nós DS interiores conectam-se a outros nós DS interiores ou a nós DS de fronteira, e podem exercer condicionamento de tráfego em situações especiais. Um nó DS de fronteira pode atuar como um nó DS interior caso não haja necessidade de condicionar o tráfego entre dois domínios. Tanto nós DS de fronteira como nós DS interiores devem ser capazes de aplicar os PHBs apropriados aos pacotes baseando-se no DSCP.

O tráfego entra em um domínio DS através de um nó DS de ingresso e sai por

[2]GROSSMAN [64] definiu ainda mais dois termos importantes: Especificação de Nível de Serviço (*Service Level Specification* - SLS) e Especificação de Condicionamento de Tráfego (*Traffic Conditioning Specification* - TCS). Estes termos objetivam separar questões contratuais e de negócio tais como prazos, cobranças e multas, contidas no SLA e no TCA, de questões puramente técnicas que ficariam concentradas no SLS e no TCS, respectivamente.

um nó DS de egresso. Nós DS de ingresso asseguram a conformidade do tráfego com o TCA. Nós DS de egresso também podem condicionar o tráfego encaminhado do seu domínio para um vizinho em função do TCA entre eles.

Uma região DS compreende um ou mais domínios DS contíguos que suportam serviços diferenciados em rotas que os atravessam. Nas fronteiras entre domínios adjacentes que suportam diferentes PHBs e mapeamentos DSCP $\mapsto$ PHB, deve ser estabelecido um SLA que especifique o condicionamento de tráfego na fronteira entre eles. Já no caso de domínios que suportam um conjunto de mesmos PHBs e mapeamentos DSCP $\mapsto$ PHB, a necessidade de condicionamento de tráfego na fronteira pode ser eliminada para os pacotes com DSCPs comuns aos dois domínios. A figura 2.2 ilustra os conceitos discutidos nesta subseção.

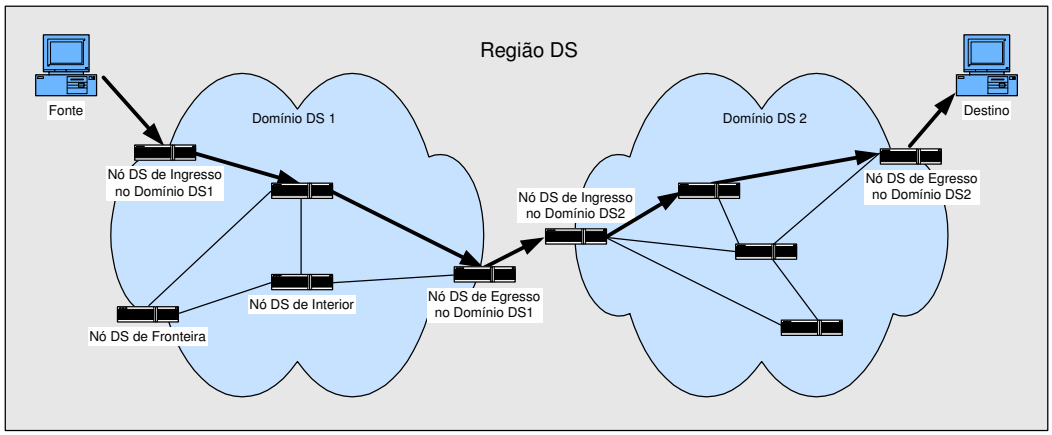


Figura 2.2: Domínios, regiões e nós DS.

2.3.2 Comportamento Por Enlace (PHB)

Um PHB corresponde à descrição das ações que um nó DS deve exercer sobre um agregado de comportamento (BA) em termos do encaminhamento dos seus pacotes. Serviços podem ser construídos utilizando um ou vários PHBs relacionados entre si e especificados em conjunto, constituindo um grupo de PHBs.

Os PHBs devem ser definidos em termos de características de comportamento relevantes à política de provisão de serviços e não em termos de mecanismos particulares para implementá-los. Isto é, são válidas especificações de reserva de recursos tais como *buffer* e largura de faixa, e de características observáveis de tráfego incluindo retardo, perdas e prioridade em relação a outros PHBs. Por outro lado, PHBs não podem ser definidos em cima de disciplinas de filas ou algoritmos de es-

calonamento específicos, ainda que tais mecanismos possam ser sugeridos. O fato de a definição de um PHB não estar presa às formas de implementá-lo é fundamental para permitir a continuidade e evolução dos serviços diferenciados. Ou seja, novos mecanismos podem sempre estar sendo desenvolvidos para melhorar o desempenho do PHB, enquanto que o seu conceito permanece o mesmo. Um analogia pode ser estabelecida com o roteamento na Internet. Passados vários anos desde o início da Internet, o conceito de roteamento continua basicamente o mesmo, onde o pacote é encaminhado nó a nó em função do endereço IP destino. Porém, ao longo de todos estes anos novos algoritmos foram criados e modificados para melhorar o desempenho desta tarefa.

A implementação, configuração, operação e administração dos PHBs e grupos de PHB suportados nos nós de um domínio DS devem particionar efetivamente os recursos da rede por entre os agregados de comportamento, de modo a garantir o cumprimento dos SLAs. Condicionadores de tráfego podem controlar o uso destes recursos através da aplicação dos TCAs e também de informações operacionais dos nós e de outros condicionadores de tráfego no domínio.

2.3.3 Classificação e Condicionamento de Tráfego

Para estender serviços através de domínios DS, é necessário estabelecer entre esses domínios um SLA cujo TCA especificará regras de classificação e condicionamento de tráfego nas suas fronteiras.

Uma política de classificação de pacotes identifica um subconjunto de tráfego que deve ser condicionado de acordo com o TCA, recebendo os valores apropriados de DSCP. Para executar esta tarefa, dispositivos denominados classificadores baseiam-se em alguma informação do cabeçalho dos pacotes IP para dirigí-los aos seus condicionadores de tráfego. Na arquitetura DiffServ dois tipos foram definidos [12]: o classificador BA (*Behavior Aggregate*) que seleciona os pacotes baseando-se apenas no campos DS; e o classificador MF (*Multi-Field*) que utiliza um ou mais campos dentre o endereço IP fonte, endereço IP destino, campo DS, identificador de protocolo, porta fonte e porta destino.

BERNET *et al.* [65] apresentaram classificadores de forma mais flexível, onde podem ser utilizados vários critérios de seleção denominados filtros. Valores exatos, prefixos, máscaras, intervalos e cadeias de caracteres podem ser utilizados para a classificação em cima de campos do protocolo IP e até da camada de enlace. Exemplos deste último caso incluem o campo prioridade do padrão IEEE 802.1p e o campo

identificador de VLAN (*Virtual Local Area Network*) do padrão IEEE 802.1Q.

Condicionadores de tráfego objetivam assegurar que o tráfego que entra num domínio DS está em conformidade com o perfil especificado no TCA. O perfil de tráfego corresponde a um conjunto de propriedades temporais de um fluxo de tráfego selecionado por um classificador. Logo, medindo estas propriedades temporais pode-se determinar se um pacote está dentro (*in-profile*) ou fora (*out-profile*) do perfil e diferentes ações podem ser aplicadas em função desta condição.

Os condicionadores de tráfego executam funções (não necessariamente todas) de medição, suavização (*shaping*), marcação e policiamento (descarte). A complexidade destas funções depende do perfil de tráfego, que por sua vez está ligado ao tipo de serviço que se quer oferecer.

A figura 2.3 mostra o diagrama de blocos de um condicionador de tráfego genérico processando um fluxo de tráfego selecionado por um classificador. O medidor mede as propriedades temporais do tráfego selecionado por um classificador contra o perfil desejado. O resultado desta medição age sobre a marcação, o descarte e a suavização aplicada sobre o tráfego, conforme ele esteja dentro ou fora do perfil. O marcador apenas atribui ou modifica (remarca) o valor do campo DS com o *code-point* apropriado, adicionando cada pacote a um agregado de comportamento. O suavizador (*shaper*) atrasa pacotes de um fluxo de tráfego para ajustá-lo ao perfil. Com o mesmo propósito, o "descartador" (*dropper*) pode eliminar pacotes.

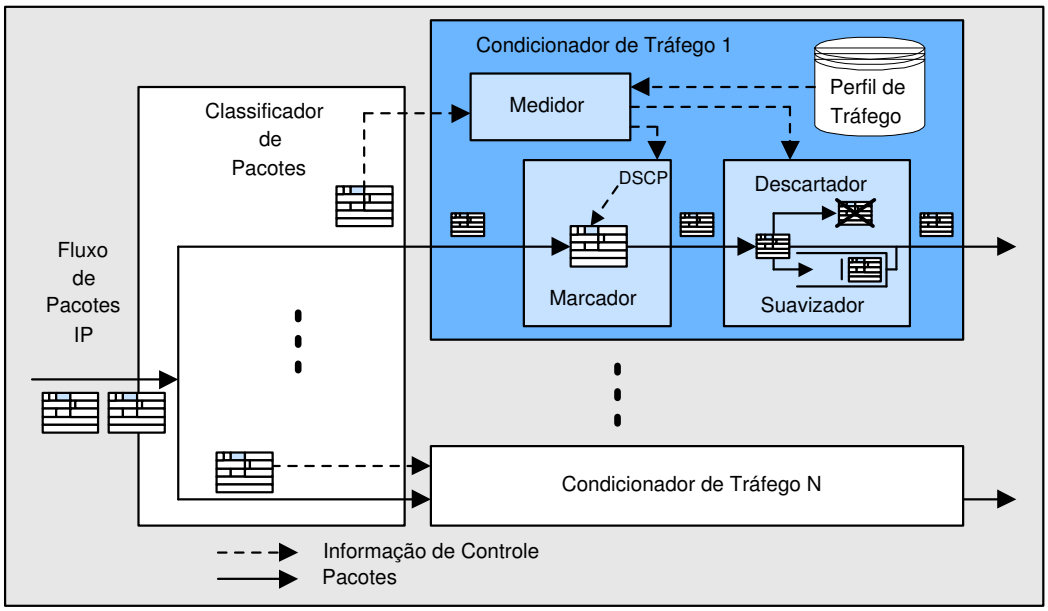


Figura 2.3: Condicionador de tráfego.

Geralmente, condicionadores de tráfego e classificadores localizam-se na fronteira de um domínio DS, no nó de egresso do domínio anterior (*upstream domain*) ou no nó de ingresso do domínio posterior (*downstream domain*). Ainda há a possibilidade de ambos os nós de fronteira desempenharem estas funções, o primeiro executando e o segundo policiando para assegurar obediência ao TCA.

A classificação e o condicionamento de tráfego podem ainda estar localizados no domínio DS de origem (onde o tráfego é gerado), em domínios não-DS e em nós DS interiores. No primeiro caso, fontes de tráfego ou nós intermediários se encarregam destas funções. As vantagens desta marcação inicial ou pré-marcação são a simplificação das regras de classificação devido à proximidade da aplicação, e a diminuição da carga de processamento nos nós de fronteira através da distribuição da marcação. Neste caso ainda pode haver a necessidade de condicionamento na fronteira para policiar o tráfego. No segundo caso, fontes de tráfego e nós intermediários em domínios não em conformidade com a proposta DiffServ executam as funções de classificação e condicionamento (pré-marcação), assim como no caso anterior. Porém, as regras locais que regem estas funções ficam escondidas. Além disso, o condicionamento na fronteira se torna obrigatório antes do ingresso em um domínio DS. No último caso, nós DS interiores abrigam classificadores e condicionadores de tráfego. Apesar da arquitetura DiffServ determinar que a complexidade advinda da implementação destes mecanismos esteja nas fronteiras, o aparecimento destas funções no interior da rede não é proibido. Contudo, esta alternativa só deve ser utilizada em casos muito específicos devido aos problemas de propriedade escalar. Um exemplo seria a restrição de acesso a enlaces de dados estratégicos de um provedor para determinados clientes.

2.4 Propostas de PHBs

2.4.1 PHB Padrão

Deve existir um PHB padrão em cada nó DS [11]. O nível de serviço deve corresponder ao de melhor esforço. Este PHB é util quando se deseja implementar um nível de serviço para pacotes que ingressam num domínio DS com *codepoint* não padronizado ou não definido para uso local. Assim, é permitido que usuários que não contratam serviços diferenciados continuem utilizando a rede. O valor de *codepoint* recomendado para o PHB padrão é `000000`.

2.4.2 PHBs Selecionadores de Classe

Para assegurar compatibilidade com o uso corrente do campo precedência do protocolo IP (subseção 1.2.2), um grupo de PHBs denominado PHBs Selecionadores de Classe foi especificado. O espaço de *codepoints* reservado é `XXX000`. Dados dois pacotes com valores de *codepoint* diferentes e pertencentes a PHBs selecionadores de classe, o pacote de maior valor de *codepoint* deve ter maior ou (no mínimo) igual prioridade de encaminhamento em relação ao de menor valor.

2.4.3 Encaminhamento Expresso (PHB-EF)

O Encaminhamento Expresso ou PHB-EF [49] é definido como o tratamento de um agregado de comportamento particular, onde sua taxa de saída em cada nó sempre é maior ou igual a uma taxa configurada. O PHB-EF deve maximizar o recebimento desta taxa independentemente da intensidade de outros tráfegos atravessando um nó, minimizando ao mesmo tempo o impacto nestes.

Para que a presença de outros PHBs não interfira no tráfego EF, mecanismos de preempção podem ser utilizados. Porém, em caso de preempção ilimitada, deve haver uma taxa máxima permitida para o tráfego EF a fim de restringir o prejuízo a outros PHBs. Todo o excesso em relação à taxa máxima deve ser descartado. Portanto, em termos de configuração, o PHB-EF pode ser definido através de uma taxa mínima e uma taxa máxima.

O PHB-EF pode ser usado para construir um serviço fim-a-fim de baixa perda, baixo retardo, baixo *jitter* e largura de faixa assegurada. Devido a estas características, este serviço é conhecido como linha privada virtual. Para satisfazer os seus requerimentos, as filas ao longo do caminho devem ser mantidas vazias ou quase vazias, pois perdas, retardo e *jitter* são causados pelo enchimento das filas. Isto pode ser conseguido garantindo que em cada nó a taxa máxima de chegada do tráfego agregado EF seja menor do que a sua taxa de saída.

A implantação do serviço fim-a-fim descrito acima requer a configuração dos nós DS para que os agregados tenham sempre uma taxa de saída bem definida. Isto é, filas mantidas praticamente vazias e influências de outros tipos de tráfego minimizadas. Esta configuração também deve levar em conta os pontos de convergência e divergência dos agregados de tráfego através dos enlaces. Além disso, funções de condicionamento de tráfego, incluindo descarte e suavização, devem ser aplicadas a cada agregado, garantindo conformidade à taxa contratada.

Em termos de implementação, vários mecanismos podem ser empregados, dentre eles: uma fila com prioridades simples, uma ou várias filas servidas por um escalonador WRR (*Weighted Round-Robin*) [66] onde a taxa do tráfego EF corresponde à taxa contratada e, por último, um escalonador CBQ (*Class-Based Queuing*) [67] onde o tráfego EF tem prioridade sobre os demais até a taxa contratada.

O valor de *codepoint* recomendado para o PHB-EF é `101110`.

2.4.4 Encaminhamento Assegurado (PHB-AF)

Conforme visto no capítulo 1, existe uma demanda para um encaminhamento assegurado de pacotes IP na Internet. Ou seja, a necessidade de garantir que pacotes IP sejam entregues com alta probabilidade desde que tráfego não exceda uma determinada taxa contratada.

O Encaminhamento Assegurado ou PHB-AF [50] especifica quatro grupos de PHBs (classes AF). Para cada classe é reservada uma quantidade de recursos em cada nó DS. Dentro de cada classe existem três níveis diferentes de precedência de descarte. Pacotes IP que queiram se utilizar de um grupo de PHBs AF devem ser marcados (pelo cliente ou provedor) com DSCP pertencente à classe AF correspondente, e com uma das três prioridades dentro dessa classe. Em caso de congestionamento, a prioridade de descarte determina a importância relativa de cada pacote dentro de cada classe.

Portanto, em cada nó de um domínio DS que implemente o PHB-AF, o nível de garantia de encaminhamento de um pacote IP dependerá, na ordem que se segue:

- da quantidade de recursos reservada para a classe AF a que pertence o pacote;
- da carga de tráfego existente na rede para a classe AF a qual pertence o pacote (nível de congestionamento dentro da classe);
- da prioridade de descarte do pacote dentro de sua classe em casos de congestionamento.

A tabela 2.2 mostra os *codepoints* recomendados para o PHB-AF, cujo formato geral é CP0. "C" corresponde aos três bits com o número da classe (de 1 a 4) e "P" aos dois bits que indicam a prioridade de descarte (de 1 a 3). Quanto maior o valor da prioridade de descarte, maiores são as chances do pacotes ser descartado. Um *codepoint* também pode ser representado pela notação AFxy, onde "x" representa a classe e "y" a prioridade de descarte.

Tabela 2.2: *Codepoints* recomendados para o PHB-AF.

Prioridade de descarte	Classe 1	Classe 2	Classe 3	Classe 4
1 - Baixa	001010	010010	011010	100010
2 - Média	001100	010100	011100	100100
3 - Alta	001110	010110	011110	100110

Dentro de cada classe AF, um nó DS deve aceitar pacotes dos três DSCPs e prover no mínimo dois níveis diferentes de probabilidade de descarte. Caso apenas duas probabilidades de descarte sejam implementadas em uma classe "x", pacotes com o DSCP AFx1 terão probabilidade de descarte menor enquanto que pacotes com os DSCPs AFx2 e AFx3 terão probabilidade de descarte maior.

Recomenda-se que um nó DS implemente todas as quatro classes AF. Além disso, mais classes e prioridades de descarte podem ser definidas para uso local. Pacotes de uma classe têm que ser encaminhados de forma independente em relação aos das demais. Porém, nada impede que uma classe AF possa receber recursos de outros grupos de PHBs e outras classes AF, havendo disponibilidade.

Quanto ao condicionamento de tráfego, um domínio DS pode controlar a quantidade de tráfego AF de cada prioridade de descarte que entra e sai deste através de descartes, suavização, aumento e diminuição da prioridade de descarte ou mudança de classe AF. No entanto, um nó DS nunca poderá reordenar pacotes de um mesmo microfluxo e mesma classe AF.

O PHB-AF deve minimizar congestionamentos de longa duração. Quando ocorrerem, pacotes devem ser descartados através de um algoritmo que seja insensível às características de curta duração de um microfluxo. Isto é facilitado através do uso de uma função aleatória de descartes. Assim sendo, fluxos de diferentes formatos de rajadas em escalas de tempo mais curtas, mas com mesmas taxas de transmissão em escalas de tempo mais longas, devem ter as mesmas taxas de perda a longo prazo.

Por outro lado, congestionamentos de curta duração devem ser suportados a fim de acomodar rajadas de tráfego. Quando ocorrerem, pacotes devem ser enfileirados. Como consequência, deve-se utilizar disciplinas de filas que operem em função do nível médio de pacotes. Recomenda-se também que congestionamentos sejam sinalizados aos nós fontes através de descartes graduais, a fim de permitir que o sistema entre em equilíbrio (não oscile).

2.5 Serviços Propostos

Conforme visto anteriormente, a definição dos serviços que podem ser construídos através da arquitetura DiffServ forma uma camada independente do modo de encaminhamento dentro da rede representada pelos PHBs. Isto é, vários serviços distintos podem ser desenvolvidos utilizando um mesmo PHB. De qualquer maneira, estes dois componentes importantes sempre estarão intimamente ligados no sentido da adequação do PHB ao serviço que se quer disponibilizar.

A seguir, as propostas de serviço diferenciados mais importantes serão descritas sucintamente. Vale a pena notar que a maioria delas surgiu antes da especificação da arquitetura DiffServ, servindo de impulso para iniciativas na direção de padronizá-la.

2.5.1 Serviço Premium

O serviço premium [68] é caracterizado por uma taxa de transmissão de pico e por retardos extremamente baixos. Além disso, a taxa de pico para o tráfego entrante em cada borda deve ser menor que a capacidade do enlace de saída. Nas bordas da rede, o tráfego agregado é condicionado de forma a garantir esta condição. Duas ações são permitidas: descartar os pacotes acima da taxa de pico (policiamento) ou atrasá-los até que o tráfego entre novamente em conformidade com o perfil especificado (suavização).

Pelas suas características, não é difícil observar que o serviço premium pode ser suportado pelo PHB-EF. Na verdade, sua proposta antecede a do PHB-EF e serviu de estímulo à sua especificação. Este serviço pode ser utilizado para voz [43, 69, 70], videoconferência, linha privada virtual, transferência de arquivos em tempo fixo e aplicações de baixo retardo.

2.5.2 Serviço Assegurado

O serviço assegurado [51, 71] é caracterizado por um nível de garantia estatístico e portanto menos rígido que o serviço premium. O nível de serviço é definido normalmente por uma largura de faixa contratada e possivelmente um certo grau de flutuação (rajadas curtas). O tráfego é condicionado nas bordas para cada cliente de modo a garantir conformidade com os seus respectivos perfis de tráfego. Para cada usuário, o tráfego que obedece ao seu perfil de tráfego deve ser entregue com alta probabilidade. O restante, fora do perfil, constitui um tráfego oportunista que é marcado de forma diferente. Em instantes de congestionamento, pacotes marcados

como fora do perfil terão preferência de descarte. Cabe ao provedor provisionar a sua rede para garantir a alta probabilidade de encaminhamento do tráfego dentro do perfil. Este serviço é adequado tanto para aplicações tradicionais baseadas em TCP tais como transferências de arquivos, acessos a banco de dados e servidores WWW, como para aplicações mais sofisticadas como voz [43, 70] e vídeo em tempo real.

Os princípios do serviço assegurado foram introduzidos por CLARK e FANG [52] através de uma proposta denominada estrutura de capacidade esperada (*expected capacity framework*). Este trabalho em muito impulsionou o surgimento da proposta DiffServ. Mais tarde, este esquema foi combinado com o serviço premium para a definição de uma arquitetura de serviços diferenciados utilizando um campo de dois bits no cabeçalho do protocolo IP [72]. Dependendo do valor deste campo um dos serviços seria escolhido, introduzindo a idéia mais tarde formalizada através da definição do campo DS, onde o *codepoint* define o PHB.

O serviço assegurado serviu de base para a especificação do PHB-AF. Conforme visto na seção 4.1, o serviço assegurado constitui o cenário para os estudos deste trabalho e será visto em detalhes no capítulo 3.

2.5.3 Serviço Olímpico

O serviço olímpico [73] provê três níveis de serviço: ouro, prata e bronze. Em casos de congestionamento, o serviço ouro irá obter uma largura de faixa maior que o serviço prata, que por sua vez irá obter uma fatia maior que o serviço bronze. Quando não existem pacotes dos serviços ouro e prata, o serviço bronze pode obter toda a largura de faixa disponível. Os pacotes são classificados e marcados de acordo com o nível de serviço desejado nas bordas da rede. Não existe uma forma de diferenciação pré-definida para os três níveis. Um exemplo seria atribuir percentuais de obtenção de largura de faixa de 60% para o ouro, 30% para o prata e 10% para o bronze. Estas seriam as frações de largura de faixa obtidas por cada agregado quando um enlace estiver congestionado. No serviço olímpico, usuários não especificam um perfil de tráfego particular, não há controle de admissão e os fluxos não são suavizados nem policiados de nenhuma forma. Um exemplo de aplicação desta solução inclui diferenciação do nível de serviço para usuários dentro de um ISP, empresa ou campus.

2.5.4 *User-Share Differentiation* (USD)

O serviço USD [74] introduz dois termos para a atribuição de largura de faixa: usuário e fatia. O usuário se refere ao cliente podendo ser uma rede, um grupo de redes ou um usuário individual. A cada usuário é atribuída uma fatia calculada em função de alguns parâmetros como por exemplo o quanto cada usuário está pagando pelo serviço. Definidas as fatias de cada usuário, o serviço USD irá dividir a largura de faixa excedente em cada enlace por entre estes usuários de forma proporcional à fatia de cada um. Portanto, esta proposta permite um ISP diferenciar fluxos de tráfego por usuário da seguinte forma. Primeiro, cada usuário tem uma garantia de largura de faixa mínima em alguns ou todos os enlaces do seu domínio, que corresponde à sua taxa contratada. Segundo, a largura de faixa excedente em cada enlace é dividida proporcionalmente à fatia de cada usuário. Este serviço difere do serviço olímpico na medida em que esta proporção deve ser mantida independentemente da ocorrência de congestionamentos na rede. Logo, enquanto a fatia total de cada usuário depende do tráfego competidor (outros agregados), a proporção entre eles é mantida fixa.

BAUMGARTNER *at al.* [75] descreveram o serviço USD de forma mais detalhada juntamente com os serviços anteriores. BASU e WANG [76] compararam o desempenho do serviço USD, a arquitetura de dois bits e a estrutura de capacidade esperada.

2.5.5 Diferenciação Relativa de Serviços

Na diferenciação relativa de serviços [46], o tráfego da rede é agrupado em N classes de serviço, as quais são diferenciadas através do nível de qualidade no encaminhamento de seus pacotes. Mais formalmente, uma classe i é melhor (ou pelo menos não pior) do que a classe $(i-1)$, para $1 \leq i \leq N$, em termos de retardo e taxa de perda em cada nó. Um exemplo simples de diferenciação relativa são os PHBs selecionadores de classe. Neste caso, oito classes distintas são obtidas com níveis de serviço diferenciados, onde a classe mais alta (maior valor de *codepoint*) tem maior probabilidade de encaminhamento de seus pacotes (melhor nível de serviço) que a classe imediatamente inferior e assim por diante.

O conceito de diferenciação relativa se opõe à noção de diferenciação absoluta de serviços [77]. Neste último caso, o usuário requisita à rede um parâmetro absoluto de desempenho, tal como largura de banda ou retardo fim-a-fim. Cabe então ao

provedor aceitar ou não o pedido em função dos recursos disponíveis, e de outras garantias porventura existentes e passíveis de sofrerem alguma degradação. Uma característica importante deste modelo é a necessidade de reserva de recursos, e por conseguinte, controle de admissão. O serviço premium é um exemplo de diferenciação absoluta com relação à largura de faixa contratada.

Já na diferenciação relativa os usuários não obtêm garantias de nível de serviço absoluto, e pode não haver reserva de recursos nem controle de admissão. A única garantia está no fato de que quanto maior (melhor) a classe, melhor o serviço. Sendo assim, os usuários e aplicações escolhem a classe que melhor satisfaz os seus requisitos de desempenho, custo e outras restrições. De uma forma genérica, a quantidade de serviço recebida por uma classe e a resultante qualidade de serviço percebida por um usuário ou aplicação dependem da carga de tráfego instantânea de cada classe na rede. Portanto, variações de qualidade podem ocorrer, fazendo com que este modelo seja mais adequado para aplicações adaptativas.

Existem diversas maneiras de prover diferenciação relativa de serviços. Talvez a mais simples delas seja através de prioridades rígidas onde as classes maiores são servidas primeiro. Porém, este modelo pode levar a longos períodos de ausência de serviço para classes menores. Além disso, não há como ajustar o "espaço" (diferença) entre as classes. Por estas razões, diz-se que o modelo é incontrolável. Uma segunda forma de diferenciação relativa seria por discriminação de custo, tal como o esquema de Tarifação do Metrô de Paris (*Paris Metro Pricing* - PMP) [42]. A premissa deste esquema é impor preços maiores para classes mais altas para reduzir a carga e melhorar sua qualidade de serviço em relação às classes mais baixas e populosas. Uma desvantagem deste esquema é que existe a possibilidade de sobrecarga de usuários "ricos", fazendo com que o nível de serviço de classes mais baixas seja momentaneamente superior ao de classes mais altas. Este problema torna este modelo imprevisível. Um terceiro exemplo de diferenciação relativa é através da reserva de quantidades maiores de recursos (largura de faixa e espaço em *buffer*) para classes maiores, calculadas relativamente à carga esperada para cada classe. Este esquema, assim como no caso anterior, pode apresentar problemas em escalas de tempos mais curtas. Neste caso, grandes surtos de tráfego em classes mais altas podem levar o seu nível de serviço para patamares inferiores aos de classes mais baixas.

Com o objetivo de endereçar as limitações presentes nos esquemas citados de diferenciação relativa de serviços, um modelo de diferenciação proporcional foi definido [77]. O modelo de diferenciação proporcional é um serviço relativo no sentido

de que há garantia de que a classe i será sempre melhor do que a classe $i-1$. Porém, esta diferenciação obedece a uma relação proporcional onde uma classe i é c_i/c_j vezes melhor em relação à classe j. Os parâmetros $c_1 < c_2 < ... < c_N$ são denominados Parâmetros de Diferenciação de Qualidade (*Quality Differentiation Parameters* - QDPs).

Este modelo objetiva oferecer um serviço controlável na medida em que os administradores das redes devem ser capazes de ajustar a diferença de qualidade entre as classes, isto é, ajustar os QDPs de acordo com seus próprios critérios. Além disso, esta diferença deve ser consistente inclusive em escalas de tempo mais curtas e independente das variações de carga de cada classe. Isto torna este modelo também previsível, diferentemente do que acontece nos esquemas de diferenciação relativa abordados anteriormente. Foram definidas e avaliadas propostas de implementação do modelo proporcional para níveis diferenciados de retardo [47] e taxa de perda [48].

2.6 O QBone da Internet 2

O projeto Internet 2 é formado por uma parceira de centenas de organizações, incluindo universidades, empresas e outras entidades. Um dos seus principais objetivos técnicos é o de conceber e realizar uma arquitetura de QoS escalável, administrável e interoperável. Para atingir este objetivo, no final de 1998 foi lançada a iniciativa QBone [78]. Trata-se de um ambiente de teste instrumentado para prover DiffServ entre domínios. No QBone, serviços experimentais serão desenvolvidos, depurados e refinados de forma a acomodar as novas e avançadas aplicações de rede. Inicialmente, apenas o serviço premium do Qbone ou QPS (*QBone Premium Service*) será testado, utilizando o PHB-EF.

O QBone é formando por várias redes entre elas vBNS, Abilene e CA*net II. Para que os serviços diferenciados possam interoperar, SLAs (mais especificamente SLSs) devem especificar o serviço de trânsito dos agregados de tráfego por entre os domínios. Se o número de serviços diferenciados oferecido for pequeno e os contratos relativamente estáticos, os SLSs entre os domínios poderão ser manualmente negociados e os equipamentos configurados através de intervenção humana. Porém, se o número de serviços e usuários for grande, será necessário automatizar a negociação dos SLSs, o controle de admissão e a configuração dos equipamentos, de forma a suportar os serviços de QoS provisionados. Para isto, é necessário um mecanismo que execute sinalização de QoS entre estações e roteadores, e entre os domínios DS.

É importante notar que a arquitetura DiffServ proposta pelo IETF não inclui tal mecanismo como requerimento [79].

Para suplantar esta carência, foi introduzido o conceito de Corretores de Largura de Faixa (*Bandwidth Brokers* - BB). Um BB mantém informação sobre os SLSs definidos entre um domínio DS e seus clientes, sejam eles usuários locais ou redes adjacentes. Além disso, utiliza as informações dos SLSs para configurar os roteadores no domínio DS local e tomar decisões de controle de admissão.

A figura 2.4 ilustra o funcionamento dos BBs. Antes que seus pacotes possam ser transmitidos, uma fonte deve contactar o BB local para iniciar a reserva de um serviço. Neste momento, funções de autenticação e controle de admissão são desempenhadas. Se o serviço for admitido, o BB local inicia um pedido de reserva fim-a-fim através dos BBs dos domínios DS que formam o caminho que será atravessado pelo fluxo de tráfego. Nos domínios DS onde o serviço é admitido, cada BB configura os roteadores para suportá-lo. Deste modo, os BBs permitem que domínios DS administrados separadamente gerenciem seus recursos de forma independente, e ainda assim cooperem com outros domínios para prover serviços de QoS fim-a-fim, dinamicamente.

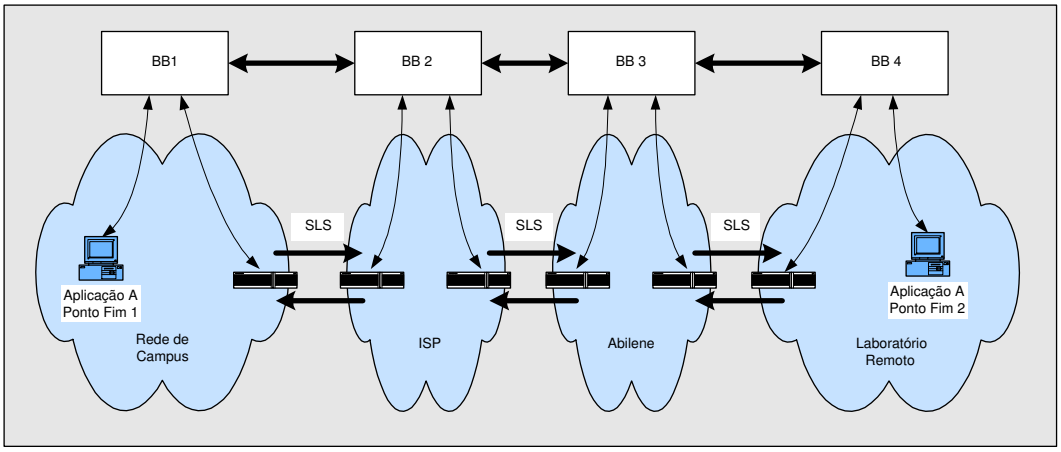


Figura 2.4: QBone: DiffServ interdomínios através de SLSs e BBs.

Capítulo 3

O Serviço Assegurado

Conforme visto na subseção 2.5.2, o Serviço Assegurado é um dos serviços diferenciados propostos para a Internet. Neste capítulo, este tipo de serviço será descrito de forma mais detalhada que no capítulo 2, juntamente com os mecanismos utilizados na sua implementação.

3.1 Definição

A multiplexação estatística na Internet proporciona um uso mais eficiente da largura de faixa, permitindo um número crescente de usuários e aplicações[1]. Por outro lado, há incerteza quanto à capacidade disponível e, consequentemente, quanto ao nível de serviço experimentado por usuários e aplicações a cada instante. Mas nem por isso os usuários da Internet se sentem inibidos de utilizar a rede, já que possuem uma expectativa quanto a este nível de serviço. Além disso, pequenas degradações são toleradas na maioria das vezes.

Baseando-se nestas premissas, o serviço assegurado visa atender usuários que porventura desejam contratar um nível de serviço que não precisa ser sempre satisfeito, mas que ao mesmo tempo deve apresentar baixas probabilidades de falha. Este serviço pode perfeitamente coexistir com os serviços de garantia absoluta, ideais para determinadas aplicações e necessidades de negócio, e que requerem muitas vezes 100% de disponibilidade. Sob este aspecto, o serviço assegurado representa uma alternativa menos custosa tanto para o provedor quanto para o cliente.

Este perfil de serviço provisionado de forma estatística foi batizado [51, 52] de

[1]Este ganho de multiplexação estatística ocorre devido à natureza explosiva (em rajadas) da maioria do tráfego que circula na Internet.

"perfil esperado" (*expected profile*), e corresponde normalmente a um valor de largura de faixa reservada para cada usuário (rede de uma empresa, estação de trabalho ou aplicação). Este termo não sugere uma garantia absoluta, mas sim uma expectativa que o usuário tem sobre o nível de serviço que vai receber em momentos de congestionamento. De uma certa forma, isto se assemelha ao que acontece no cenário atual da Internet, onde quase sempre se tem uma idéia (ainda que pouco precisa) do nível de serviço que será recebido, mesmo em horários de pico de tráfego. Por outro lado, se na Internet atual cada usuário recebe uma fatia imprevisível da capacidade disponível em momentos de congestionamento, no serviço assegurado esta divisão deve ser bem definida. Além disso, os níveis de expectativa de serviço podem ser diferentes para cada usuário, possibilitando políticas de tarifação com preços bem diferenciados.

Essencialmente, uma garantia estatística é função direta do aprovisionamento de recursos. Quanto maior o nível de aprovisionamento da rede, maiores serão as garantias de obtenção do serviço e menores as probabilidades de degradação do seu nível. O provedor deve portanto controlar a quantidade de tráfego assegurado que atravessa os vários enlaces da rede, e prover recursos suficientes para suportá-lo. Portanto, o controle de admissão passa a ser muito útil como forma de automatizar este processo. Além disso, deve-se proteger o tráfego prioritário, constituído pela soma das porções de cada usuário que obedece aos seus perfis contratados. Sendo assim, em instantes de congestionamento deve-se descartar preferencialmente o tráfego oportunista, representado pela soma das porções fora dos perfis contratados e pelo tráfego de melhor esforço. Consequentemente, um usuário que não ultrapassar o seu perfil de tráfego individual terá probabilidades muito baixas de descarte de pacotes. Caso o seu perfil seja ultrapassado, o usuário deve entender que o tráfego em excesso não será entregue com a mesma alta probabilidade.

Em termos de implementação, o serviço assegurado necessita de dois mecanismos principais: um antes da entrada do domínio do provedor, para identificar a porção de tráfego de cada usuário que está dentro ou fora do perfil; e outro, no interior da rede, para priorizar o tráfego dentro do perfil em momentos de congestionamento. Em termos de elementos da arquitetura DiffServ, o primeiro mecanismo corresponde aos condicionadores de tráfego, enquanto que o segundo corresponde ao PHB-AF.

Portanto, nas fronteiras do domínio DS do provedor, condicionadores de tráfego marcam pacotes com maior ou menor prioridade de descarte, de acordo com a aderência ou não ao perfil de tráfego contratado. No interior da rede, disciplinas

de gerenciamento ativo de filas aplicam o PHB-AF através de probabilidades de descarte distintas em função da marcação de cada pacote.

O serviço assegurado provê grande flexibilidade quanto à definição dos serviços a serem disponibilizados. Basicamente, um serviço será implementado através de um condicionador de tráfego específico, o qual assegura a aderência ao perfil de tráfego correspondente. Logo, para mudar o serviço, basta mudar o condicionador de tráfego. Internamente, os roteadores irão exercer um mecanismo único para cada agregado de comportamento, independente da natureza dos serviços oferecidos. Conforme os pacotes são transportados através da rede e agregados a outros fluxos, os roteadores nas bordas dos domínios vizinhos policiam somente o tráfego agregado.

São especialmente importantes para a descrição de um serviço:

- o que exatamente está sendo oferecido ao cliente (por exemplo, um 1 Mbit/s de largura de faixa);
- onde o serviço é válido (do cliente para um destino específico, do cliente para um grupo de destinos, em todo o domínio do provedor, etc.);
- o nível de garantia do serviço ou, de forma equivalente, o nível de incerteza que pode ser tolerado pelo cliente.

Uma questão importante é que o usuário pode não querer contratar serviços distintos para cada microfluxo (ou par fonte e destino), preferindo um contrato em cima do tráfego agregado. Na verdade existem fortes desvantagens em se definir níveis distintos de serviço para fluxos de fina granulosidade (baixo nível de agregação). Em primeiro lugar, um número exorbitante de especificações pode ser necessário. Em segundo lugar, a soma das larguras de faixa asseguradas de cada fluxo está limitada a princípio à capacidade do enlace de acesso do cliente para o provedor[2], o que força a diminuição do nível de serviço para cada fluxo individualmente. Finalmente, nem sempre é possível saber todos os destinos possíveis de antemão. Estes motivos favorecem ao condicionamento de tráfego por agregação, conforme será visto no capítulo 4.

Um outro ponto importante sobre a utilização do PHB-AF no serviço assegurado é o número de prioridades de descarte a ser utilizado. Conforme visto na seção 2.4.4, o PHB-AF provê três níveis de prioridade em cada uma de suas quatro classes.

[2]Uma forma de contrariar esta regra seria trabalhar com um certo grau de subdimensionamento da rede a ser compensado por um ganho de multiplexação estatística.

Apesar disso, sua implementação pode incluir duas ou três prioridades conforme a necessidade. De acordo com a definição do serviço assegurado, é necessário apenas diferenciar o tráfego prioritário do restante oportunista. Logo, duas prioridades de descarte são suficientes a princípio. Porém, duas razões distintas impulsionaram trabalhos que propõem a utilização de três níveis de prioridade no encaminhamento dos pacotes:

- o desejo de implementar mecanismos de condicionamento de tráfego mais sofisticados [53, 54, 55], os quais medem o tráfego contra um número maior de parâmetros. Um exemplo disso é medir o tráfego contra uma taxa assegurada e uma taxa de pico, marcando os pacotes com uma prioridade de descarte intermediária quando o tráfego se encontra entre estes dois valores. Algumas destas propostas serão descritas na seção 3.2 deste capítulo;

- melhorar a justiça no compartilhamento das larguras de faixa assegurada e excedente entre fluxos agregados de diferentes clientes, ou fluxos de uma mesma agregação de tráfego. Conforme será visto em detalhes no capítulo 4, problemas de justiça ocorrem devido a diversos fatores. Dentre eles, a presença de tráfego não responsivo[3] motivou o uso de três prioridades de descarte como forma de proteção ao tráfego TCP [61, 80, 81, 82, 83, 84].

A seguir, condicionadores de tráfego (seção 3.2) e disciplinas de gerenciamento ativo de filas (seção 3.3) serão vistos com mais detalhes a respeito dos seus empregos na implementação do serviço assegurado.

3.2 Condicionadores de Tráfego para o Serviço Assegurado

Conforme visto na subseção 2.3.3, os condicionadores de tráfego objetivam assegurar que os pacotes que entram num domínio DS estão em conformidade com o perfil contratado. O resultado principal do condicionamento de tráfego é a marcação dos pacotes, a qual determinará o tipo de encaminhamento que estes receberão

[3]O termo não responsivo se refere à ausência de controle de congestionamento, o qual força a redução da taxa de transmissão de um fluxo nesta situação. Aplicações que utilizam o UDP como protocolo de transporte são exemplos de tráfego não responsivo na Internet.

dentro do domínio DS. Por este motivo, os condicionadores de tráfego são chamados muitas vezes de marcadores[4].

No serviço assegurado, o perfil de tráfego corresponde basicamente a uma taxa assegurada. Portanto, o mecanismo de medição dos condicionadores de tráfego deve incluir uma forma de comparar a taxa de transmissão do tráfego com a taxa assegurada. Apesar do desempenho de uma aplicação ser sensível à vazão efetiva[5] (*goodput*), três razões importantes estimulam a medição da taxa de transmissão [85]. As duas primeiras são simplicidade e independência do protocolo de transporte. No caso do TCP por exemplo, medir a vazão efetiva implicaria em ter acesso aos pacotes de reconhecimento. Isto nem sempre é possível dependendo da localização do marcador, já que os caminhos de ida e volta podem ser distintos. Mas a principal razão para medir a vazão local é incentivar o envio de pacotes que têm grandes chances de serem entregues. Com isso, uma fonte de tráfego não responsiva mal comportada pode ser penalizada independentemente da entrega dos seus pacotes.

Os condicionadores de tráfego propostos na literatura podem ser classificados em duas categorias principais quanto à forma de medição: baseados em baldes de fichas e baseados em estimadores de taxa média. A seguir, serão descritas as características de cada categoria juntamente com alguns condicionadores de tráfego propostos.

3.2.1 Condicionadores de Tráfego Baseados em Baldes de Fichas

O algoritmo balde de fichas (*token bucket*) [86] pode ser utilizado para controle de congestionamento em redes (suavização de tráfego), assim como o algoritmo do balde furado (*leaky bucket*). Embora semelhantes, eles não devem ser confundidos.

Ambos os algoritmos possuem apenas dois parâmetros: o tamanho do balde B e a taxa de preenchimento (balde de fichas) ou vazamento (balde furado) do balde T. O parâmetro B pode ser expresso em bytes (ou bits) ou pacotes conforme os algoritmos sejam utilizados no modo em bytes[6] ou em pacotes, respectivamente. Da mesma forma, o parâmetro T pode ser expresso em bytes (ou bits) ou pacotes por segundo. A fim de comparar os dois algoritmos, será considerado que cada um dos suavizadores funciona no modo em pacotes e está conectado a um enlace com

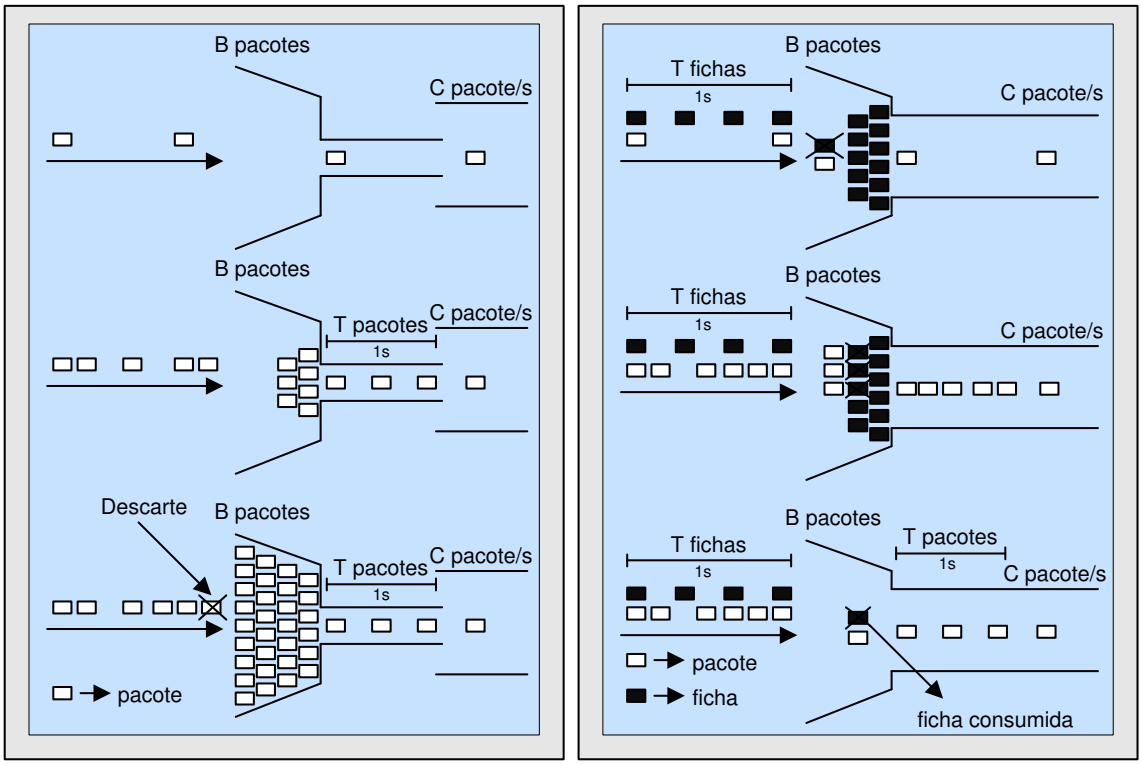
[4]Este texto utilizará daqui para frente o termo marcador e marcação como sinônimos de condicionador e condicionamento, respectivamente.

[5]Quantidade de informação corretamente entregue ao destino por unidade de tempo.

[6]Ideal para redes onde os pacotes têm tamanho variável.

capacidade de transmissão C, onde $C \geq T$.

No algoritmo balde furado (figura 3.1a), o tráfego pode ser injetado na rede a uma taxa máxima T, independente do valor de C. Enquanto a taxa de chegada não ultrapassa T, a taxa de saída acompanha a taxa de chegada[7] (primeiro desenho da figura 3.1a). Porém, quando a taxa de chegada é maior que T, o tráfego na saída fica limitado em T, sendo o excesso acumulado no balde (segundo desenho da figura 3.1a). Se esta situação persistir até que o balde seja totalmente preenchido com B pacotes, descartes começarão a ocorrer (terceiro desenho da figura 3.1a). Portanto, este algoritmo não permite que ocorram rajadas de tráfego com taxa superior à T, na saída do dispositivo. Em termos de implementação, o balde furado consiste em um fila de capacidade de armazenamento B e taxa de serviço constante T.



(a) Balde furado.

(b) Balde de fichas.

Figura 3.1: Mecanismos de suavização de tráfego.

No algoritmo balde de fichas (figura 3.1b), o balde é preenchido com fichas a uma taxa T. Cada pacote deve consumir uma ficha para ser transmitido[8]. Caso

[7] Assume-se que a taxa de saída está sendo observada em escalas de tempo bem maiores do que o tempo de transmissão de um pacote, onde a taxa de saída vale zero ou C.

[8] No modo em bytes, cada pacote deve consumir uma quantidade de bytes em fichas igual ao seu

não haja fichas, o pacote não é transmitido, podendo ser armazenado ou descartado dependendo da implementação. Do mesmo modo que no balde furado, a taxa de saída acompanha a taxa de chegada até quando o valor desta última é menor ou igual à T (primeiro desenho da figura 3.1b). Neste caso as fichas não consumidas vão sendo acumuladas no balde até enchê-lo. A partir daí, as fichas são perdidas. Porém, quando a taxa de chegada é maior que T, a taxa de saída vai depender da quantidade de fichas armazenadas no balde. Enquanto houver fichas a consumir, a taxa de saída acompanha a taxa de entrada até um máximo igual à velocidade do enlace C (segundo desenho da figura 3.1b). Quando não há mais fichas a consumir, o tráfego é forçado a obedecer à taxa de geração de fichas T (terceiro desenho da figura 3.1b). Logo, este algoritmo permite que ocorram rajadas de tráfego com taxa superior à T na saída do dispositivo, sendo mais flexível que o balde furado. A duração máxima Δt de uma rajada de taxa C (balde cheio no início da rajada) pode ser calculada através da equação 3.1 [86].

$$B + T.\Delta t = C.\Delta t \Rightarrow \Delta t = B/(C - T) \tag{3.1}$$

Essencialmente, o balde de fichas é um mecanismo mais tolerante quando dá ao tráfego explosivo (em rajadas) mais chances de injetar informação na rede a uma taxa média igual à T. Isto ocorre porque quando o tráfego está a uma taxa R menor que T, a largura de faixa não aproveitada $(T - R)$ vai sendo armazenada em forma de fichas para uso futuro. Desta forma, partindo do princípio que $C > T$ e que nenhuma ficha foi perdida, ainda será possível atingir a taxa média T.

Em redes Diffserv que implementam o serviço assegurado, condicionadores de tráfego baseados em balde de fichas se caracterizam por utilizar como mecanismo de medição um ou mais baldes de fichas. A marcação é função geralmente apenas da disponibilidade de fichas no(s) balde(s), embora algoritmos mais complexos possam ser introduzidos para realizar esta função. Outra característica importante é que o número de baldes de fichas é função do número de níveis utilizados na marcação. Para dois níveis de marcação, um balde é utilizado. Para três níveis de marcação, dois baldes são utilizados.

Marcador Balde de Fichas ou TBM (*Token Bucket Marker*)

Um condicionador de tráfego simples pode ser construído a partir de uma adaptação do algoritmo balde de fichas. Este marcador pode ser utilizado quando o

tamanho para ser transmitido. O excesso de fichas, caso exista, é reservado para consumo futuro.

perfil de tráfego consiste apenas em uma taxa contratada e possivelmente em um grau de absorção de rajadas. Existem apenas dois níveis de marcação (prioridades de descarte) conforme o tráfego obedeça ou não a este perfil de tráfego.

O mecanismo funciona da seguinte forma (figura 3.2). O balde de tamanho CBS (*Committed Burst Size*) é preenchido com fichas a uma taxa CIR (*Committed Information Rate*). Portanto, a cada segundo o balde é incrementado de CIR bytes em fichas até o valor máximo de CBS bytes[9]. Para cada pacote de um microfluxo ou fluxo agregado, um único teste é realizado. Se há um número de fichas suficiente, isto é, o balde tem uma quantidade de bytes em fichas maior ou igual ao tamanho do pacote, o resultado do teste é positivo e o pacote é considerado dentro do perfil ou *in* (*in-profile*). Neste caso, as fichas são consumidas e o pacote é marcado com *codepoint* $AFx1$. Se não há um número de fichas suficiente, o resultado do teste é negativo e o pacote é considerado fora do perfil ou *out* (*out-profile*). Desta vez, nenhuma ficha é consumida e o pacote é marcado com *codepoint* $AFx2$ ou $AFx3$.

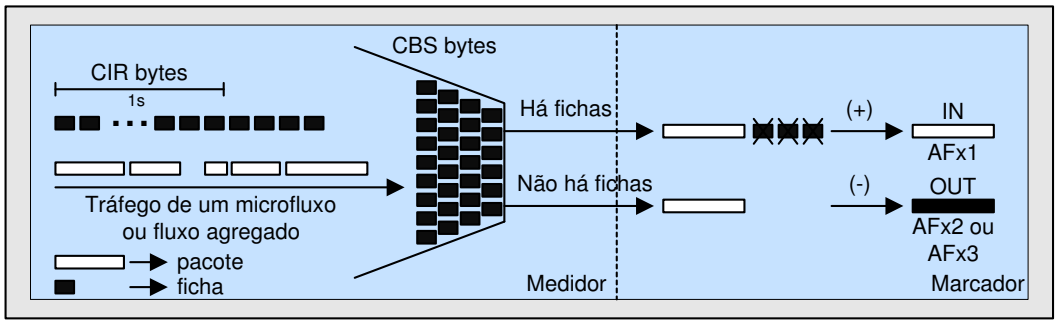


Figura 3.2: Marcador balde de fichas ou TBM.

Marcador de Três Cores de Taxa Única ou SRTCM (*Single Rate Three Color Marker*)

O SRTCM [53] é útil por exemplo em casos onde o comprimento das rajadas do tráfego que está sendo condicionado é mais importante que a taxa de pico. O perfil de tráfego é composto por uma taxa contratada e dois limites para os tamanhos de rajada. O marcador possui três parâmetros: CIR (*Committed Information Rate*), CBS (*Committed Burst Size*) e EBS (*Excess Burst Size*).

[9]Será assumido que todos os marcadores apresentados operam no modo em bytes, por ser mais adequado para a Internet. Portanto, todos os tamanhos de baldes são medidos em bytes e todas as taxas de preenchimento são medidas em bytes por segundo.

O marcador utiliza dois baldes, C e E, e três níveis de prioridade de descarte (figura 3.3). CBS corresponde ao tamanho do balde C e EBS ao tamanho do balde E. A taxa CIR é utilizada para encher o balde C e o balde E. Porém, o balde E só recebe fichas quando o balde C já está cheio. Para cada pacote, um teste é realizado de acordo com o número de fichas no balde C. Se há um número de fichas suficiente, estas são consumidas e o pacote é marcado com *codepoint* $AFx1$ (cor verde). Caso contrário, um novo teste é realizado de acordo com o número de fichas no balde E. Se este for suficiente, as fichas são consumidas e o pacote é marcado com *codepoint* $AFx2$ (cor amarela). Se o número de fichas for insuficiente, nenhuma ficha é consumida e o pacote é marcado com *codepoint* $AFx3$ (cor vermelha).

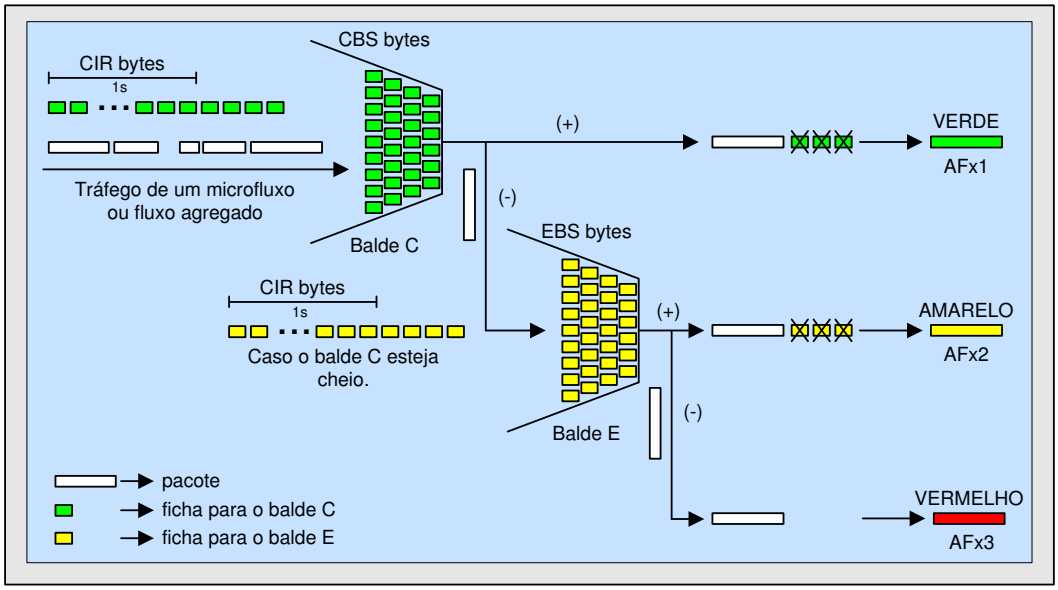


Figura 3.3: Marcador de três cores de taxa única ou SRTCM.

Deste modo, o SRTCM é quase idêntico ao TBM. Porém, em momentos de grande ociosidade do tráfego, o balde C pode encher e a partir daí fichas serão armazenadas no balde E. Com isso, rajadas de maior duração serão permitidas no futuro. Porém, o excesso da rajada, composto pela parte que consome as fichas amarelas do balde E, terá uma probabilidade intermediária entre o tráfego dentro (pacotes verdes) e fora (pacotes vermelhos) do perfil.

Este marcador possui ainda um modo de operação denominado atento às cores (*color-aware*), útil para condicionamento de tráfego pré-marcado. Neste modo, o pacote pode manter ou diminuir a prioridade de encaminhamento. Para mantê-la, o resultado da marcação deve ser igual ou maior do que o da pré-marcação. Caso

Tabela 3.1: SRTCM: resultado da marcação no modo atento às cores.

Pré-marcação	Marcação	Resultado final
Verde	Verde, amarelo ou vermelho	Mesmo da marcação
Amarelo	Amarelo ou verde	Amarelo
Amarelo	Vermelho	Vermelho
Vermelho	Não importa	Vermelho

contrário, prevalece a marcação. A tabela 3.1 descreve esta lógica.

Marcador de Três Cores de Taxa Dupla ou TRTCM (*Two Rate Three Color Marker*)

O TRTCM [54] se destina por exemplo a casos em que o perfil de tráfego especifica não só uma taxa assegurada, mas também uma taxa de pico que precisa ser controlada. O marcador possui portanto quatro parâmetros: CIR (*Committed Information Rate*), CBS (*Committed Burst Size*), PIR (*Peak Information Rate*) e PBS (*Peak Burst Size*).

São utilizados dois baldes, P e C, e três níveis de prioridade de descarte (figura 3.4). PBS corresponde ao tamanho do balde P e CBS ao tamanho do balde C. A taxa PIR é utilizada para encher o balde P e a taxa CIR para encher o balde C. Para cada pacote, um teste é realizado de acordo com o número de fichas no balde P. Se este não for suficiente, o pacote é marcado com *codepoint* $AFx3$ (cor vermelha). Caso contrário, fichas do balde P são consumidas e um novo teste é realizado de acordo com o número de fichas no balde C. Se este não for suficiente, o pacote é marcado com *codepoint* $AFx2$ (cor amarela). Senão, fichas do balde C são consumidas e o pacote é marcado com *codepoint* $AFx1$ (cor verde).

Logo, o TRTCM basicamente compara o tráfego com duas taxas. O tráfego recebe alta prioridade de encaminhamento caso obedeça à taxa assegurada, prioridade intermediária caso se situe entre a taxa assegurada e a de pico, e baixa prioridade caso ultrapasse a taxa de pico.

Assim como o SRTCM, o TRTCM também possui um modo atento à cores que implementa a mesma lógica descrita na Tabela 3.1.

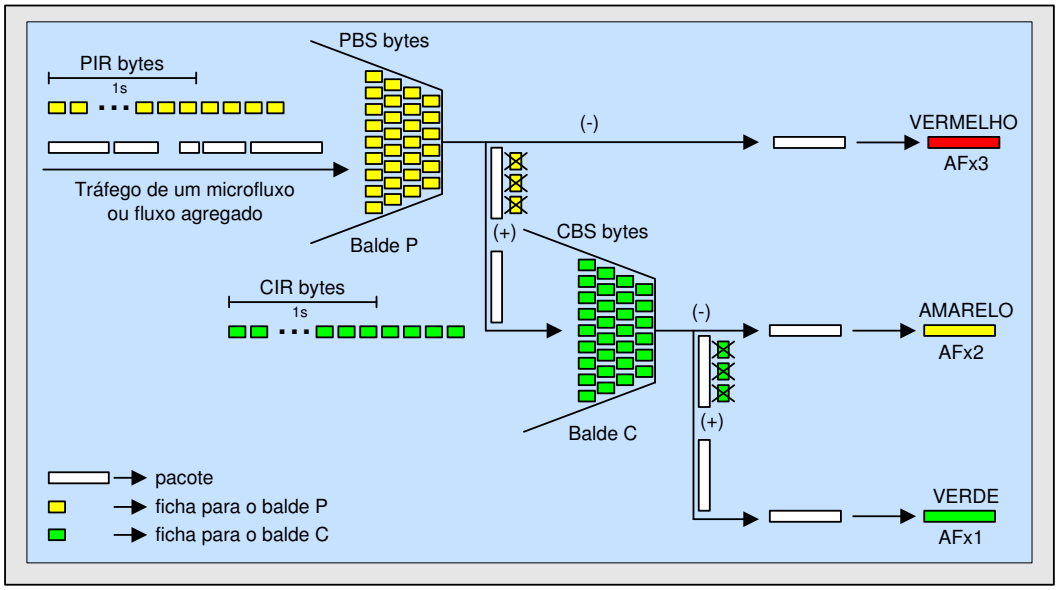


Figura 3.4: Marcador de três cores de taxa dupla ou TRTCM.

3.2.2 Marcadores Baseados em Estimadores de Taxa Média

Na marcação baseada em estimadores de taxa média, o processo de medição é feito através da estimativa da taxa média de informação enviada pelos fluxos individuais ou agregados. A razão para medir a taxa média ao invés da instantânea é acomodar a natureza em rajadas do tráfego TCP. Nos mecanismos anteriores, esta habilidade é viabilizada pelo acúmulo de fichas nos baldes.

Talvez a forma aparentemente mais óbvia para se obter a taxa média de um fluxo de tráfego seja através de uma filtragem passa-baixas da taxa instantânea[10]. Porém, tal abordagem apresenta uma desvantagem. A taxa média decai com a chegada dos pacotes e não com o tempo. Isto faz com que uma conexão TCP rápida (RTT baixo) esqueça a história passada mais rapidamente do que uma conexão TCP lenta (RTT alto), em um mesmo intervalo de tempo. Portanto, algoritmos mais complexos são necessários para medir de forma eficiente a taxa média do tráfego.

Outra característica importante das propostas de marcadores baseados em estimadores de taxa média é a existência de um algoritmo separado para a marcação propriamente dita. Porém, isto não é uma condição obrigatória para que um marcador seja classificado nesta categoria, pois nada impede que a marcação seja função

[10] Qualquer expressão na forma $TM = TM.w + TI.(1-w)$ onde TM é a taxa média, TI a taxa instantânea calculada através da divisão do tamanho do pacote pelo intervalo entre a chegada de pacotes, e w um valor entre 0 e 1 que regula a frequência de corte do filtro.

direta do resultado da medição, assim como no caso dos marcadores baseados em baldes de fichas apresentados anteriormente.

Marcador de Janela Deslizante no Tempo ou TSW (*Time Sliding Window*)

O TSW [52] é a proposta pioneira para marcadores de tráfego baseados em estimadores de taxa média. É útil para perfis de tráfego constituídos de uma taxa assegurada R_T e utiliza dois níveis de marcação. No algoritmo de medição, a taxa média é estimada a cada chegada de um pacote. Mas para evitar o problema descrito anteriormente, a taxa é calculada em uma janela de tamanho finito e portanto decai com o tempo. O algoritmo 3.1 descreve a lógica para a estimativa da taxa média utilizada pelo TSW.

Algoritmo 3.1: Estimativa da taxa média no TSW.

Variáveis:

win_length: tamanho da janela de tempo para a estimativa da taxa média

avg_rate: taxa média estimada

t_front: instante de tempo da chegada do penúltimo pacote

now: instante de tempo da chegada do último pacote

pkt_size: tamanho do último pacote

Inicialmente:

$win\_length \leftarrow constante$

$avg\_rate \leftarrow R_T$

$t\_front \leftarrow 0$

À cada chegada de um pacote:

$bytes\_in\_TSW \leftarrow avg\_rate.win\_length$

$new\_bytes \leftarrow bytes\_in\_TSW + pkt\_size$

$avg\_rate \leftarrow new\_bytes/(now - t\_front + win\_length)$

$t\_front \leftarrow now$

Quanto ao algoritmo de marcação, o objetivo é manter uma conexão ou grupo de conexões TCP oscilando entre $0.66R_T$ e $1.33R_T$, de modo que na média a taxa R_T seja atingida. O tamanho da janela utilizado para a medição é da ordem de um dente de serra TCP de $0.66R_T$ à $1.33R_T$. Os pacotes são marcados como *out* com probabilidade $P = (avg\_rate - R_T)/(avg\_rate)$, quando avg_rate excede R_T. O uso de probabilidade visa espalhar a marcação de pacotes como *out* ao longo do

tempo, reduzindo a chance de *timeout* para o TCP e consequentemente da entrada na fase de início lento (*slow start*)[11] [87]. Para valores de avg_rate abaixo de R_T, todos os pacotes são marcados como *in*.

É importante notar que quando $avg\_rate > R_T$, um aumento no valor de avg_rate aumenta a probabilidade de marcação de pacotes como *out*. Este tipo de solução adaptativa é comum em marcadores baseados em estimadores de taxa média [85] e objetiva punir proporcionalmente ao grau de desobediência ao perfil de tráfego desejado. No entanto, este tipo de marcação probabilística pode favorecer o tráfego não-responsivo [56]. Por exemplo, uma fonte de tráfego CBR (*Constant Bit Rate*) transmitindo a uma taxa maior que a contratada R_T irá obter uma vazão de pacotes *in* maior que este valor. Isto não acontece com o balde de fichas pois a vazão de pacotes *in* é limitada pela taxa de geração de fichas, que deve corresponder à taxa contratada.

Marcador de Janela Deslizante no Tempo de Três Cores TSWTCM (*Time Sliding Window Three Color Marker*)

O TSWTCM [55] é util para perfis de tráfego que especifiquem uma taxa assegurada e uma taxa de pico, assim como o TRTCM. Estes parâmetros são denominados CTR (*Committed Target Rate*) e PTR (*Peak Target Rate*).

No TSWTCM, nenhum algoritmo particular de medição é estipulado. No entanto, recomenda-se que a estimativa da taxa média decaia com o tempo, tal como no TSW. No algoritmo de marcação, semelhante ao do TSW, o crescimento da taxa média aumenta a probabilidade de marcação dos pacotes como *amarelo* se $CTR < avg\_rate \leq PTR$, e como *vermelho* se $avg\_rate > PTR$. O algoritmo 3.2 descreve a lógica completa de marcação para o TSWTCM.

As estratégias de marcação ainda podem ser classificadas quanto ao tipo de informação em que provedor e (ou) usuário se baseiam para marcar os pacotes. A grande maioria dos trabalhos apresentam duas possibilidades: marcação por fluxo e marcação por agregado, conforme o marcador atue em uma conexão ou em um grupo de conexões respectivamente. Estas estratégias, assim como uma terceira intermediária entre elas, serão apresentadas e discutidas no capítulo 4 sob a ótica da justiça entre os fluxos de um mesmo agregado de tráfego.

[11] O controle de congestionamento do protocolo TCP é descrito no apêndice A.

Algoritmo 3.2: Marcação no TSWTCM.

se $avg\_rate \leq CTR$
  o pacote é marcado como verde
senão se $CTR < avg\_rate \leq PTR$
  calcular $P0 = (avg\_rate - CTR)/avg\_rate$
  com probabilidade $P0$ o pacote é marcado como amarelo
  com probabilidade $1 - P0$ o pacote é marcado como verde
senão
  calcular $P1 = (avg\_rate - PTR)/avg\_rate$
  calcular $P2 = (PTR - CTR)/avg\_rate$
  com probabilidade $P1$ o pacote é marcado como vermelho
  com probabilidade $P2$ o pacote é marcado como amarelo
  com probabilidade $1 - (P1 + P2)$ o pacote é marcado como verde

3.3 Disciplinas de Gerenciamento Ativo de Filas

Foi visto que os condicionadores de tráfego marcam os pacotes entrantes no domínio DS, identificando as porções de tráfego dentro e fora do perfil. Para completar a implementação do serviço assegurado, resta aplicar o tratamento diferenciado aos agregados de comportamento, dentro destes domínios. Para realizar esta função, utiliza-se o PHB-AF com duas ou três prioridades de descarte.

Conforme visto na subseção 2.4.4, o PHB-AF deve minimizar congestionamentos de longa duração através de descartes, assim como suportar os congestionamentos de curta duração através do enfileiramento dos pacotes. Para atingir este objetivo, deve-se utilizar disciplinas de filas que operem em cima do nível médio de congestionamento. Isto significa dizer que os descartes só serão efetuados quando o nível médio de ocupação da fila for alto, caracterizando um congestionamento de longa duração. Curtas rajadas serão absorvidas sem descartes caso não provoquem alterações significantes no tamanho médio da fila.

O PHB-AF também deve ser insensível às características de curta duração de um fluxo. Isto é, fluxos de diferentes formatos de rajadas curtas mas mesmas taxas de transmissão a longo prazo devem ter as mesmas probabilidades de perda de pacotes. Além disso, congestionamentos devem ser sinalizados de forma gradual para evitar oscilações. A obtenção destas propriedades pode ser facilitada através do uso de uma função aleatória de descartes, espalhando as perdas de pacotes entre as conexões e no tempo.

Um algoritmo com tais características é o RED (*Random Early Detection*) [88], que será visto com mais detalhes na subseção 3.3.1 a seguir. O RED é um tipo de disciplina de gerenciamento ativo de filas. Estes algoritmos têm como principais objetivos:

- reduzir o tamanho médio da filas nos roteadores e consequentemente reduzir o retardo fim-a-fim dos pacotes;
- fazer com que os recursos da rede sejam utilizados mais eficientemente, reduzindo o número de perdas que ocorrem quando as filas se enchem por completo.

O desenvolvimento de disciplinas de gerenciamento ativo de filas é necessário porque o controle de congestionamento do TCP não é suficiente para impedir que tais situações ocorram. Além disso, nem todo o tráfego da Internet é responsivo. Portanto, algum mecanismo interno à rede deve suplantar esta necessidade.

3.3.1 RED (*Random Early Detection*)

O RED [88] objetiva principalmente evitar congestionamentos longos, controlando o nível médio de ocupação das filas. Sua proposta se baseia no fato de que o roteador é o elemento mais apto a detetar uma situação de congestionamento, sem o risco de confundir os retardos de propagação e transmissão com o retardo de enfileiramento. Partindo deste princípio, os roteadores podem sinalizar situações de congestionamento às fontes, de forma a controlar a quantidade de carga na rede. Portanto, o RED é direcionado para redes onde o protocolo de transporte responde às indicações de congestionamento, ou, mais especificamente, onde uma única marcação ou descarte de um pacote é suficiente para sinalizar tal situação.

O TCP é um protocolo com esta propriedade pois nas suas implementações mais comuns[12], a janela de transmissão é reduzida na ocorrência de *timeouts* e no recebimento de reconhecimentos duplicados. Isto ocorre porque partindo do princípio de que os temporizadores estão bem ajustados, isto é, *timeouts* não são disparados precipitadamente, ambos os eventos são causados por descartes. Por sua vez, os descartes na Internet ocorrem predominantemente devido ao esgotamento dos recursos[13] [89]. Logo, o TCP considera *timeouts* e reconhecimentos duplicados

[12]Quatro das implementações mais utilizadas do TCP, Tahoe [89], Reno [90], New Reno [91] e SACK [92], encontram-se descritas no apêndice A.

[13]Razões menos frequentes incluem perda da rota (TTL - *Time To Live*) e corrupção do conteúdo (CRC - *Cyclic Redundancy Check*).

indicadores de congestionamento. Isto faz com que o RED se torne adequado para ser utilizado em conjunto com o TCP, tendo em vista que um único descarte proposital é o sinal que este último precisa para reagir ao congestionamento.

O RED possui ainda dois objetivos adicionais:

- evitar o sincronismo global[14] e a rejeição ao tráfego em rajadas, os quais ocorrem em filas FIFO convencionais (*Tail Drop*);
- impor um limite superior ao tamanho médio da fila mesmo na ausência de protocolos de transporte que respondem ao congestionamento.

O algoritmo funciona da seguinte maneira. Quando a fila está cheia, o RED monitora o tamanho médio da fila, calculando-o através de uma filtragem passa-baixas do seu valor instantâneo. A equação 3.2 mostra como é feito este cálculo, sendo avg e q os tamanhos médio e atual da fila, e w_q um fator ponderador que controla a frequência de corte do filtro.

$$avg = (1 - w_q).avg + w_q.q \tag{3.2}$$

Quando a fila está vazia, avg decai gradualmente em função do seu tempo de ociosidade, conforme a equação 3.3. O valor de m equivale ao número de pacotes que poderiam ser transmitidos durante um período de ociosidade, e pode ser calculado pela razão entre este intervalo de tempo e o tempo de transmissão de um pacote de tamanho típico.

$$avg = (1 - w_q)^m.avg \tag{3.3}$$

À cada chegada de um novo pacote, é calculado o tamanho médio da fila avg, o qual é comparado a dois outros parâmetros: min_{th} e max_{th}. Quando $avg < min_{th}$, o algoritmo está no modo de operação normal e o pacote é enfileirado. Quando $min_{th} \leq avg < max_{th}$, a fila entra no modo de prevenção ao congestionamento. Nesta fase, o algoritmo calcula uma probabilidade de descarte para o pacote que chegou. Esta probabilidade cresce com o nível de congestionamento avg até um valor máximo max_p, conforme avg tende à max_{th}. Quanto maior max_p, mais a probabilidade de descarte acompanha as variações no nível médio da fila, tornando o algoritmo mais agressivo. Portanto, o RED não espera que a fila encha para descartar os pacotes. Ao contrário, ele descarta os pacotes mais cedo a fim de sinalizar

[14] Fenômeno no qual as conexões TCP reduzem e aumentam as suas janelas de transmissão em sincronia, causando instabilidade na rede (caráter oscilatório).

para as fontes de tráfego um princípio de congestionamento, e desta forma controlar o tamanho médio da fila (primeiro objetivo). O uso da função aleatória de descartes visa espalhá-los entre as conexões e no tempo, a fim de não punir o tráfego em rajadas e evitar o sincronismo global, respectivamente (segundo objetivo). Finalmente, quando $avg \geq max_{th}$, o RED entra na fase de controle de congestionamento, onde todos os pacotes são descartados. Neste caso trata-se de um congestionamento longo, o que deve ser evitado. Para isso, o RED impõe um limite incondicional ao tamanho médio da fila (terceiro objetivo). O algoritmo original completo do RED encontra-se no apêndice B. Sua representação gráfica é ilustrada na figura 3.5. Vale notar que o RED pode funcionar no modo em bytes ou em pacotes.

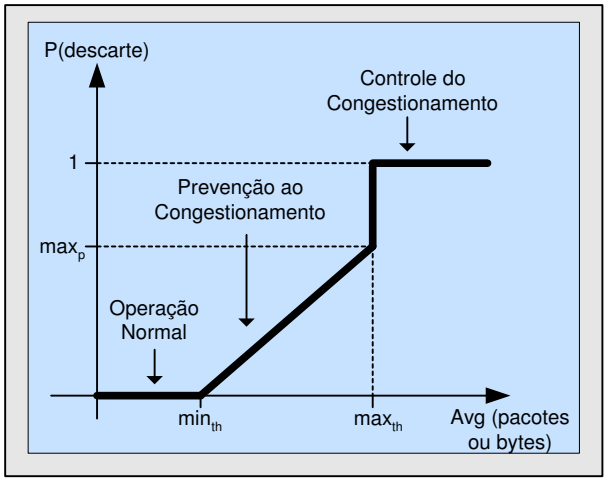


Figura 3.5: Representação gráfica do RED.

Apesar do seu uso ter sido recomendado pelo IRTF (*Internet Research Task Force*) num documento conhecido como "RED Manifesto" [93], a eficácia do RED ainda tem sido motivo de muita pesquisa. MAY *et al.* [94] realizaram experimentos com uma mistura de tráfego HTTP, FTP e UDP. Foi observado que para filas de tamanho pequeno, o RED tem desempenho equivalente a uma fila *Tail Drop* em termos de vazão e de retardo. No entanto, o RED se mostrou eficiente no controle do nível de ocupação de filas maiores. CHRISTIANSEN *et al.* [95] compararam o desempenho do RED com o da fila *Tail Drop* em termos de tempo de resposta ao usuário, para tráfego somente WWW (HTTP). Os resultados obtidos mostraram uma grande equivalência de desempenho entre os dois mecanismos, e foi apontada a necessidade de mais pesquisa para justificar a implantação do RED na Internet. BONALD *et al.* [96] avaliaram o RED através de modelos analíticos e simulações,

e os resultados apontaram algumas deficiências no seu desempenho. Em primeiro lugar, a eliminação da rejeição ao tráfego em rajadas ocorre através do aumento das perdas para o tráfego suave, e não através da diminuição das perdas para o tráfego TCP. Em segundo lugar, o RED causa o aumento do número de perdas consecutivas em relação à fila *Tail Drop*, contribuindo para o sincronismo entre as fontes TCP. Por último, apesar do RED melhorar o retardo fim-a-fim através do controle do nível médio de ocupação das filas, ele aumenta o *jitter* para fontes de tráfego suave. Isto comprometeria o uso do RED em aplicações de tráfego suave, como voz sobre IP. No entanto, mais recentemente, ABOUZEID e ROY [97] criticaram as premissas do modelo utilizado por BONALD *et al.* e estudaram o comportamento do RED em regime permanente para o tráfego TCP. Os resultados mostraram a eficácia do RED na diminuição da rejeição ao tráfego em rajadas.

Formas variantes do RED também foram propostas com o intuito de eliminar algumas de suas deficiências. LIN e MORRIS [98] observaram que o RED impõe as mesmas taxas de perda aos diferentes fluxos, independentemente do espaço ocupado por cada um destes na fila. A partir disso, propuseram um algoritmo denominado FRED (*Flow Random Early Drop*), onde a taxa de perda para cada fluxo é proporcional à sua ocupação na fila, visando o aumento da justiça. Com este mesmo propósito, ANJUM e TASSIULAS [99] propuseram um outro algoritmo denominado BRED (*Balanced RED*). OTT *et al.* [100] observaram que o impacto de uma perda na diminuição da taxa de chegada de pacotes do tráfego TCP varia com a quantidade e a natureza das conexões. Este impacto é alto quando a ocupação da fila é dominada por poucas conexões com grandes janelas de congestionamento, e baixo quando é dominado por muitas conexões com janelas pequenas. Isto porque reduzir à metade uma janela grande provoca maior redução na taxa de transmissão. OTT *et al.* propuseram então um algoritmo denominado SRED (*Stabilized RED*), no qual as probabilidades de descarte são proporcionais ao número estimado de fluxos ativos. Com isso, o nível de ocupação da fila flutua em torno de um patamar pré-estabelecido, independentemente deste valor. De forma semelhante, FENG *et al.* [101] propuseram o RED adaptativo, onde a agressividade do algoritmo é controlada pela variação do parâmetro max_p de acordo o valor de avg. Porém, vale notar que um nível médio de ocupação maior nem sempre corresponde a um maior número de fluxos ativos.

Outra questão importante tanto para o RED como para as suas formas variantes, é o ajuste adequado dos seus parâmetros. Um estudo exato sobre como obter

um ajuste ótimo para uma determinada situação ainda não foi obtido [94, 96, 97]. Porém, algumas recomendações foram feitas [102, 103] a este respeito através de estudos que utilizaram modelos analíticos. FLOYD e JACOBSON [88] fizeram uma série de considerações sobre este assunto[15]. Porém, estas dicas de configuração vêm sofrendo alterações de acordo com novos estudos [104].

Conclui-se então que há uma controvérsia a respeito dos reais benefícios da implantação do RED na Internet. Apesar disto, este mecanismo tem sido largamente utilizado na pesquisa do serviço assegurado, por preencher as recomendações do PHB-AF a respeito de congestionamentos curtos e longos, na medida em que opera em cima do tamanho médio da fila e efetua o descarte aleatório de pacotes.

3.3.2 RED no Serviço Assegurado

Apesar de preencher alguns requisitos do PHB-AF, o RED não possui nenhuma lógica para diferenciar as prioridades de descarte dos pacotes em função da marcação. O RIO (*RED with In and Out*) [52] foi a proposta pioneira neste sentido e trabalha com dois níveis de prioridade (duas cores). O algoritmo funciona da mesma maneira que o RED mas com dois conjuntos distintos de parâmetros: um para os pacotes *in* (min_{th}_in, max_{th}_in, max_p_in) e outro para os pacotes *out* (min_{th}_out, max_{th}_out, max_p_out). Além disso, duas estimativas para o tamanho médio da fila são calculadas: avg_in, levando em conta apenas os pacotes *in*, e avg_total, considerando todos os pacotes. Na chegada de um pacote *in*, o algoritmo RED é executado com avg_in e os parâmetros min_{th}_in, max_{th}_in e max_p_in. Para os pacotes *out*, são utilizados avg_total, min_{th}_out, max_{th}_out e max_p_out.

De acordo com o PHB-AF, o descarte deve ser mais agressivo para os pacotes *out*. Isto já acontece na medida em que $avg\_total \geq avg\_in$. Contudo, o RIO implementa ainda três formas adicionais de diferenciar a probabilidade de descarte entre pacotes de cores distintas. Primeiro, min_{th}_out é menor que min_{th}_in para que os pacotes *out* comecem a ser descartados mais cedo. Segundo, max_p_out é maior que max_p_in para que o descarte dos pacotes *out* seja mais agressivo do que o dos pacotes *in* na fase de prevenção de congestionamento. Por último, max_{th}_out é muito menor que max_{th}_in para que a fase de controle de congestionamento ocorra bem mais cedo para os pacotes *out*. Se além destas condições, max_{th}_out for menor ou igual à min_{th}_in, então a fase de prevenção de congestionamento para os pacotes *in* só começará quando os pacotes *out* já estiverem em controle

[15] Estas recomendações estão resumidas no apêndice B.

de congestionamento. Isto significa que o RIO descartará os pacotes *out* primeiro quando detectar um congestionamento. Se esta situação persistir, ele descartará então todos os pacotes *out*. Somente se houver um grande número de pacotes *in*, o RIO começará a descartá-los. Porém, isto não deve acontecer se a rede for bem provisionada, já que os pacotes *in* correspondem à porção de tráfego dentro do perfil. A figura 3.6 mostra a representação gráfica do RIO com todas as condições anteriores e $max_{th}\_out = min_{th}\_in$.

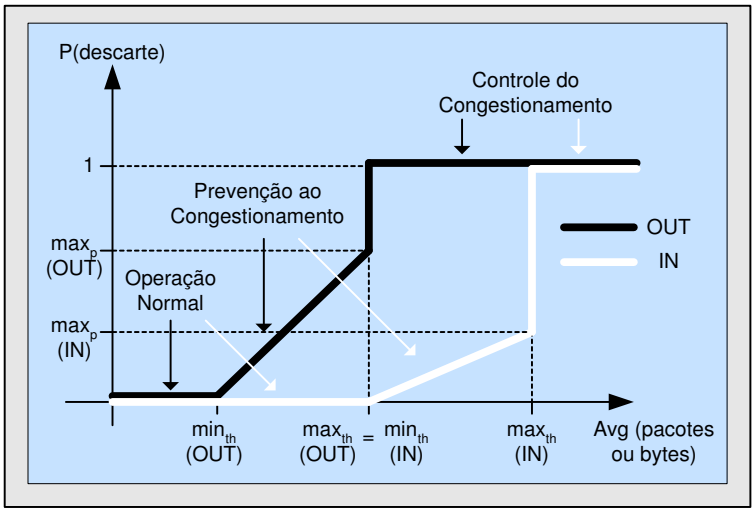


Figura 3.6: Representação gráfica do RIO.

O RIO pode ser generalizado para casos onde três cores são utilizadas, conforme mostra a figura 3.7. Neste caso seriam necessários três conjuntos de parâmetros, um para cada cor. Além disso, três tamanhos médios de fila seriam calculados utilizando somente os pacotes verdes, os pacotes verdes e amarelos, e por último todos os pacotes. Estes valores seriam utilizados pelos três algoritmos RED no tratamento dos pacotes verdes, amarelos e vermelhos, respectivamente.

O RIO não é única forma de implementar diferentes probabilidades de descarte entre agregados de comportamento para o serviço assegurado. Outras variações existem e podem ser aplicadas também às formas variantes do RED citadas anteriormente [105]. De acordo com as opções utilizadas na diferenciação, estas formas podem ser classificadas da seguinte maneira [61]:

- Única Média Único Limiar (*Single Average Single Threshold* - SAST): neste caso, existe apenas um tamanho médio de fila calculado, e os limiares min_{th} e max_{th} são os mesmos para todas as cores. Logo, a única forma de diferenciação

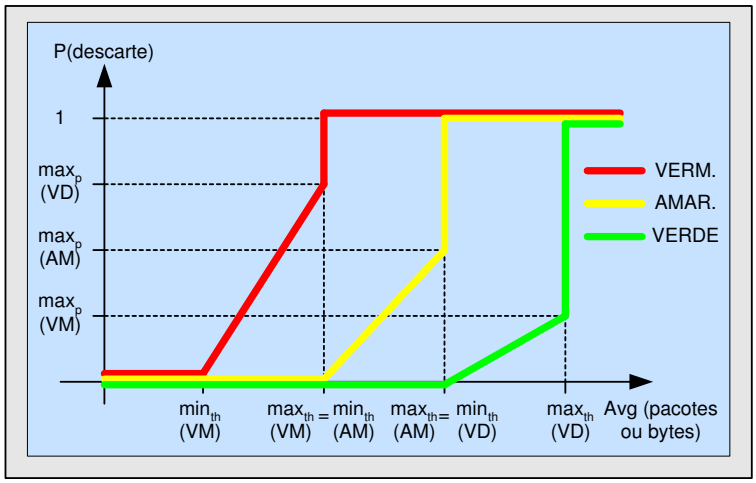


Figura 3.7: Generalização do RIO para três cores.

é atribuindo valores distintos para max_p, de acordo com a cor [83, 84]. Caso contrário, o algoritmo corresponderá ao RED tradicional sem diferenciação alguma entre as cores.

- Única Média Múltiplos Limiares (*Single Average Multiple Thresholds* - SAMT): neste caso, uma única estimativa para o tamanho médio da fila é calculada contabilizando todos os pacotes. Contudo, pacotes de cores diferentes possuem limiares de descarte distintos [61, 81].

- Múltiplas Médias Único Limiar (*Multiple Average Single Threshold* - MAST): aqui os limiares são os mesmos para todas as cores. Porém, uma estimativa para o tamanho médio da fila é feita para cada cor [83, 84].

- Múltiplas Médias Múltiplos Limiares (*Multiple Average Multiple Thresholds* - MAMT): este caso é o mais genérico, onde a diferenciação é implementada através de limiares e médias distintas. O RIO pertence a esta categoria.

Nas variações com múltiplas médias (MAST e MAMT), duas abordagens distintas podem ser utilizadas. A primeira é a descrita no RIO, onde o cálculo das médias contabiliza os pacotes da cor correspondente, juntamente com os das cores inferiores, caso existam. A segunda contabiliza apenas os pacotes da própria cor [83, 84]. Argumentos a favor da primeira abordagem incluem o fato dos pacotes fora do perfil constituirem um tráfego oportunista, para o qual geralmente não se tem uma noção exata da quantidade adequada [52]. Isto dificulta a escolha dos parâmetros para este

tráfego na variação MAMT. Além disso, com as médias acumuladas, consegue-se ter uma noção clara da ocupação total da fila, independentemente das proporções de tráfego dentro e fora do perfil.

A seguir, o algoritmo FRED será visto de forma mais detalhada, devido ao seu emprego em marcadores de tráfego que serão vistos no capítulo 4.

3.3.3 FRED (*Flow Random Early Drop*)

O uso de uma função aleatória de descartes faz com que o RED imponha a mesma taxa de perda para os fluxos que compartilham a largura de faixa do enlace de transmissão. Isto é, o número de descartes para cada fluxo é proporcional à sua ocupação na fila. Esta política causa desigualdades no compartilhamento da largura de faixa por fluxos de diferentes características [98], devido aos seguintes fatores:

- conexões TCP mais lentas (maiores RTTs) reagem mais lentamente aos descartes. Logo, a perda de um único pacote periodicamente para estas conexões pode fazer com que elas não atinjam as suas porções justas, mesmo que conexões TCP mais rápidas tenham mais perdas. Além disso, conexões TCP com janelas maiores são mais tolerantes aos descartes porque são menos suscetíveis a *timeouts*;

- o descarte aleatório pode fazer com que o RED elimine um pacote de um fluxo que possui nenhum ou poucos pacotes na fila. Isto pode acontecer inclusive para conexões simétricas (mesmo RTT) e causa uma injustiça temporária, significante principalmente em casos de tráfego composto por conexões de curta duração[16];

- a aceitação de um pacote para uma dada conexão contribui para o 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, fazendo com que os fluxos adaptativos não consigam atingir as suas porções justas.

O algoritmo funciona basicamente da mesma forma que o RED, mas com algumas modificações. Primeiro, são introduzidos dois novos limiares, min_q e max_q,

[16] ABOUZEID e ROY [97] mostraram que, em regime permanente, o RED tende a compensar estas desigualdades, proporcionando a mesma vazão para as conexões, inclusive em casos de conexões de RTTs diferentes.

correspondendo aos números mínimo e máximo de pacotes que cada fluxo pode ter na fila, respectivamente. Segundo, um estado é armazenado para cada fluxo que possui pelo menos um pacote na fila (fluxo ativo) contendo: o número de pacotes na fila para determinado fluxo i, $qlen_i$, e o número de vezes que um fluxo i tenta ter mais do que max_q pacotes na fila, $strike_i$. Por último, uma nova variável, $avgcq$, armazena o número médio de pacotes por fluxo ativo.

De modo a combater as três deficiências do RED descritas anteriormente, o FRED utiliza estes novos parâmetros e variáveis da seguinte forma. Em primeiro lugar, para proteger fluxos com menores janelas (mais lentos), o FRED permite que cada conexão enfileire até min_q pacotes na fila, desde que o algoritmo não esteja na fase de controle de congestionamento ($avg > max_{th}$). Em segundo lugar, o FRED restringe os descartes aleatórios para os fluxos que possuem mais pacotes do que $avgcq$ ou min_q, o que for maior. Em terceiro lugar, 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 em exceder este limite. Finalmente, os fluxos com altos valores para $strike_i$ ficam proibidos de armazenar mais do que $avgcq$ pacotes na fila. Deste modo, o FRED permite que fluxos adaptativos enviem rajadas de tráfego, ao mesmo tempo que proíbe que fluxos não responsivos abusem do espaço na fila.

O algoritmo FRED completo encontra-se no apêndice B, juntamente com algumas considerações sobre o ajuste dos seus parâmetros. Além de poder funcionar nos modos em bytes e em pacotes, assim como o RED, o FRED ainda possui uma forma variante para um número excessivo de fluxos.

Capítulo 4

Justiça no Serviço Assegurado

Este capítulo divide-se basicamente em duas partes. A primeira aborda o problema da justiça entre os fluxos que constituem o tráfego assegurado. De forma a facilitar seu entendimento e análise, este problema será estruturado e dividido em duas situações distintas de menor complexidade: justiça entre fluxos agregados correspondentes a diferentes perfis de tráfego e justiça entre os fluxos que compõem um mesmo fluxo agregado. Em seguida, as suas principais causas serão apresentadas e discutidas.

A segunda parte aborda especificamente a justiça entre os fluxos de um mesmo tráfego agregado, evidenciando as vantagens do uso de condicionadores de tráfego como solução para este problema. A seguir, as estratégias de marcação existentes serão apresentadas e comparadas. O capítulo termina com a descrição do marcador FM (*Fair Marker*) [62], uma proposta para fornecer justiça entre fluxos de um mesmo tráfego agregado. Finalmente, uma extensão ao FM denominada TCFM (*Three Color Fair Marker*) [63] será proposta de forma a suprir suas deficiências.

4.1 Formulação do Problema

O problema da justiça no serviço assegurado consiste na divisão dos recursos disponíveis entre diferentes fluxos ou microfluxos de tráfego, de forma não desejável segundo os interesses do provedor e (ou) do cliente[1]. Tendo em vista que no serviço assegurado a vazão obtida é a métrica de desempenho fundamental, os recursos

[1]Isto é, quem oferece e quem o utiliza o serviço, respectivamente. Estes dois papéis podem se confundir, por exemplo, quando uma política de diferenciação de serviços for implantada internamente em uma única instituição.

supracitados correspondem principalmente à largura de faixa e espaço em *buffer*.

Este assunto tem sido abordado através de cenários bastante heterogêneos dificultando o seu entendimento de uma forma global. É necessário portanto estruturá-lo, identificando as diferentes situações em que faz sentido pensar em justiça na divisão dos recursos de uma rede que implementa o serviço assegurado. Além disso, deve ficar claro o que representa um cenário justo em cada um destes casos.

Conforme visto na subseção 2.3.3, os classificadores atuam sobre fluxos agregados de tráfego (figura 4.1). Deste modo, o tráfego de melhor esforço (ME) é separado do tráfego a ser condicionado e marcado apropriadamente, em função dos serviços diferenciados contratados junto ao provedor. No caso mais genérico, um mesmo cliente contrata mais de um serviço. Isto faz com que fluxos de menor nível de agregação, ou seja, menor número de microfluxos, sejam condicionados de acordo com os seus respectivos perfis de tráfego. Além disso, um escalonador é necessário para o compartilhamento do enlace de saída.

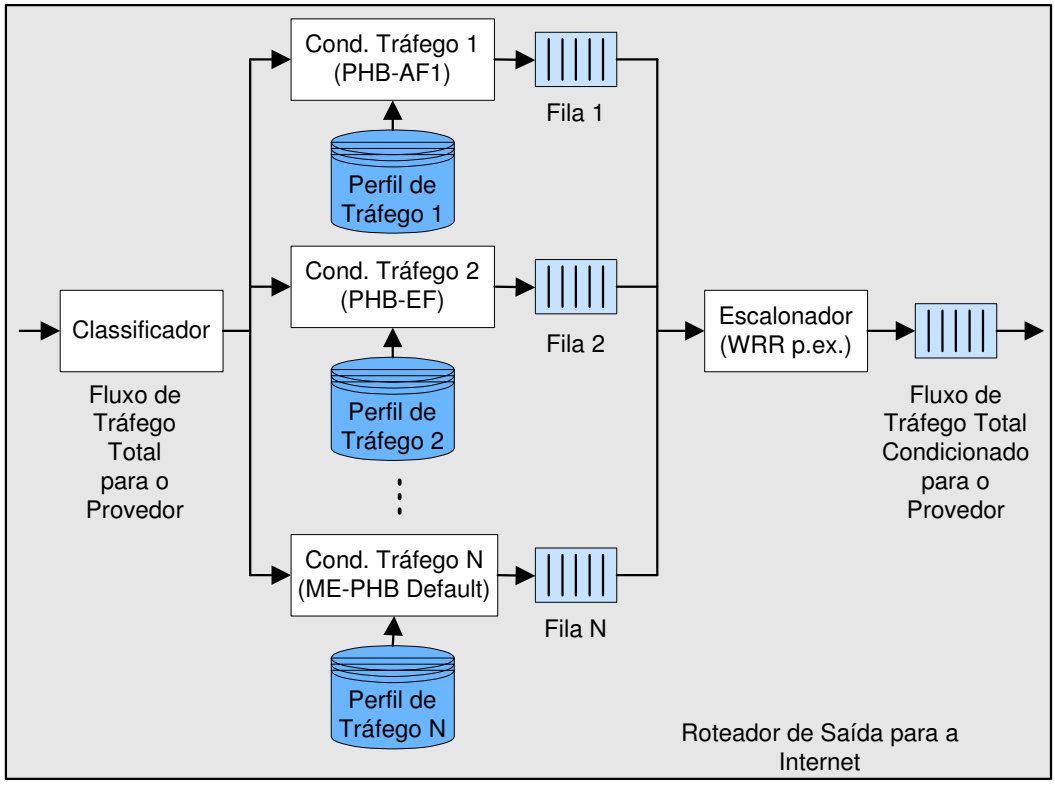


Figura 4.1: Diferentes perfis de tráfego em um mesmo cliente.

Pode-se dizer então que todo o tráfego que entra e sai de um provedor é formado pela mistura destes fluxos agregados associados a diferentes perfis de tráfego.

Normalmente, estes fluxos vão pertencer a diferentes clientes e apresentar diferentes níveis de agregação. Além disso, os diversos microfluxos que os compõem disputam os recursos do provedor. Este compartilhamento pode ser bastante complexo em função da diversidade das rotas entre fonte e destino para cada microfluxo, e da própria complexidade da arquitetura da rede do provedor. No entanto, em muitos casos, não é difícil identificar como os recursos estão sendo compartilhados. Um exemplo típico é ilustrado na figura 4.2, onde vários clientes se ligam a um ISP. Neste caso, largura de faixa e espaço em *buffer* no enlace de acesso à Internet são compartilhados pelos diversos fluxos de tráfego dos usuários.

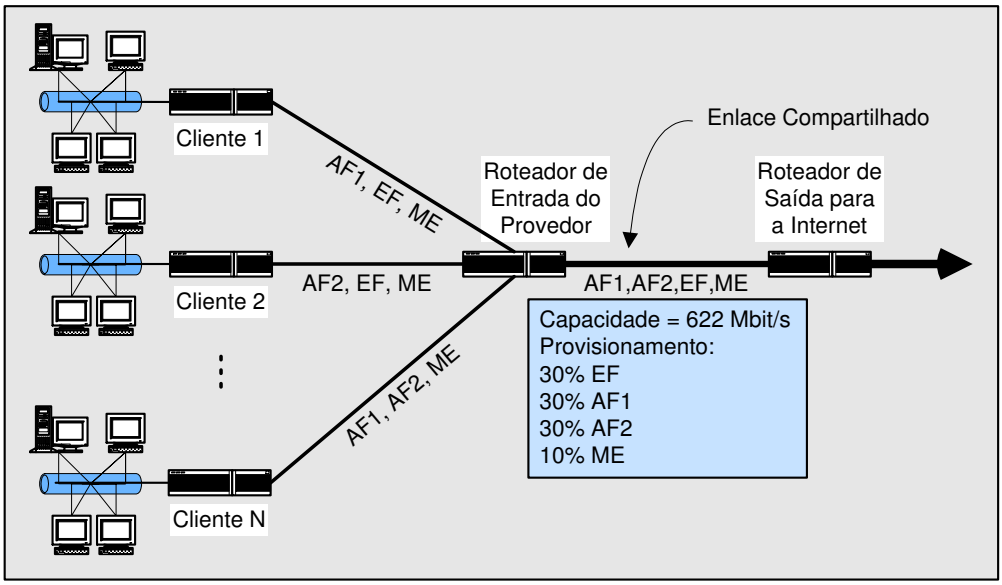


Figura 4.2: Compartilhamento de recursos entre diversos fluxos em um ISP.

Vale notar que nas figuras 4.1 e 4.2 foram representados outros PHBs além do PHB-AF, ilustrando uma situação mais genérica em que um mesmo provedor suporta mais de um PHB. Nestes casos, os recursos serão distribuídos de forma independente entre os PHBs e grupos de PHBs (classes AF distintas por exemplo) [12], através do uso de disciplinas de escalonamento tais como WRR. Para a análise do serviço assegurado, isto permite que a rede do provedor possa ser visualizada em subconjuntos formados pelos recursos reservados a cada classe AF de PHBs. Tal aproximação é justificada pelo baixo impacto da presença de outros PHBs no PHB-AF, em termos de vazão obtida[2]. A figura 4.3 mostra a mesma rede da figura 4.2 apenas com os recursos reservados à classe AF1 do PHB-AF.

[2]O mesmo pode não ocorrer para o PHB-EF, onde retardo e *jitter* são importantes.

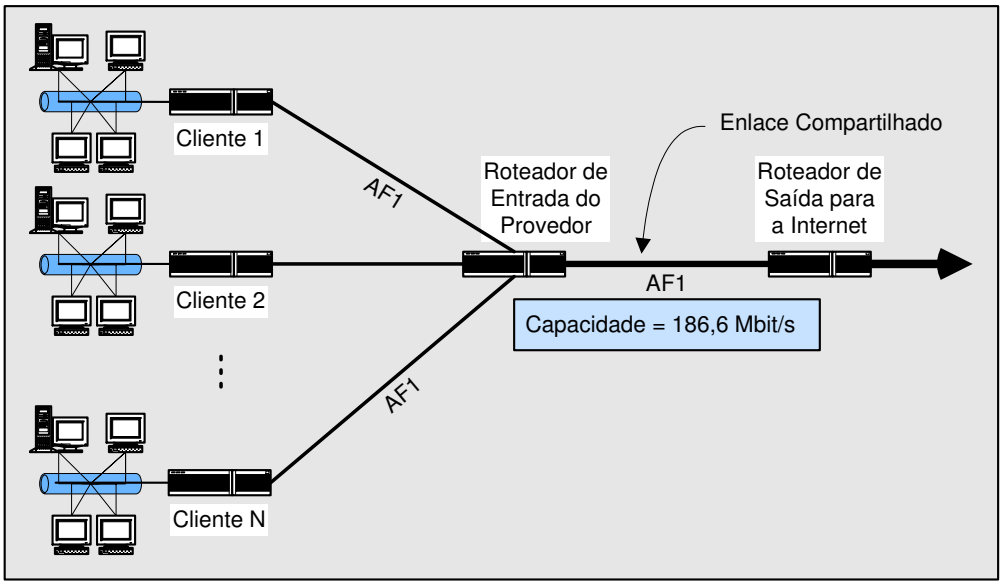


Figura 4.3: Subconjunto de recursos destinados à classe AF1.

Partindo da premissa de que é possível identificar como os recursos do provedor estão sendo compartilhados pelos diferentes fluxos agregados de uma determinada classe AF, deve-se garantir que alguns destes fluxos não sejam privilegiados em detrimento de outros, sem deixar de levar em conta as diferenças entre os respectivos perfis de tráfego. Além disso, pode ser importante a ausência de distorções na forma com que os diferentes microfluxos, usuários, aplicações ou subredes dividem os recursos obtidos por cada fluxo agregado. Sendo assim, para cada classe AF em questão, o problema da justiça pode ser dividido em dois problemas distintos e independentes:

- justiça entre os fluxos agregados que correspondem aos diferentes perfis de tráfego.

  Dentro deste contexto, a definição de justiça para este tipo de compartilhamento ainda vai depender da forma de aprovisionamento dos recursos da rede. No caso de uma rede superdimensionada, um cenário justo será aquele em que todos os fluxos agregados atingem as suas taxas contratadas. Isto é, independentemente das diferenças entre os perfis de tráfego aos quais estes fluxos estão associados, todos devem obter o nível de serviço pelo qual estão pagando, já que há recursos suficientes para que esta condição seja satisfeita. Quantos aos recursos excedentes, a questão admite três soluções mais ime-

diatas. A primeira delas considera mais justo dividir os recursos excedentes proporcionalmente aos valores das taxas asseguradas de cada perfil de tráfego [60, 106, 107]. Portanto, um cliente que contrata uma taxa assegurada de 1 Mbit/s deverá atingir uma vazão adicional duas vezes maior do que um outro que contrada 500 kbps, desde que a rede esteja superdimensionada. As demais soluções partem do princípio de que os recursos excedentes, por não estarem sendo contratados, não devem ter correlação com os perfis de tráfego. No entanto, esta mesma premissa conduz a duas abordagens distintas: dividir os recursos igualmente [80] ou até mesmo não aplicar qualquer política de compartilhamento [58].

No caso de redes subdimensionadas, não há como fazer com que todos os fluxos agregados atinjam as suas taxas asseguradas[3]. Neste caso, o mais justo será distribuir os recursos proporcionalmente aos perfis de tráfego contratados [83, 84], causando uma degradação suave no nível de serviço de cada cliente. No entanto, assim como no caso anterior, outras abordagens também são defendidas, como por exemplo a de não aplicar nenhuma política específica [58]. Este ponto de vista ganha força através do argumento de que o cliente é mais sensível ao não cumprimento do serviço, independentemente do nível da defasagem.

- justiça entre os fluxos que compõem um fluxo agregado correspondente a um perfil de tráfego. Estes "subfluxos" de menor nível de agregação não correspondem necessariamente a microfluxos (caso potencialmente mais comum), podendo representar também o tráfego de um grupo de usuários, estações de trabalho, aplicações específicas e até subredes de uma empresa ou campus.

  A definição de justiça para este caso é mais simples, pois parte-se da premissa de que no SLA é definido apenas um perfil de tráfego para o fluxo agregado. Logo, se em termos contratuais não há nenhuma diferenciação explícita sobre o nível de serviço que deve ser obtido pelos fluxos que compõem o tráfego total, a princípio deseja-se que os recursos do provedor sejam repartidos

[3]Embora situações como esta não devam ocorrer no estado de operação normal de uma rede, seu estudo é justificado pela tendência em se maximizar o lucro através do uso de multiplexação estatística, reduzindo-se a margem de superdimensionamento [58]. Deste modo, mesmo com o desenvolvimento de ferramentas que ajudem no aprovisionamento e gerenciamento de redes Diff-Serv, é prudente assumir surtos periódicos e limitados de tráfego assegurado (prioritário) acima da capacidade máxima da rede.

Tabela 4.1: Compartilhamento justo de recursos no serviço assegurado.

Aprovisionamento	Subdimensionado	Superdimensionado	
Recursos	Contratados	Contratados	Excedentes
Entre fluxos agregados	Proporcional aos perfis de tráfego contratados	De modo a atender todos os perfis de tráfego contratados	Proporcional aos perfis de tráfego contratados ou igual para todos
Entre fluxos que compõem um fluxo agregado	Igual para todos os fluxos	Igual para todos os fluxos	Igual para todos os fluxos

igualmente entre estes fluxos [108, 109]. Este ponto de vista independe do aprovisionamento da rede, valendo portanto para os casos subdimensionado e superdimensionado, no que se refere ao compartilhamento das taxas assegurada e excedente. Contudo, nada impede que o cliente adote uma política interna para a divisão dos recursos contratados, ficando o provedor responsável apenas pelo policiamento do tráfego total na entrada do seu domínio (subseção 4.3.4).

A tabela 4.1 resume de forma estruturada o problema da justiça no serviço assegurado. Para cada uma das situações discutidas acima, são descritos os cenários justos mais comuns, em função dos recursos que estão sendo compartilhados e do aprovisionamento da rede. É importante notar que estes não são critérios únicos, cabendo ao provedor definir o que deve ser feito quanto a forma de repartir os recursos, levando ou não em consideração os anseios de seus clientes. De qualquer forma, esta é uma decisão de negócios que não deve ser influenciada por limitações técnicas [107]. Logo, qualquer que seja a alternativa adotada por um provedor, ele deve ter os componentes necessários para implementá-la.

Uma vez definida a política a ser adotada, dois termos importantes devem ser bem quantificados e diferenciados para que se possa avaliar a justiça: taxa reservada (assegurada ou contratada) e taxa alvo (*target rate*). A taxa reservada corresponde ao valor contratado, enquanto que a taxa alvo é a taxa reservada mais a porção excedente, em função da política escolhida. Mais formalmente, se C é a capacidade total de transmissão de um enlace de acesso ou de gargalo compartilhado por vários fluxos em um provedor, e R_i é a taxa contratada que define um perfil de tráfego i, então a largura de faixa total assegurada R pode ser calculada pela equação 4.1,

onde N é o número total de perfis de tráfego. Consequentemente, a largura de faixa excedente total E é obtida através da equação 4.2.

$$R = \sum_{i=1}^{N} R_i \tag{4.1}$$

$$E = C - R \tag{4.2}$$

De acordo com os critérios de justiça da tabela 4.1, cada fluxo agregado i deve obter, além de sua taxa reservada R_i, uma taxa excedente E_i definida conforme a equação 4.3. Já para o critério de compartilhamento igualitário dos recursos excedentes, a taxa excedente para cada fluxo é definida de forma diferente, conforme a equação 4.4.

$$E_i = E\left(\frac{R_i}{R}\right) \tag{4.3}$$

$$E_i = \frac{E}{N} \tag{4.4}$$

Independentemente dos critérios acima no entanto, a taxa alvo T_i será obtida através equação 4.5.

$$T_i = R_i + E_i \tag{4.5}$$

Com relação aos n fluxos que compõem um fluxo agregado i, as taxas reservada r_{ij}, excedente e_{ij} e alvo t_{ij} para cada "subfluxo" (ou microfluxo) j, são definidas através das equações 4.6, 4.7 e 4.8, respectivamente.

$$r_{ij} = \frac{R_i}{n} \tag{4.6}$$

$$e_{ij} = \frac{E_i}{n} \tag{4.7}$$

$$t_{ij} = r_{ij} + e_{ij} = \frac{T_i}{n} \tag{4.8}$$

4.2 Principais Causas da Injustiça

Com o surgimento da proposta do serviço assegurado, vários trabalhos objetivaram investigar até que ponto se pode garantir a obtenção das taxas contratadas para

fluxos individuais ou agregados [56, 59, 60, 106, 110, 111]. Na grande maioria destes estudos foram utilizados os mecanismos de implementação descritos no capítulo 3, isto é, filas RED com múltiplos níveis e marcadores baseados em estimadores de taxa média ou balde de fichas (marcadores convencionais).

A partir daí, diversos fatores foram identificados como capazes de influir no desempenho do serviço assegurado (obtenção da taxa reservada) e consequentemente na questão da justiça (obtenção das taxas reservada e alvo) [58]. Alguns destes fatores serão discutidos a seguir a respeito de como contribuem para o problema da justiça no serviço assegurado.

4.2.1 Controles de Fluxo e de Congestionamento do TCP

O protocolo TCP utiliza duas janelas para efetuar os controles de fluxo e de congestionamento (apêndice A). O nó destino impõe um limite correspondente ao espaço disponível no seu *buffer* de recepção. Este valor é transmitido ao nó fonte através dos pacotes de reconhecimento e é denominado janela anunciada (*advertised window*). Já o nó fonte calcula uma janela de congestionamento como uma medida da capacidade da rede. Como resultado destas duas medições, o nó fonte não deixa circular na rede um número de pacotes não reconhecidos maior do que o mínimo entre estes dois valores. Quando a perda de um pacote é detectada, o TCP entra na fase de recuperação rápida (*fast recovery*) ou início lento (*slow start*), reduzindo a janela de congestionamento à metade ou um segmento, respectivamente. Este comportamento do TCP é conservativo na medida em que não é sensível ao nível de serviço. Consequentemente, conexões TCP podem deixar de atingir a taxa alvo ou até mesmo a taxa reservada.

Outro fator agravante é o surgimento de intervalos no fluxo de reconhecimentos, gerando consequentemente interrupções no fluxo de transmissão. Estes intervalos fazem parte de um fenômeno conhecido como compressão de reconhecimentos (*ack-compression*) [112] e podem ser causados por vários fatores [110]. O algoritmo de início lento faz com que o TCP envie dois pacotes "colados" (*back-to-back*) ao recebimento do primeiro reconhecimento, aumentando as chances de que os reconhecimentos sejam enviados pelo destino próximos uns dos outros e assim por diante. Outro fator é a dinâmica do algoritmo de recuperação rápida, a qual faz com que o TCP reduza sua janela à metade e interrompa a transmissão de novos segmentos até que metade dos pacotes da janela original deixem a rede. Deste modo, um intervalo é gerado na sequência de transmissão e propagado para o fluxo de reconhecimentos.

Finalmente, a influência do tráfego da rede através de congestionamentos nas direções normal e reversa, bem como atrasos adicionais em filas e *jitter* causado pela abertura de novas conexões, podem gerar intervalos significantes no fluxo de reconhecimentos [85]. Por estas razões o TCP apresenta características de um tráfego em rajadas.

Somam-se ainda a estes fatores alguns outros específicos da implementação do TCP que está sendo utilizada. Conforme será visto na subseção 4.2.4, as implementações mais comuns do TCP [113], Reno e New Reno, têm seu desempenho degradado na presença de múltiplos descartes em uma mesma janela de congestionamento.

Todos estes fatores contribuem para que fluxos de tráfego TCP tenham dificuldade em atingir as suas taxas reservadas. YEOM e REDDY [111] desenvolveram modelos matemáticos para estimar a vazão obtida para microfluxos isolados, ou seja, cada conexão com o seu perfil de tráfego específico. Os resultados principais indicam que para conexões com mesmo RTT e probabilidades de descarte, quanto maior a taxa reservada menor será a taxa excedente recebida relativamente, até o ponto em que nem a taxa reservada é atingida. Como consequência, o compartilhamento da banda excedente tende a favorecer os fluxos de menor taxa reservada, ferindo os critérios de justiça entre fluxos agregados nos critérios proporcional e igualitário. Como consequência, quanto menor o excedente de recursos, maior a será justiça entre fluxos agregados distintos. Estas observações correspondem aos resultados de estudos anteriores utilizando simulações [52, 56, 59, 60, 80, 114]. A justificativa para esta tendência é que a redução da janela de transmissão em virtude de um descarte afeta muito mais fluxos com taxas reservadas maiores. Também foi observado que a taxa obtida é inversamente proporcional ao RTT. Mais tarde, SAHU *et al.* [106] se inspiraram no trabalho de YEOM e REDDY para obter um modelo que levasse em conta a influência do uso de marcadores baseados em baldes de fichas. Resultados qualitativamente semelhantes foram obtidos. Além disso, foram identificadas condições em que onde o aumento do balde não afeta a vazão obtida. Mais especificamente, a influência do tamanho do balde diminui juntamente com o valor da taxa reservada, o RTT e a probabilidade de descarte de pacotes. No entanto, para casos de RTTs e taxas reservadas mais altas, o aumento do balde ajuda na obtenção da taxa assegurada até um limite onde rajadas muito grandes de pacotes prioritários (*in* ou verdes) podem provocar o descarte de tráfego assegurado [110].

Como forma alternativa ao aumento do tamanho dos baldes, BONAVENTURE

e CNODDER [115] propuseram dois suavizadores de tráfego. Denominados SRRAS (*Single Rate Rate Adaptive Shaper*) e TRRAS (*Two Rate Rate Adaptive Shaper*), foram criados para serem utilizados em conjunto com os marcadores SRTCM e TRTCM, respectivamente. Estes suavizadores se localizam antes do condicionador de tráfego e correspondem a uma fila FIFO com taxa de serviço variável em função da ocupação da fila. Os suavizadores visam eliminar parte da característica em rajadas do TCP antes da passagem pelo condicionador de tráfego, de forma a aumentar a taxa de marcação de pacotes verdes e contribuir para a obtenção da taxa reservada. Os resultados apresentaram uma melhora no desempenho [82, 115] apesar da desvantagem da introdução de retardos adicionais.

Outra consequência da dinâmica do TCP já mencionada é a influência do RTT na vazão obtida por uma conexão. O TCP utiliza um mecanismo de janelas deslizantes onde a chegada de reconhecimentos dispara novas transmissões. Logo, a velocidade com que uma conexão cresce a sua janela de transmissão é inversamente proporcional ao RTT. Este efeito é ilustrado formalmente na expressão 4.9 [88], onde VAZ é a vazão, MSS (*Maximum Segment Size*) o tamanho máximo do segmento e p a probabilidade de descarte de pacotes[4]. O mesmo vale para as velocidades de detecção e reação à ocorrência de descartes. Para o serviço assegurado, estes fatores fazem com que fluxos TCP de menor RTT tenham mais chance de atingir as suas taxas reservadas. Além disso, para o caso de redes superdimensionadas, estes fluxos obtêm a maior parte da largura de faixa excedente. Estes resultados foram observados em vários estudos através de simulações [52, 56, 59, 80], experimentos [60] e também nos modelos matemáticos propostos por YEOM e REDDY [111] e SAHU *et al.* [106].

$$VAZ \propto \frac{MSS}{RTT.\sqrt{p}} \tag{4.9}$$

Para o caso de fluxos agregados compostos por microfluxos TCP de RTT distintos, os fluxos de menor RTT também tendem a obter melhor desempenho quanto às taxas assegurada e excedente, quando houver [107, 108, 109].

4.2.2 Presença de Tráfego Não-Responsivo

O tráfego não-responsivo, ao contrário do TCP, não efetua controle de fluxo nem controle de congestionamento. Sendo assim, não reduz a taxa de transmissão

[4]Esta expressão vale para a fase de prevenção de congestionamento (*congestion avoidance*) do TCP (apêndice A), não sendo representativa para outras fases.

na ocorrência de descartes. FLOYD e FALL [116] propuseram mecanismos para identificar e penalizar estes fluxos. Outros mecanismos incluem: escalonamento por microfluxo, como por exemplo *Fair Queueing* [117]; escalonamentos que trabalham com níveis de granulosidade maior do que microfluxos, tais como *Class-Based Queueing* (CBQ) [67] e *Stochastic Fair Queueing* (SFQ) [118]; ou filas FIFO com descarte diferencial para fluxos com consumo desproporcional de largura de faixa, como o FRED.

No serviço assegurado, fontes de tráfego UDP podem impedir que fontes TCP obtenham as suas taxas reservadas caso ambos os tipos de tráfego sejam condicionados da mesma forma, ou seja, utilizem as mesmas prioridades de descarte de uma mesma classe AF. Isto vai depender de alguns fatores como a agressividade das fontes UDP em relação à capacidade do enlace compartilhado, os valores dos RTTs e das taxas reservadas para as fontes TCP, e também a escolha dos parâmetros das filas RIO. Sobre este último fator, recomenda-se uma diferenciação bem agressiva para pacotes fora do perfil. Um exemplo é adotar o esquema MAMT onde os limiares mínimo e máximo para os pacotes *in* e *out* não se superpõem e as médias são acumulativas. Deste modo, os fluxos TCP podem ter suas taxas reservadas protegidas [60, 114] ou não [52, 56], a depender de todos os demais fatores. Em redes subdimensionadas, os fluxos UDP podem "esmagar" as fontes TCP quando a soma de suas taxas asseguradas for comparável à capacidade do enlace gargalo [114].

Uma solução bastante proposta para este problema é a utilização de três prioridades de descarte, de forma que os tráfegos TCP e UDP sejam diferenciados quanto ao número e escolha das prioridades de descarte. ELLOUMI e CNODDER [82] implementaram esta proposta utilizando o TRTCM em um cenário com várias fontes TCP e uma fonte UDP. Para as fontes TCP o marcador é configurado com PIR igual à capacidade do enlace gargalo enquanto que para a fonte UDP $PIR = CIR$. Desta forma, o tráfego excedente para as fontes TCP é apenas amarelo e para a fonte UDP apenas vermelho. Este esquema se mostrou eficaz na proteção da taxa reservada para as fontes TCP. Porém, o fluxo UDP obtêm apenas o valor da taxa reservada independentemente da sua taxa de transmissão. Isto ocorre porque no cenário utilizado a largura de faixa excedente é totalmente ocupada pelo "tráfego amarelo" das fontes TCP, o que fere os critérios de justiça entre fluxos agregados para o compartilhamento da largura de faixa excedente. GOYAL *et al.* [61, 81] fizeram um estudo semelhante utilizando o mesmo critério de marcação para fontes TCP e UDP. Através da utilização do método ANOVA (*Analysis of Variance*) [119],

ressaltaram a grande influência da taxa de marcação de pacotes amarelos na justiça no compartilhamento da largura de faixa excedente. SEDDIGH *et al.* [83, 84] estudaram esta questão de forma mais abrangente através de experimentos. Variando os mapeamentos dos pacotes dentro e fora de perfil para os fluxos TCP e UDP de seis formas diferentes, chegaram a conclusão de que o cenário com resultados mais próximos dos critérios de justiça entre os fluxos agregados foi o mesmo utilizado nos estudos acima[5].

Uma outra solução mais imediata para este problema é mapear os tráfegos UDP e TCP em classes AF distintas, utilizando recursos reservados separadamente. NANDY *et al.* [107] comprovaram a eficácia deste método em relação ao uso de probabilidades de descarte distintas. No entanto, a desvantagem está na maior complexidade nas regras de classificação e aprovisionamento. Além disso, o volume de tráfego UDP, quando comparado ao volume de tráfego TCP (principalmente HTTP e FTP), pode não justificar a contratação de um serviço exclusivo, o que reforça a coexistência dos dois numa mesma classe AF.

No caso de fontes TCP e UDP fazerem parte de um mesmo perfil de tráfego, as fontes TCP são levadas a constantes *timeouts* e não conseguem atingir suas porções justas da taxa reservada do fluxo agregado [120], bastando para isso que as fontes UDP sejam agressivas o suficiente. Com isso, o critério de justiça para fluxos de um mesmo agregado não é possível de ser atingido. Soluções para este problema serão vistas em detalhes na seção 4.3.

4.2.3 Número de Fluxos Ativos

O número de fluxos também influi na justiça, pois fluxos agregados com maior volume de microfluxos tendem a obter maior desempenho quanto às taxas assegurada e excedente, se houver [60]. Isto ocorre porque quanto maior o número de microfluxos em um fluxo agregado, menor serão os tamanhos de suas janelas de congestionamento para uma dada taxa contratada [100]. Sendo assim, as perdas sofridas têm menor impacto na redução da taxa de transmissão total. Além disso, o TCP aumenta sua janela de congestionamento até que haja uma perda. Portanto, quanto mais fluxos maior o aumento do tráfego total do agregado em um dado intervalo de tempo. Desta forma, mesmo que as taxas reservadas dos agregados sejam iguais, aquele que tiver o maior número de fluxos será mais competitivo no compartilhamento da

[5]SEDDIGH *et al.* [83, 84] utilizaram no máximo duas prioridades de descarte para cada tipo de tráfego.

largura de faixa excedente [108, 109].

O número total de fluxos ativos no núcleo da rede também tem influência na vazão obtida pelos microfluxos de um fluxo agregado. A flutuação na vazão obtida por fluxos TCP e a consequente falta de justiça no compartilhamento de um enlace gargalo aumentam com o número de microfluxos. Segundo MORRIS [121], este efeito começa a aparecer quando o número de fluxos supera o de pacotes que cabem na memória da rede (produto entre o retardo e a largura de faixa de gargalo).

A eficácia dos parâmetros do RED também depende do número de fluxos ativos [98]. Toda conexão TCP tenta manter pelo menos um pacote em trânsito. Logo, um número grande de conexões tende a encher as filas, fazendo com que o limiar médio máximo (max_{th}) seja atingido, causando muitas perdas e levando algumas conexões ao *timeout*. Sendo assim, o compartilhamento da largura de faixa obtida por um fluxo agregado por entre os seus microfluxos tende a ser menos uniforme quanto maior o número de fluxos, mesmo para RTTs iguais [113].

4.2.4 Implementação do TCP

Cada implementação do TCP tem a sua maneira de reagir aos descartes de pacotes. O TCP Tahoe entra em início lento reduzindo sua janela a um segmento tanto na ocorrência de *timeout* como no recebimento de reconhecimentos duplicados. Isto representa um comportamento extremamente conservativo quando são poucos os pacotes descartados dentro de uma mesma janela de congestionamento. Já o TCP Reno evita a execução do algoritmo de início lento para o caso de reconhecimentos duplicados através do algoritmo de recuperação rápida. No entanto, foi mostrado que o TCP Reno tem o seu desempenho degradado quando vários pacotes de uma mesma janela são descartados. Mais especificamente, três descartes ou mais numa mesma janela, a depender do espaçamento entre eles, quase sempre fazem necessário a espera de um *timeout* para a retransmissão [122]. Uma forma variante, chamada New Reno, evita a necessidade de muitos destes *timeouts*. Porém, o nó fonte fica limitado a transmitir no máximo um segmento por RTT, degradando em muito o seu desempenho nestas ocasiões. FALL e FLOYD [122] mostraram estas deficiências através de simulações, comparando as três implementações acima com o TCP com reconhecimentos seletivos ou SACK (*Selective ACKnowledgments*) [92]. Através deste estudo, defenderam o uso do TCP SACK para corrigir os problemas acima e evitar a retransmissão de pacotes entregues com sucesso. No entanto, enquanto os TCPs Reno e New Reno requerem modificações apenas no nó fonte, o TCP SACK

requer modificações em ambos os nós da conexão. Logo, apesar do desenvolvimento agressivo de novas versões, muitas delas ficam comprometidas quanto à implantação em larga escala na Internet.

Portanto, a capacidade de manter a vazão na presença de descartes é diferente em cada caso. Sendo assim, duas conexões ou fluxos agregados com as mesmas probabilidades de descarte podem apresentar diferentes desempenhos em termos de obtenção das larguras de faixa assegurada e excedente, dependendo da implementação utilizada. FEROZ *et al.* [113] compararam o desempenho de dois marcadores com as implementações Reno, New Reno e SACK. Foi observado que o TCP SACK reduz em muito o número de *timeouts* mas o desempenho quanto a taxa de perdas e vazão média obtida pelos fluxos foi ligeiramente inferior ao Reno e ao New Reno. Não foram observadas grandes diferenças quanto à justiça.

4.2.5 Tamanho do Pacote IP

O tamanho do pacote utilizado por uma conexão TCP também influencia na forma como os recursos serão divididos. Fluxos TCP com mesmo RTT mas tamanhos de pacotes diferentes podem apresentar diferentes desempenhos em termos de obtenção das larguras de faixa assegurada e excedente. Mais precisamente, para fluxos individuais ou agregados com mesma taxa assegurada e mesmo RTT, aqueles que tiverem maior tamanho de pacote irão obter maior largura de faixa excedente, pois num mesmo intervalo de tempo enviam um maior volume de informação [60]. Isto explica porque as expressões matemáticas para a vazão do TCP são diretamente proporcionais ao tamanho do segmento [111, 122]. O mesmo vale para o compartilhamento de largura de faixa entre fluxos de um mesmo agregado.

4.3 Justiça entre Fluxos de um Mesmo Agregado

A partir daqui, este trabalho se concentra nas propostas para atacar o problema da justiça entre fluxos de um mesmo agregado. O grande estímulo para este problema é a tendência natural de que os clientes contratem serviços baseando-se no tráfego total em direção ao provedor. Conforme visto anteriormente, vários fatores influenciam a justiça no compartilhamento das larguras de faixa assegura e excedente. As principais abordagens para amenizar o efeito destes fatores serão discutidas a seguir.

4.3.1 Soluções Propostas

As três principais soluções para a falta de justiça diferem basicamente pela localização do mecanismo que tenta fornecer a justiça. A primeira abordagem se localiza no próprio protocolo e propõe modificações no controle de congestionamento do TCP de forma a torná-lo sensível ao nível de serviço desejado. O protocolo modificado denomina-se TCP de duas janelas (*two-windows TCP*) [80, 85] porque divide a janela de congestionamento em uma janela reservada e outra excedente. A janela reservada é calculada multiplicando-se a taxa reservada pelo RTT estimado pelo TCP, enquanto que a excedente é obtida subtraindo a janela reservada da janela de congestionamento. O TCP é modificado para reduzir somente a janela excedente pela metade quando um pacote *out* for descartado (algoritmo 4.1). Para o descarte de pacotes *in* não há modificações e a janela de congestionamento é reduzida à metade.

Algoritmo 4.1: Controle de congestionamento modificado.

Variáveis:
  $cwnd$: janela de congestionamento
  $rwnd$: janela reservada
  $ewnd$: janela excedente
Parâmetros:
  r_i: taxa assegurada
Após cada perda de um pacote:
  se o pacote descartado é *out*
    $rwnd = RTT.r_i$
    se $(rwnd < cwnd)$
      $ewnd = cwnd - rwnd$
      $cwnd = rwnd + ewnd/2$
  senão
    $cwnd = cwnd/2$

A proposta TCP de duas janelas, apesar de fornecer uma maior proteção quanto à obtenção da taxa reservada, possui algumas deficiências que limitam ou até mesmo condenam a sua adoção:

- requer modificações no protocolo de transporte direcionadas ao serviço assegurado, quando a Internet e o próprio DiffServ oferecem vários outros serviços;

- necessita que o nó fonte receba informações do condicionador de tráfego para saber o resultado da marcação de cada pacote. Nenhuma proposta foi feita sobre como implementar este mecanismo adicional;
- não há nenhum mecanismo para garantir a obtenção da taxa excedente. Isto é, se $rwnd$ é sensível a r_i, o mesmo não ocorre para $ewnd$ e e_i. Desta forma, fluxos com menor RTT serão favorecidos sob este aspecto;
- funciona apenas para o TCP e não lida com a influência dos fluxos não-responsivos.

Uma segunda abordagem propõe modificações nos roteadores dos nós DS através do uso de disciplinas de gerenciamento ativo que protejam os fluxos TCP mais frágeis (maiores RTTs) e penalizem os fluxos não responsivos. Uma forma de implementar este mecanismo é utilizando filas FRED de múltiplos níveis [85].

Porém, no interior de um domínio DS o nível de agregação se torna maior e problemas de ordem escalar passam a ser uma preocupação, mesmo considerando que o FRED só mantém estados para os fluxos que possuem pacotes na fila. Sob este aspecto talvez a utilização de filas FRED seja mais adequada para a justiça entre fluxos agregados distintos [105]. Além disso, o algoritmo FRED tenta distribuir a largura de faixa disponível igualmente entre todos os fluxos. Logo, alguma alteração deve ser feita no algoritmo para tratar casos com perfis contratados distintos (potencialmente mais comuns).

Uma outra questão a ser considerada é a adição de complexidade no interior da rede. Além disso, apesar dos serviços diferenciados estarem ainda em processo de padronização e portanto suscetível a mudanças, a sua arquitetura hoje determina que o tratamento recebido no interior de um domínio DS seja função apenas do valor do *codepoint*. Portanto, adotar esta estratégia significa mudar este conceito na medida em que o PHB passar a ser função também do nível de serviço contratado por cada cliente.

Finalmente, os roteadores do provedor teriam que ser reconfigurados na contratação e encerramento de serviços, de forma a redistribuir os recursos contratados e excedentes de forma justa, segundo o critério adotado.

A terceira abordagem é a utilização de estratégias de marcação que enderecem a questão da justiça. Esta proposta consiste em distribuir o volume de informação marcada prioritariamente de forma justa entre os fluxos que compõem o tráfego total. Importantes vantagens podem ser identificadas com relação às abordagens

anteriores:

- simplicidade pois a informação sobre o perfil de tráfego fica armazenada nos próprios condicionadores de tráfego, evitando a necessidade de comunicação entre componentes distintos da rede;
- atua nas bordas da rede sendo mais escalável e de acordo com a filosofia DiffServ;
- permite constantes evoluções dos algoritmos cujas atualizações causam menor impacto do que nos casos anteriores. Nesta abordagem, apenas os marcadores nas bordas de domínios DiffServ devem ser modificados, certamente em número menor do que os roteadores DiffServ (nós DS) e estações com TCP/IP instalado.
- é mais flexível na medida em que soluções específicas podem ser desenvolvidas dependendo do tipo de tráfego a ser condicionado em um domínio restrito.

Por todos estes motivos, este trabalho defende fortemente o uso de marcadores justos como a principal e mais adequada forma de endereçar o problema da justiça entre fluxos de um mesmo agregado. A seguir, as possíveis estratégias de marcação para atingir este objetivo serão classificadas e comparadas quanto aos seguintes critérios: propriedade escalar, utilização dos recursos da rede, complexidade do marcador, custo financeiro, custo de manutenção e, obviamente, justiça.

4.3.2 Estratégias de Marcação para Obtenção de Justiça

4.3.3 Marcação por Agregado (MA)

Na marcação por agregado um único condicionador atua diretamente sobre o tráfego agregado do cliente em direção ao provedor. Além disso, a medição e a marcação são feitas baseando-se apenas no estado total do fluxo agregado (figura 4.4). Portanto, esta estratégia de marcação não fornece nenhum mecanismo para conter as causas da falta de justiça discutidas na subseção 4.2, o que inviabiliza sua escolha quando a justiça é uma métrica de desempenho fundamental. Os marcadores convencionais apresentados no capítulo 3 são exemplos desta estratégia quando atuam sobre um fluxo agregado de tráfego.

Mesmo não sendo uma estratégia indicada para resolver o problema da justiça, vale a pena verificar alguns de seus benefícios a título de comparação com as abordagens que serão apresentadas a seguir. Além disso, podem surgir casos em que a

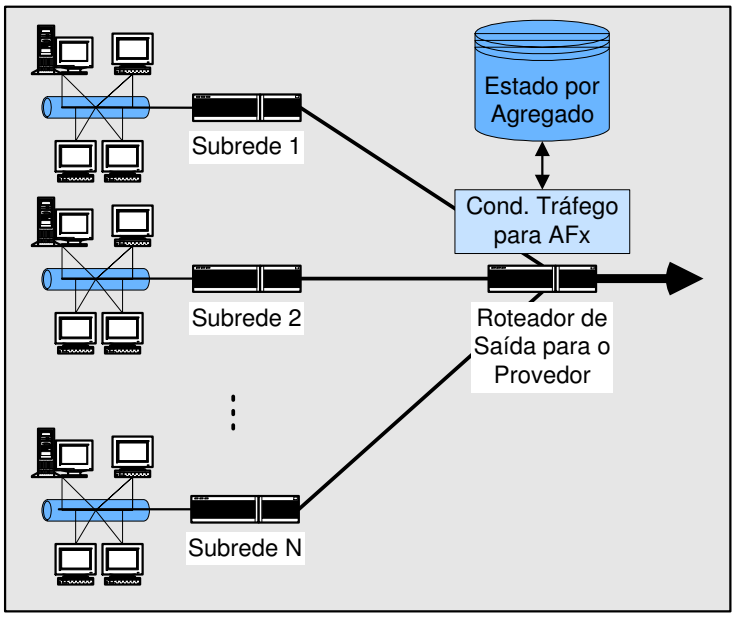


Figura 4.4: Marcação por agregado.

justiça não seja um fator fundamental para a qualidade de serviço contratada. Uma primeira vantagem é a baixa complexidade dos condicionadores de tráfego dada a simplicidade dos seus algoritmos de marcação. Outra é o baixo custo com equipamento específico, pois apenas um condicionador é necessário para todo o tráfego. Isto reduz ainda os custos de manutenção. Outra opção é que o próprio provedor realize o condicionamento de tráfego ao invés de policiamento simples, eliminando a necessidade de qualquer componente da arquitetura DiffServ no cliente. Finalmente, não há problemas de ordem escalar na medida em que o número de fluxos que compõe o tráfego agregado nada influi na complexidade do marcador.

4.3.4 Marcação por Fluxo (MF)

Uma primeira alternativa de tentar resolver o problema da justiça é utilizar vários marcadores, um para cada um dos n subfluxos para os quais se deseja um compartilhamento justo da largura de faixa. A condição necessária para possibilitar esta solução é saber de antemão quantos e quais são estes subfluxos, o que é mais viável quando correspondem a porções de tráfego bem definidas e com um certo nível de agregação. Um exemplo clássico seria o tráfego de subredes de um campus. Neste caso, um condicionador de tráfego seria colocado em cada enlace entre as subredes e o roteador de saída para o provedor (figura 4.5).

Esta estratégia apresenta a vantagem de permitir ainda que o perfil de tráfego contratado seja repartido da forma desejada, não necessariamente de forma igual entre todos os subfluxos. Sendo assim, há uma grande flexibilidade para o estabelecimento de políticas internas dentro do domínio do cliente. No entanto, uma única restrição é que a soma dos perfis de tráfego de cada subfluxo não ultrapasse o valor contratado, a fim de evitar que o policiamento do provedor efetue descartes na entrada do seu domínio.

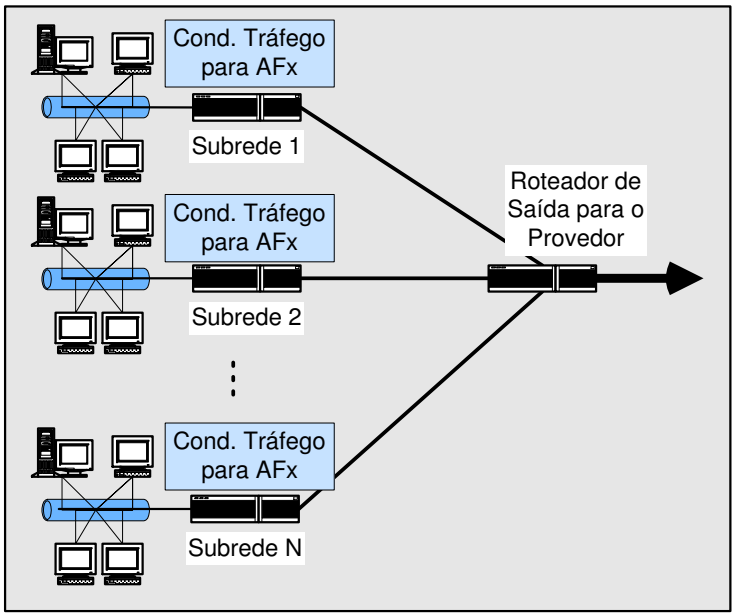


Figura 4.5: Marcação por fluxo.

Outra vantagem é a ausência de problemas de propriedade escalar dado que a sua aplicação é restrita a casos onde o número de fluxos é limitado. Além disso, condicionadores de tráfego convencionais de simples implementação podem ser utilizados para dividir a taxa assegurada entre os fluxos. Para repartir as larguras de faixa assegurada e excedente, podem ser empregados marcadores de três cores. Dois exemplos seriam o TRTCM com $CIR = r_i$ e $PIR = t_i$, e o TSWTCM com $CTR = r_i$ e $PTR = t_i$.

Dentre os pontos negativos desta alternativa estão os maiores custos financeiro e de manutenção associados ao maior número de equipamentos necessários para a sua implementação. Uma outra desvantagem muito importante é a atribuição de porções fixas do tráfego assegurado (r_i) para cada marcador. Isto faz com que subfluxos em períodos mais ociosos desperdicem sua fatia do tráfego assegurado, enquanto que

os demais ficam impedidos de reaproveitá-la porque são condicionados por outros marcadores[6]. Este problema não ocorre na marcação por agregado.

No entanto, para os casos onde os subfluxos correspondem a fluxos de menor nível de agregação, tais como o tráfego de usuários, estações ou até microfluxos, a adoção deste tipo de solução vai se tornando inviável. Isto porque o problema da ineficiência na utilização da largura de faixa assegurada se torna mais crítico. Além disso, especialmente para o caso de microfluxos, provavelmente será impossível prever o número de microfluxos bem como a duração de suas conexões. Logo, uma outra estratégia se faz necessária para satisfazer a justiça nestes casos, conforme será visto a seguir.

4.3.5 Marcação por Agregado Atenta a Fluxos (MAF)

A marcação por agregado atenta a fluxos visa unir as vantagens das duas propostas anteriores. Isto significa prover justiça entre os fluxos que compõem o tráfego agregado sem os problemas de ineficiência no aproveitamento da largura de faixa assegurada, e ser capaz de lidar com um número imprevisível de fluxos.

Para atingir este objetivo, esta estratégia utiliza apenas um marcador para todo o fluxo de tráfego do cliente, da mesma forma que a MA. Porém, estados são criados e mantidos para cada subfluxo para o qual se deseja prover uma porção justa do tráfego assegurado (figura 4.6), conforme seus pacotes passam pelo condicionador de tráfego. Estes estados guardam informações que serão utilizadas por um algoritmo de marcação que possui duas funções básicas. A primeira é distribuir o número de pacotes marcados como dentro do perfil da forma desejada. A segunda é evitar que esta distribuição prejudique o fluxo agregado em termos de obtenção da taxa assegurada. Portanto, a eficácia da estratégia MAF quanto à capacidade de prover a justiça e à utilização da largura de faixa assegurada dependem do algoritmo utilizado.

É verdade que a complexidade dos marcadores tende a ser maior para esta estratégia. Porém, os custos com equipamento e manutenção são basicamente os mesmos para o caso da marcação por agregado, pois um único marcador é utilizado[7]. A tabela 4.2 resume a comparação entre as estratégias de marcação apresentadas segundo todos os critérios discutidos.

[6]Uma forma de atacar esta limitação seria através de mecanismos de comunicação entre os marcadores.

[7]Este custo pode ser aumentado em função da complexidade associada à implementação da lógica do marcador.

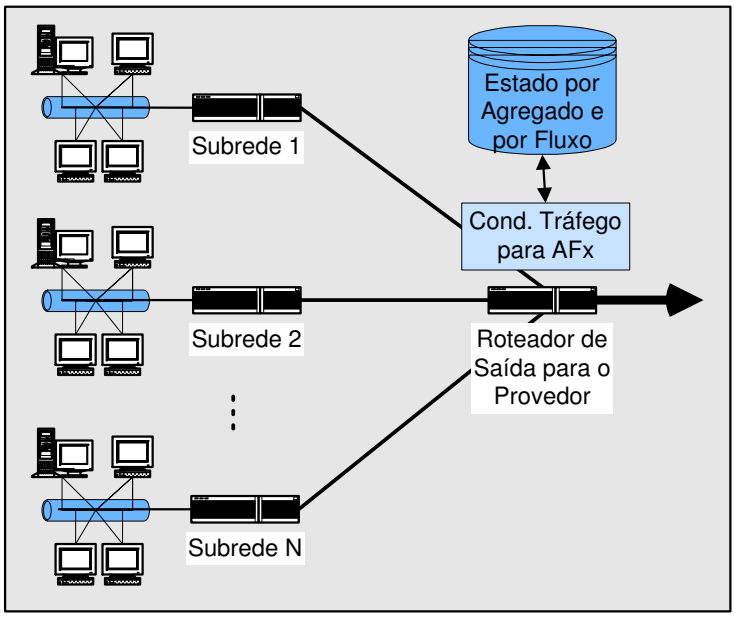


Figura 4.6: Marcação por agregado atenta a fluxos.

Apesar de seus benefícios potenciais, poucos marcadores foram propostos adotando esta estratégia como forma de combater a justiça. FEROZ *et al.* [113] propuseram um marcador baseado em balde de fichas cujo algoritmo tenta proteger os fluxos TCP mais frágeis (com maiores valores de RTT). Para isto, o algoritmo distribui as fichas disponíveis prioritariamente entre estes fluxos. Além disso, o algoritmo tenta espaçar os pacotes marcados como *in* para cada fluxo de forma a evitar rajadas de tráfego assegurado. Os resultados mostraram um desempenho superior ao balde de fichas quanto à justiça. No entanto, foram observados resultados muito baixos (em torno de 50%) para o percentual entre a taxa obtida e a taxa assegurada

Tabela 4.2: Quadro comparativo entre as estratégias de marcação.

Critério	Justiça	Propriedade escalar	Utilização larg.faixa	Complex. marcador	Custo financeiro	Manutenção
MA	†	⋆	⋆	⋆	⋆	⋆
MF	⋆	⋆	†	⋆	†	†
MAF	$f(alg)$	$f(alg, n)$	$f(alg)$	†	⋆	⋆

† → ponto fraco, ⋆ → ponto forte, $f(alg)$ → função do algoritmo de marcação, $f(n)$ → função do número de fluxos

do fluxo agregado, comprometendo o desempenho algoritmo em um dos objetivos expostos anteriormente.

YEOM e REDDY [108, 109] propuseram um mecanismo de marcação baseado em estimativa de taxa média para o fluxo agregado e também para cada fluxo individual. O marcador se mostrou superior ao TSW na obtenção da taxa assegurada e também na justiça. Porém, por ser um marcador baseado em estimativa de taxa média, o o fluxo agregado chega a obter uma vazão de tráfego assegurado 17% maior do que o valor contratado. Além disso, algoritmo precisa saber de antemão quantos são os fluxos para obter as suas taxas alvo.

O fato dos algoritmos destas duas propostas utilizarem informações que devem ser mantidas para cada um dos fluxos é um fator a ser considerado. Isto pode causar problemas de ordem escalar quando o número de fluxos for demasiadamente grande. Além disso, não foi especificada nenhuma maneira de lidar com a dinâmica de abertura e fechamento das conexões TCP, a qual pode pode fazer com que estados sejam criados e não mais utilizados. Isto é, não é eficiente manter estados para todos os fluxos que passaram pelo menos uma vez pelo marcador, sobretudo quando o nível de agregação é alto. Portanto, é necessário algum critério para apagar os estados dos fluxos inativos. Estes estudos também careceram de resultados na presença de tráfego não-responsivo.

KIM [62] propôs uma forma variante da estratégia MAF através da manutenção de estados apenas para fluxos ativos. Esta proposta mostrou resultados superiores a todas as demais propostas tanto em termos de justiça quanto em termos de obtenção da taxa assegurada. Além disso, se mostrou eficaz também na presença de tráfego não-responsivo. Por estes motivos, este marcador será descrito em detalhes a seguir.

4.3.6 O Marcador Justo (*Fair Marker*)

O FM (*Fair Marker*) [62] consiste em um marcador baseado em balde de fichas, e que efetua a marcação por agregado atenta a fluxos. Como qualquer marcador que utiliza a estratégia MAF, o FM mantém estados para os fluxos do tráfego agregado de forma a fornecer justiça entre eles. Neste caso, cada estado contém informação a respeito do consumo de fichas dos fluxos monitorados. Além disso, o FM armazena estados apenas para os fluxos que consumiram fichas (tiveram pacotes marcados como *in*) durante o último intervalo de tempo correspondente ao preenchimento completo do balde, denominado $TBFT$ (*Token Bucket Fill Time*) e que vale CBS/CIR. Desta forma, o FM evita problemas de ordem escalar através da

manutenção de estados apenas para os fluxos ativos.

Resta saber então como o FM realiza a distribuição de fichas de forma a garantir a justiça entre os fluxos. Para isso, o FM se baseia em uma analogia criada entre um balde de fichas e uma fila de mesmo tamanho, onde manter estados para os fluxos que consumiram fichas durante o último $TBFT$ é análogo a manter estados para fluxos que possuem pacotes numa fila. Pacotes marcados como *in* correspondem a pacotes armazenados na fila. Pacotes marcados como *out* correspondem a pacotes descartados e portanto não armazenados na fila. Quando o balde está vazio, nenhum pacote pode ser marcado como *in*, assim como quando a fila está cheia nenhum pacote pode ser armazenado. Fichas são desperdiçadas quando o balde enche e nenhum pacote chega, assim como um enlace de transmissão é subutilizado quando a fila está vazia e não há pacotes para serem enviados. Indo mais além, pode-se imaginar um pacote consumindo fichas como uma situação análoga ao que seria o seu "rastro" (*trace*) substituindo estas fichas no balde. Na prática, cria-se uma fila complementar onde estes rastros são armazenados. Sempre que uma ficha é gerada, consulta-se a fila para saber se o número de fichas no balde é suficiente para retirar o rastro do pacote no início da fila (o mais "antigo"). Para obter a justiça basta então aplicar um algoritmo justo de gerenciamento de filas. Ou seja, se o espaço da fila for compartilhado de forma justa, então o mesmo acontecerá para o consumo de fichas no balde. Portanto, determinar se um pacote deve consumir fichas (ser marcado como *in*) equivale a determinar se o seu rastro pode ser armazenado na fila complementar, de acordo com o algoritmo escolhido.

Logo, duas condições devem ser satisfeitas para que os pacotes de um fluxo sejam marcados como *in*. A primeira é que haja fichas disponíveis no balde. A segunda é que o controle por fluxo representado pela fila de rastros e pelo seu algoritmo de gerenciamento justo permitam que este fluxo possa ter um pacote marcado como *in*, baseando-se no seu consumo de fichas durante o último $TBFT$. A figura 4.7 ilustra o funcionamento do FM onde CIR é a taxa de preenchimento de fichas e CBS o tamanho do balde.

A lógica da marcação encontra-se no algoritmo 4.2. Para cada pacote que chega, o balde de fichas será atualizado primeiro. Serão acrescentadas $(t\_now\_ - t\_last\_).CIR$ fichas até CBS. O número de fichas acrescentado ao balde, $(T\_last\_ - T\_now\_)$, é utilizado para retirar rastros da fila complementar. Isto equivale a atualizar a tabela de estados deixando apenas informações sobre fluxos que consumiram fichas durante o último $TBFT$. Conforme os rastros dos pacotes vão sendo

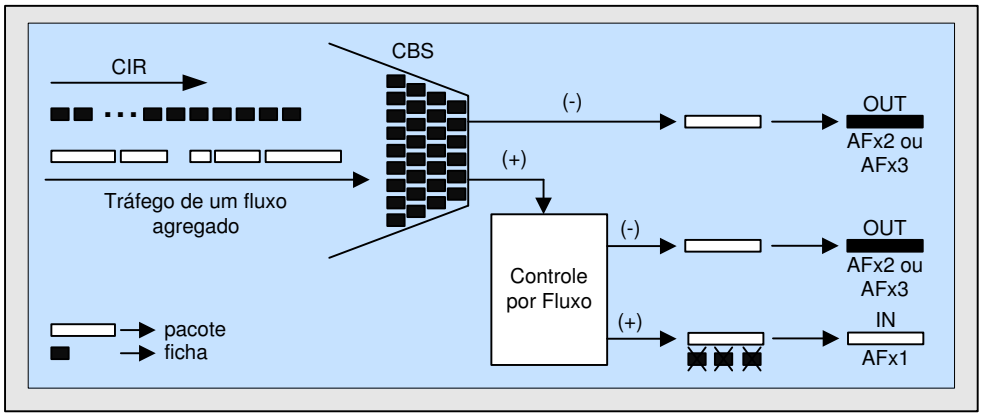


Figura 4.7: O marcador justo.

retirados, as estatísticas de consumo de fichas no balde (utilização do espaço na fila de rastros) vão sendo atualizadas, tarefa realizada pelo próprio algoritmo justo de gerenciamento de filas.

Algoritmo 4.2: Marcação no FM.

Variáveis:

$t\_now\_$: instante de tempo da chegada do último pacote

$t\_last\_$: instante de tempo da chegada do penúltimo pacote

$T\_now\_$: quantidade de fichas no balde em $t\_now\_$

$T\_last\_$: quantidade de fichas no balde em $t\_last\_$

$p\_now\_$: último pacote recebido

Funções:

$tempo\_corrente()$: retorna o instante de tempo atual

$tamanho(P)$: retorna o tamanho de um pacote P

$desenfileira\_rastros()$: retira rastros da fila e atualiza estados

$enfileira\_rastro(P)$: se um pacote P pode ser enfileirado na fila de rastros, atualiza estados e retorna verdadeiro; caso contrário, retorna falso

Inicialmente:

$T\_last\_ = T\_now\_ = CBS$

$t\_last\_ = t\_now\_ = tempo\_corrente()$

A cada pacote que chega:

$T\_now\_ = T\_now\_ + (tempo\_corrente() - t\_last\_).CIR$

se $(T\_now\_ > CBS)$

$T\_now\_ = CBS$

Algoritmo 4.2 (cont.): Marcação no FM.

```
A cada pacote que chega: (cont.)
  desenfileira_rastros()
  se (T_now_ >= tamanho(p_now_))
    se (enfileira_rastro(p_now_)
      marcar pacote como in
      T_now_ = T_now_ − tamanho(p_now_)
    senão
      marcar pacote como out
  senão
    marcar pacote como out
  t_last_ = tempo_corrente()
  T_last_ = T_now_
  enviar pacote
```

O próximo passo então é analisar o número de fichas no balde. Se este for insuficiente, o pacote será marcado como *out* e enviado. Seu rastro não é colocado na fila complementar e nenhuma estatística por fluxo é alterada. Se o número de fichas for suficiente, então o consumo de fichas no último $TBFT$ determinará então se o pacote pode consumir as fichas (ser enfileirado na fila de rastros) ou não. Caso ele não possa ser enfileirado, o seu rastro não é colocado na fila, o pacote é marcado como *out* e enviado. Caso ele possa ser enfileirado, o seu rastro entra na fila e a tabela de estados é atualizada. Além disso as fichas são consumidas, o pacote é marcado como *in* e enviado.

Como para qualquer marcador que utiliza a estratégia MAF, o seu desempenho quanto à justiça e obtenção da taxa assegurada é função do algoritmo escolhido. No caso do FM, apenas as funções $desenfileira\_rastros()$ e $desenfileira\_rastros()$ serão modificadas de acordo. Além disso, novos parâmetros podem ser adicionados para configurar o algoritmo utilizado. KIM [62] comparou duas opções, FRED e DT (*Dynamic Threshold*) [123], sendo que o desempenho do FRED foi muito superior.

Aplicando o funcionamento do FRED (subseção 3.3.3) na analogia entre o enfileiramento de pacotes e a distribuição de fichas, pode-se dizer que o FM executará o seguinte controle por fluxo, em cada $TBFT$:

- todo fluxo poderá marcar até min_q pacotes como *in* desde que o número médio de pacotes marcados como *in* não exceda max_{th};

- assim como no RED, pacotes adicionais estarão sujeitos a serem marcados como *out*, de forma aleatória, se o número médio de pacotes marcados como *in* exceder min_{th}. Porém, isto só acontecerá para fluxos que já tenham marcado mais do que *avgcq* pacotes como *in*;

- um fluxo nunca poderá marcar mais do que max_q pacotes como *in*, e cada tentativa de ultrapassar este limite será contabilizada e fará com que este não consiga marcar mais do que *avgcq* pacotes como *in*.

Desse modo, a implementação do FM utilizando FRED tende a melhorar a justiça entre fluxos, na medida em que:

- protege fluxos que se adaptam mais lentamente a uma maior disponibilidade de largura de faixa (maiores RTTs);

- ameniza a heterogeneidade entre os fluxos descartando mais daqueles que estão obtendo uma maior taxa de marcação de pacotes como *in*;

- permite que fluxos adaptativos enviem rajadas de tráfego assegurado, mas evita que o tráfego não-responsivo monopolize o consumo de fichas do marcador.

Além destes fatores, a aplicação desta lógica apenas para fluxos que tenham marcado pacotes como *in* no último $TBFT$ garante a propriedade escalar da solução.

ALVES *et al.* [120] estudaram o desempenho do FM em função dos ajustes dos parâmetros do algoritmo FRED, e propuseram recomendações neste sentido. Um estudo semelhante e mais completo será descrito no capítulo 5. ALVES *et al.* constataram também que apesar do FM ser muito eficaz na obtenção e compartilhamento justo da taxa assegurada, não consegue prover justiça no compartilhamento da banda excedente, principalmente na presença de tráfego não-responsivo. Isto acontece porque nenhum tratamento diferenciado é aplicado em cima dos pacotes que não são marcados como *in*. Seu desempenho com relação a esta métrica é portanto comparável ao de um marcador balde de fichas convencional (TBM) [120].

4.3.7 O Marcador Justo de Três Cores

De modo a corrigir a deficiência do FM no compartilhamento do tráfego excedente, é proposto neste trabalho uma extensão ao FM chamada TCFM (*Three Color Fair Marker*) [63]. O TCFM trabalha com três prioridades de descarte e é obtido do FM adicionando outro balde com o seu próprio controle por fluxo. O primeiro balde

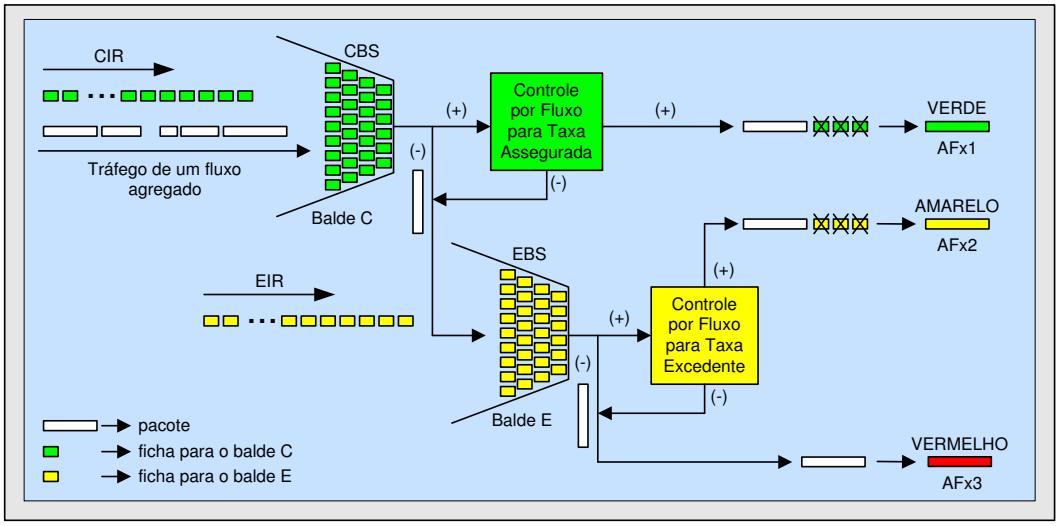


Figura 4.8: O marcador justo de três cores.

tem tamanho CBS e é preenchido com a taxa reservada CIR. O seu controle por fluxo trata de distribuir as "fichas verdes" de forma justa tal como no FM para os pacotes *in*. O outro balde possui tamanho EBS (*Excess Burst Size*) e é preenchido com taxa EIR. O controle de fluxo adicional trata de dividir as "fichas amarelas" de forma a fornecer a justiça na largura de faixa excedente. O valor de EIR pode ser configurado com o valor E_i conforme as equações 4.3 e 4.4, a depender da política adotada pelo provedor para divisão da capacidade excedente. Outra opção é que EIR e CIR somadas correspondam a um valor de taxa de pico contratado, tal como o valor PIR para o TRTCM.

O TCFM funciona da seguinte forma (figura 4.8). Um pacote será marcado com verde se existe um número de fichas suficiente no "balde verde" e se este pode ser enfileirado na "fila de rastros verde". Senão, isto é, se pelo menos uma das condições não é satisfeita, o pacote será marcado como amarelo se as mesmas condições valem para o "balde amarelo" e a "fila de rastros amarela". Caso contrário, o pacote é marcado como vermelho. Um marcador semelhante foi proposto por ANDRIKOPOULOS e PAVLOU [105].

Resultados preliminares mostraram uma substancial melhora (em torno de 600%) em relação ao FM, ambos utilizando o algoritmo FRED para o controle por fluxo. O desempenho do TCFM será avaliado com mais detalhes no capítulo 5.

Capítulo 5

Resultados

Neste capítulo serão apresentados e discutidos os resultados dos estudos realizados através de simulações. Na primeira parte, serão feitas considerações gerais sobre a ferramenta utilizada, objetivos dos estudos, cenário escolhido e métricas de desempenho. Na segunda parte, o FM será analisado quanto ao impacto do ajuste dos seus parâmetros na obtenção da justiça. Finalmente, o TCFM, proposto neste trabalho, terá o seu desempenho comparado ao do FM em várias situações distintas.

5.1 Considerações Gerais

5.1.1 Técnica de Avaliação

Foi escolhida a técnica de simulação através da ferramenta NS-2 (*Network Simulator version 2*) [124], largamente utilizada na maioria dos estudos realizados em DiffServ.

O NS-2 consiste em um simulador dirigido a eventos discretos que implementa as abstrações de nós e enlaces, bem como os protocolos de roteamento e das camadas de rede, transporte e aplicação da arquitetura TCP/IP. Para estudar o serviço assegurado, foi necessário desenvolver módulos adicionais ao NS-2 contendo:

- os condicionadores de tráfego TBM (*Token Bucket Marker*), FM (*Fair Marker*) e TCFM (*Three Color Fair Marker*), os quais medem as propriedades temporais do tráfego, comparam-nas com o perfil de tráfego e marcam o DSCP dos pacotes de acordo;
- as disciplinas de gerenciamento ativo de filas RIO e RED com três níveis, de forma a implementar o PHB-AF nos nós DS como função do valor do DSCP.

5.1.2 Objetivos

Os quatro objetivos principais dos estudos realizados, não necessariamente em ordem de importância, estão relacionados abaixo:

- entender a influência do ajuste dos parâmetros do algoritmo FRED no desempenho do FM com relação à justiça[1];
- avaliar o desempenho do TCFM frente à proposta original FM, principalmente quanto à justiça no compartilhamento da largura de faixa excedente;
- evidenciar os benefícios da estratégia MAF (Marcação por Agregado Atenta à Fluxos) para endereçar o problema da justiça entre os fluxos de um mesmo tráfego agregado, comparando os desempenhos dos marcadores justos FM e TCFM com o do marcador convencional balde de fichas (TBM).

5.1.3 Cenário Escolhido

Topologia

Para atingir os objetivos definidos anteriormente, foi utilizado um sistema constituído de uma rede IP, de acordo com a figura 5.1. A topologia de um único gargalo é ideal para o estudo da justiça porque constitui um cenário bastante comum onde vários fluxos de tráfego compartilham um mesmo recurso na rede de um provedor de serviços.

As larguras de faixa e retardos de propagação associados aos enlaces da espinha dorsal da rede do provedor (roteadores $R1$, $R2$, $R3$ e $R4$) estão representados na própria figura 5.1. Os roteadores $F1, F2, ..., F10$ transmitem o tráfego gerado pelas fontes de tráfego em direção ao provedor através de enlaces de acesso de 10Mbps para o roteador $R1$. As fontes de tráfego de um roteador F_i transmitem sempre para algum nó D_i onde $i = 1, 2, ..., 10$. Os nós D_i se ligam ao roteador $R4$ também por intermédio de enlaces de 10Mbps e retardos de propagação de 1ms. Os valores dos retardos de propagação para os enlaces entres as fontes e o roteador $R1$ podem variar. Em alguns estudos, todos estes retardos são iguais de forma a fazer com que não haja diferença entre os RTTs mínimos de cada conexão TCP. Em outros estudos, eles são diferentes de forma a observar a influência do RTT na justiça entre os fluxos.

[1] Conforme comentado no capítulo 4, o algoritmo FRED será utilizado nas implementações do FM e TCFM como forma de distribuição justa das fichas entre os fluxos.

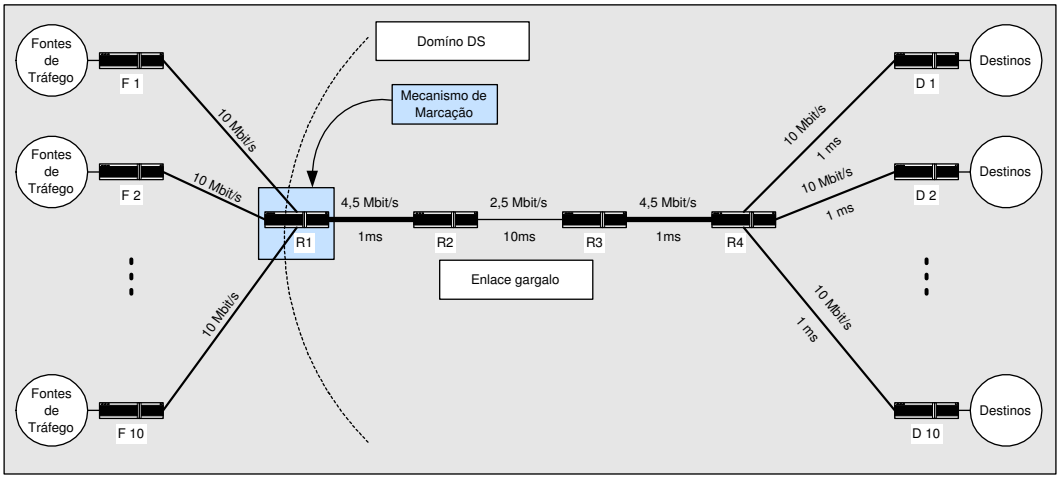


Figura 5.1: Cenário escolhido para as simulações.

O condicionador de tráfego se situa sempre no roteador de entrada para o provedor $R1$, fazendo a marcação em cima de todo o tráfego agregado.

Maiores detalhes sobre cada cenário serão vistos nas seções 5.2 e 5.3.

Simplificações

O serviço fornecido pela rede consiste na entrega de pacotes IP de um nó fonte a um nó destino. Os seguintes resultados são possíveis, de acordo com as características da arquitetura TCP/IP.

- pacote entregue em ordem ao destino, ou seja, quando não há nenhum anterior a ele na sequência de dados recebidos que ainda esteja pendente;
- pacote entregue fora de ordem ao destino;
- pacote duplicado entregue ao destino[2];
- pacote descartado por motivos de congestionamento;
- pacote descartado por motivos de corrupção dos dados (*checksum*);
- pacote descartado por número excessivo de nós atravessados (TTL - *Time To Live*).

Devido às métricas de desempenho, topologia e técnica de avaliação utilizadas, alguns destes resultados são condensados em um, ou até mesmo não contemplados.

[2]Estas três primeiras possibilidades valem para o caso de conexões TCP.

Para o cálculo das métricas de desempenho (subseção 5.1.4), é importante o número de pacotes entregues corretamente em um determinado intervalo de tempo. Para obter esta informação, é utilizada a diferença entre os números de reconhecimento contidos nos últimos pacotes de reconhecimento enviados pelo destino no início e no fim deste intervalo[3]. Sendo assim, a diferenciação entre pacotes que foram entregues em ordem ou fora de ordem não é relevante. Além disso, as entregas em duplicidade não são contabilizadas.

O NS-2 não contempla a possibilidade de descarte do pacote por erros de transmissão. O uso de uma topologia com rotas estáticas e com um número pequeno de roteadores também elimina a possibilidade de descarte por número excessivo de nós atravessados. Porém, estas duas simplificações não comprometem os estudos realizados pois representam uma parcela muito reduzida dos casos de descarte de pacotes em redes TCP/IP [89].

Existem ainda mais três simplificações a serem destacadas. A primeira é o tempo de processamento nulo nos roteadores (marcadores), o qual pode ser desprezado se comparado aos retardos de transmissão e de propagação utilizados. O segunda é o fato do TCP do NS-2 utilizar um pacote de reconhecimento para cada pacote recebido. No entanto, a ausência de reconhecimentos atrasados (*delayed ACKs*) é uma questão que visa apenas aumentar a eficiência na utilização da largura de faixa (apêndice A), não tendo grandes implicações na dinâmica do TCP. Além disso, a arquitetura DiffServ é unidirecional. Logo, o aprovisionamento no sentido contrário (onde os reconhecimentos atravessam a rede) é um problema a parte. Finalmente, o NS não implementa o controle de fluxo através do anúncio do espaço de armazenamento disponível no destino. No lugar deste mecanismo existe apenas um parâmetro que define um valor máximo fixo para a janela de congestionamento[4].

[3]No NS-2, o número de reconhecimento presente no cabeçalho dos pacotes de reconhecimento indica o número do último pacote recebido. Isto é, a sequência de dados é medida em pacotes e não em bytes. Embora diferente do caso real onde o número de reconhecimento contém o próximo byte a ser recebido, esta simplificação não afeta os resultados tendo em vista que o tamanho dos pacotes é fixo para uma dada conexão. Porém, existe a possibilidade de que alguns pacotes entregues ao destino não sejam contabilizados caso a entrega de algum pacote anterior a estes fique pendente até o instante final do intervalo de simulação.

[4]Em todas as simulações este valor foi escolhido muito alto para que a janela de transmissão fosse limitada apenas pelo valor da janela de congestionamento.

5.1.4 Métricas de Desempenho

Para medir a justiça no compartilhamento da largura de faixa será utilizado o índice de justiça proposto por JAIN *et al.* [125]. Seu cálculo é feito através da equação 5.1, onde n é o número de fluxos de um tráfego agregado i e v_j é a vazão de cada um destes n fluxos[5].

$$fi_i = \frac{(\sum_{j=1}^{n(i)} v_j)^2}{n(i).\sum_{j=1}^{n(i)} (v_j)^2} \tag{5.1}$$

O índice de justiça apresenta vantagens em relação a outras métricas de desempenho. Para citar alguns exemplos, ele é independente de métrica e escala, o que não ocorre com a variância. Também é limitado entre 0 e 1, o que não ocorre com a coeficiente de variação, o qual pode assumir valores entre 0 e ∞.

O índice de justiça possui uma propriedade que esclarece bem o seu significado. Se uma quantidade k de um recurso qualquer é divida igualmente entre m de n usuários (cada um recebe k/m), onde $0 \leq m \leq n$, então o índice de justiça vale m/n. Isto significa que a distribuição foi justa para $(m/n).100\%$ dos usuários.

Nos estudos realizados, o fi pode medir a justiça no compartilhamento das larguras de faixa assegurada, excedente e total. Isto depende apenas de como serão calculadas as vazões de cada fluxo através da equação 5.2, onde TP_j é o tamanho dos pacotes IP para a conexão j em bytes, TS o intervalo de tempo de tempo de duração da simulação em segundos e NP_j é o número de pacotes entregues ao destino para a conexão j durante o intervalo de tempo TS, sem contar eventuais duplicatas.

$$v_j = \frac{NP_j.TP_j.8}{TS}(bps) \tag{5.2}$$

Para medir a justiça no compartilhamento da largura de faixa total, NP_j representa todos os pacotes entregues ao destino. Para a largura de faixa assegurada, serão considerados os pacotes dentro do perfil (*codepoint* AFx1). Isto equivale aos pacotes *in* para no caso de marcadores que trabalham com duas prioridades de descarte (FM e TBM) e aos pacotes verdes no caso do TCFM. Por último, para a largura de faixa excedente, serão considerados os pacotes fora do perfil (*codepoints* AFx2 e AFx3, se houver). Isto equivale aos pacotes *out* para no caso de marcadores que trabalham com duas prioridades de descarte (FM e TBM) e aos pacotes amarelos e vermelhos no caso do TCFM.

[5]Como apenas a justiça entre fluxos de um único agregado estará sendo analisada, o índice i será omitido a partir deste ponto.

5.2 Primeiro Estudo - Ajuste dos Parâmetros do FM

O objetivo principal deste primeiro estudo é ter uma idéia de como o ajuste dos parâmetros do algoritmo FRED pode influenciar qualitativamente no desempenho do FM, e a partir daí obter algumas diretivas quanto à forma de configurar este marcador. Este estudo também permitirá comparar as estratégias de marcação MAF e MA quanto à justiça na largura de faixa assegurada e corresponde a uma versão mais completa e abrangente do estudo feito por ALVES *et al.* [120].

Utilizando a topologia da figura 5.1, são criados dois cenários denominados TCPs heterogêneos (RTTs mínimos distintos) sem tráfego não responsivo e TCPs homogêneos (mesmo RTT mínimo) com tráfego não responsivo. Com isso, é possível observar separadamente o desempenho do FM na presença de causas distintas para o problema da justiça: a diferença de RTTs e a presença de tráfego não adaptativo.

Em ambos os cenários existe uma fonte de tráfego FTP/TCP[6] da fonte n para o destino $n+10$, onde $n = 1, 2, ..., 10$. O TCP Reno foi escolhido por ser o mais comum na Internet. No cenário TCPs heterogêneos (figura 5.2), o retardo de propagação nos enlaces entre fontes e o roteador $R1$ varia de 10ms até 100ms numa progressão aritmética de razão 10ms, o que resulta em retardos de propagação totais entre fonte e destino variando de 23ms até 113ms. No cenário TCPs homogêneos (figura 5.3), este valor é de 5ms. Além disso, existe uma fonte de tráfego CBR (*Constant Bit Rate*)/UDP do nó 1 para o nó 11 com taxa de transmissão de 2,5Mbps (100% da velocidade de transmissão do enlace gargalo). Todos os pacotes de dados são de 1500bytes. Os roteadores do provedor possuem filas RIO MAMT (*Multiple Average Multiple Thresholds*) com limiares acumulativos. Os parâmetros para as filas *in* e *in + out* valem [0,5; 0,8; 0,002; 0,02][7] e [0,2; 0,5; 0,002; 0,1] respectivamente[8].

O valor para o tamanho das filas nos roteadores ($qlim$) é de 50 pacotes (75000 bytes) e o tamanho do balde (CBS) é de 100 pacotes. Para verificar se estes valores não comprometem a validade do estudo, isto é, não estão sub ou superdimensionados, são feitos dois conjuntos de simulações utilizando o cenário TCPs heterogêneos sem CBR/UDP[9]. No primeiro conjunto, o tamanho do balde é mantido fixo e o tamanho

[6] O FTP pode ser utilizado no NS-2 para fazer com que o TCP sempre tenha dados a transmitir.

[7] [$min_{th}/qlim$; $max_{th}/qlim$; w_q; max_p], onde $qlim$ corresponde ao tamanho da fila.

[8] Considerações sobre o ajuste de parâmetros no RED e no FRED encontram-se no apêndice B.

[9] Este cenário foi escolhido para que os parâmetros fossem dimensionados de acordo com a dinâmica do TCP.

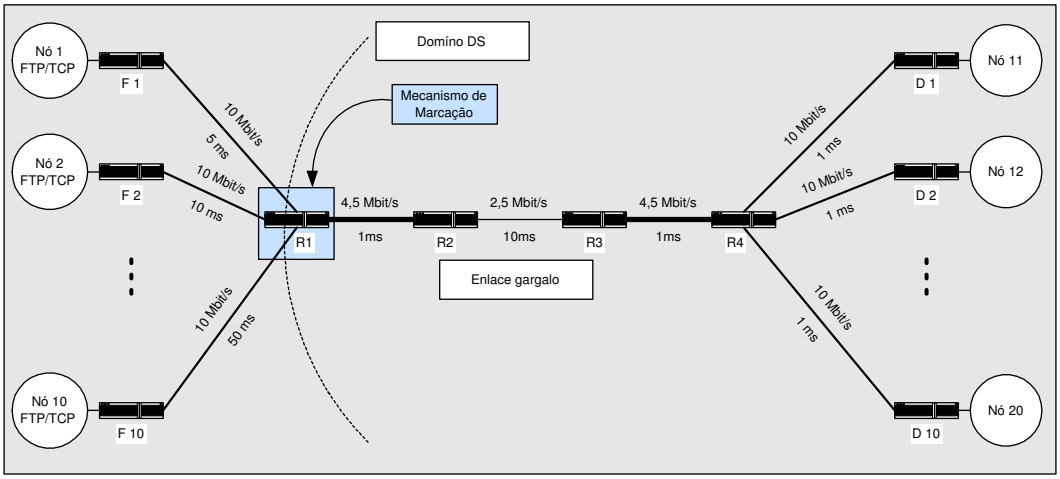


Figura 5.2: Cenário TCP heterogêneos sem CBR/UDP.

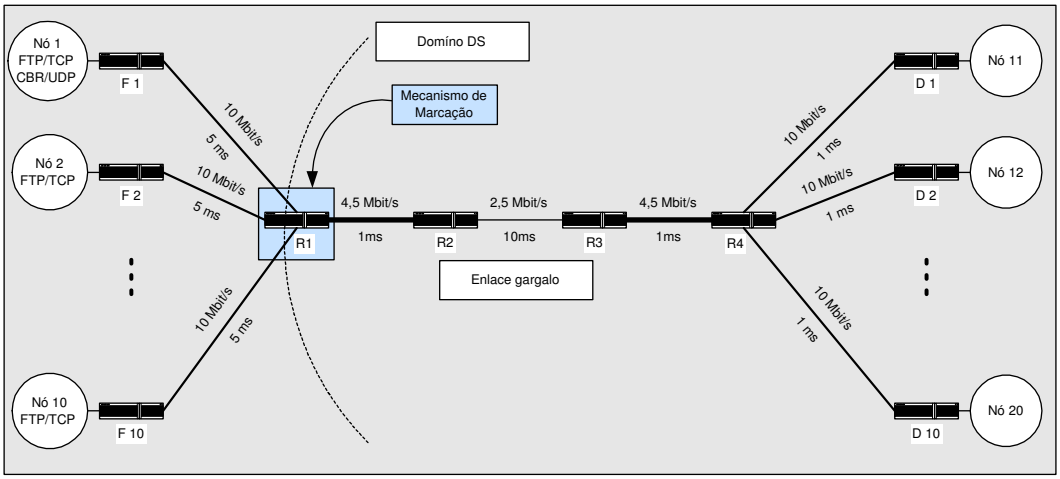
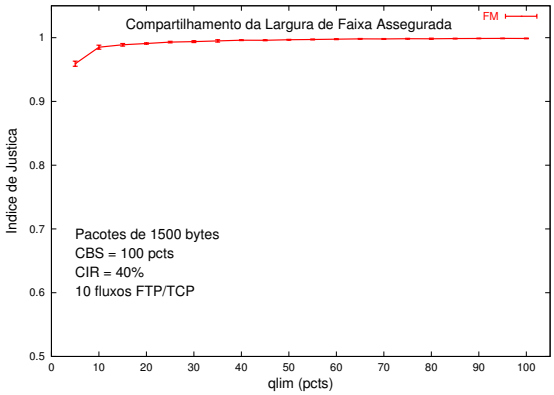


Figura 5.3: Cenário TCP homogêneos com CBR/UDP.

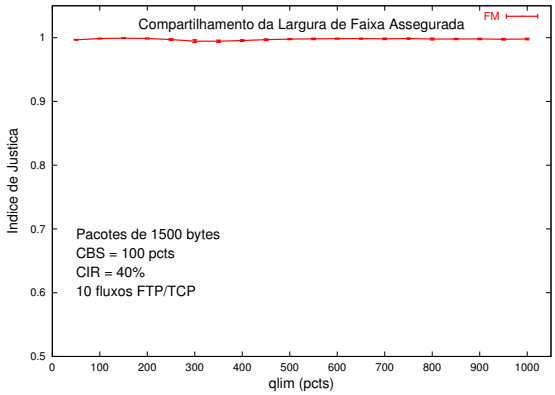
das filas é variado. No segundo conjunto é feito o contrário. Em ambos os casos a taxa de preenchimento do balde (CIR) vale 1Mbps, correspondendo à 40% da capacidade do enlace gargalo. Os valores dos parâmetros do algoritmo FRED associado ao FM foram escolhidos conforme recomendado por LIN e MORRIS [98] (apêndice B). Sendo assim, $w_q = 0,002$, $max_p = 0,02$, $min_q = 2$, $max_q = min_{th} = 25\%CBS$ e $max_{th} = 2.min_{th} = 50\%CBS$.

O tempo total para cada simulação é de 500s. As fontes começam a transmitir entre 0 e 10s de forma aleatória para evitar o sincronismo para o tráfego TCP. Para desconsiderar a influência do transiente do TCP, os primeiros 50s são desprezados.

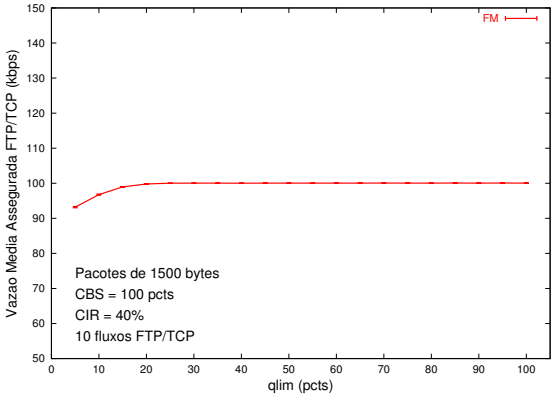
A figura 5.4 mostra os resultados para a variação do tamanho das filas. O índice



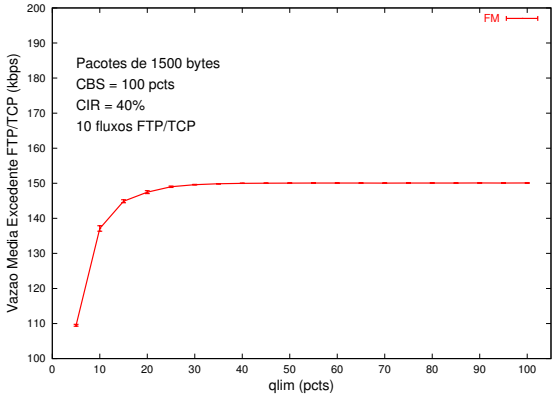
(a) De 0 até 100 de 5 em 5.



(b) De 0 até 1000 de 50 em 50.



(c) De 0 até 100 de 5 em 5.



(d) De 0 até 100 de 5 em 5.

Figura 5.4: Variação do tamanho da fila.

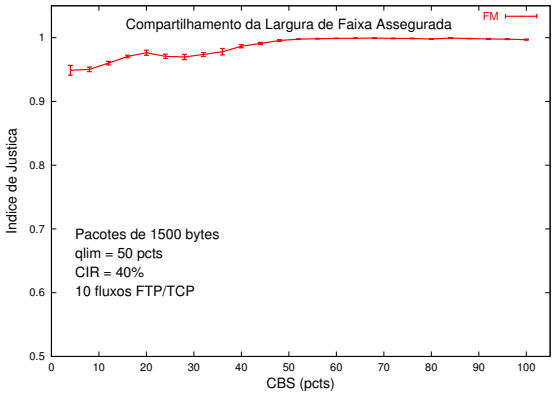
de justiça para o compartilhamento da largura de faixa assegurada é mostrado nas figuras 5.4a e 5.4b, variando o tamanho das filas de 0 até 100 pacotes e de 0 até 1000 pacotes, respectivamente. As figuras 5.4c e 5.4d mostram as vazões médias assegurada e excedente considerando as dez conexões, variando o tamanho das filas de 0 até 100 pacotes. Para obter qualquer resultado, são feitas dez simulações e tiradas as médias e os intervalos de confiança de $\pm 0,5\%$ (nível de confiança de 99%). Por exemplo, para obter o índice de justiça na largura de faixa assegurada para um dado tamanho de fila, é efetuada a média aritmética dos índices de justiça calculados para cada uma das dez simulações conforme equação 5.1, onde $n = 10$. Para obter a vazão v_j de cada conexão, é utilizada a equação 5.2, onde NP_j é igual aos números de pacotes *in* entregues ao destino em TS, $TP_j = 1500$ bytes e $TS = 500 - 50 = 450$s.

Pode-se concluir que valores muito baixos de $qlim$ (menores que 25 pacotes) prejudicam a justiça (figura 5.4a) e (ou) a vazão (figura 5.4c). Filas muito pequenas enchem mais facilmente e aumentam as chances de operação na fase de controle de congestionamento, diminuindo a eficácia da combinação entre os mecanismos de marcação e de gerenciamento ativo de filas. Isto porque não basta distribuir pacotes in justamente se a rede não é capaz de protegê-los. Estes descartes de pacotes prioritários podem ser comprovados através da figura 5.4c, onde os fluxos TCP não conseguem atingir a taxa assegurada média desejada de 100kbps (r_j) para valores de $qlim$ menores que 25 pacotes. A figura 5.4d mostra que o problema acontece de forma mais intensa (para $qlim \leq 30$ pacotes) com a vazão média excedente, cujo valor desejado é 150kbps (e_j). Para valores mais altos de $qlim$ (≥ 20), o índice de justiça se mantém aproximadamente no mesmo patamar (figuras 5.4a e 5.4b)[10]. A figura 5.4 também permite concluir que o valor de 50 pacotes para $qlim$ representa uma boa escolha para o estudo do marcador FM no cenário escolhido, tendo em vista que nenhum dos problemas acima ocorre.

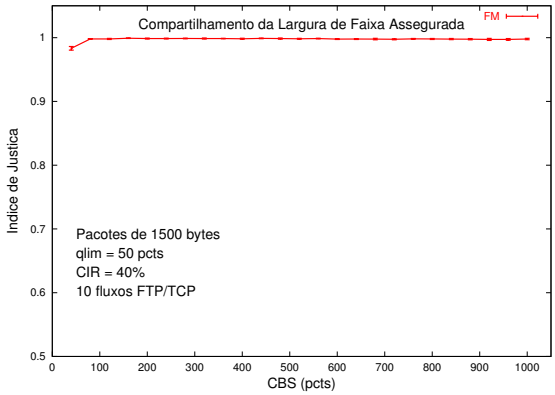
A figura 5.5 mostra os mesmos resultados para a variação do tamanho do balde. De maneira semelhante ao caso anterior, valores mais baixos de CBS (menores que 50 pacotes) prejudicam a justiça (figura 5.5a) e (ou) a vazão (figura 5.5b). Quando o balde é muito pequeno não há espaço suficiente para todos os fluxos, o que prejudica a distribuição das fichas de forma justa para um tráfego explosivo como o TCP. Com isso, alguns fluxos acabam sendo favorecidos. Além disso, fichas são perdidas mais facilmente pelo transbordo do balde fazendo com que a vazão média assegurada desejada não seja atingida (figura 5.5c). Em compensação, já que o tamanho de 50 pacotes das filas RIO é adequado, a vazão média excedente (figura 5.5d) aumenta proporcionando 250kbps de vazão média total (t_j). De acordo com os gráficos, o valor de 100 pacotes para CBS também representa uma boa escolha para o estudo do marcador FM.

Definidos estes parâmetros pode-se prosseguir com o estudo do marcador FM. Para entender a influência dos parâmetros do FRED no seu desempenho, os valores de min_q, $max_q = min_{th}$ e max_{th} são variados conforme a tabela 5.1, respeitando as desigualdades $min_q < (max_q = min_{th}) < max_{th}$. Os demais parâmetros são mantidos constantes com valores $w_q = 0,002$ e $max_p = 0,02$ [98]. Logo, são obtidas

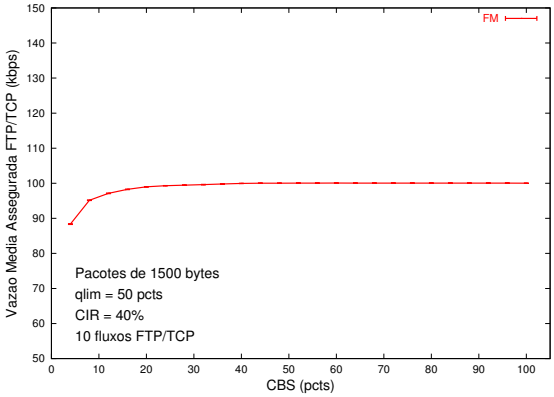
[10] Os gráficos para as vazões médias assegurada e excedente quando o tamanho da fila varia de 0 até 1000 pacotes mostram que estes valores se mantêm respectivamente em 100kbps e 150kbps conforme o tamanho da fila aumenta, da mesma forma que nas figuras 5.4c e 5.4d para os valores mais altos de $qlim$. Estes gráficos foram omitidos por razões de espaço.



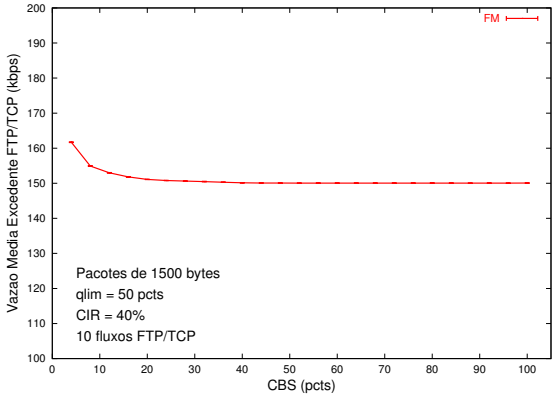
(a) De 0 até 100 de 4 em 4.



(b) De 0 até 1000 de 40 em 40.



(c) De 0 até 100 de 4 em 4.



(d) De 0 até 100 de 4 em 4.

Figura 5.5: Variação do tamanho da balde.

vinte e nove configurações cujos valores dos parâmetros estão descritos na tabela 5.2, juntamente com um número identificador[11].

Tabela 5.1: Variação dos parâmetros do FM: percentuais e valores.

min_q	2; 4; 15 (15%); 25 (25%)
$max_q = min_{th}$	10 (10%); 25 (25%); 50 (50%); 75 (75%)
max_{th}	25 (25%); 50 (50%); 75 (75%); 100 (100%)

A fim de comparar as estratégias de marcação MAF (FM) e MA, são obtidos resultados para o marcador balde de fichas convencional (TBM), cujo tamanho do balde CBS é igual ao do FM (50 pacotes). Para não limitar o estudo a um único

[11] As configurações recomendadas por LIN e MORRIS [98] correspondem às de número 5 e 15.

Tabela 5.2: Configuração obtidas: numeração e valores dos parâmetros.

#	min_q	$max_q = min_{th}$	max_{th}
1	2	10	25
2	2	10	50
3	2	10	75
4	2	10	100
5	2	25	50
6	2	25	75
7	2	25	100
8	2	50	75
9	2	50	100
10	2	75	100
11	4	10	25
12	4	10	50
13	4	10	75
14	4	10	100
15	4	25	50
16	4	25	75
17	4	25	100
18	4	50	75
19	4	50	100
20	4	75	100
21	15	25	50
22	15	25	75
23	15	25	100
24	15	50	75
25	15	50	100
26	15	75	100
27	25	50	75
28	25	50	100
29	25	75	100

perfil de tráfego, o valor da taxa contratada varia de 10% à 90% em incrementos de 20%. Assim como nos resultados obtidos anteriormente, são feitas dez simulações

para cada caso, cada uma durando 500s, com corte dos primeiros 50s e nível de confiança de 99%. As configurações são então comparadas pelo valor médio do índice de justiça no compartilhamento da largura de faixa assegurada. Portanto, é utilizado como métrica o fi (equação 5.1), onde $n = 10, 11$ em cenários sem e com tráfego não responsivo, respectivamente.

De acordo com os critérios de justiça expostos no capítulo 4, a tabela 5.3 mostra os valores para as taxas reservadas r_j e excedente e_j para cada conexão j em função do valor de CIR (percentual em relação ao enlace gargalo) e do cenário. Vale notar que a taxa alvo t_j independe do valor de CIR e vale 250kbps (2,5Mbps/10) para o cenário TCP heterogêneos e 227,3kbps (2,5Mbps/11) para o cenário TCP homogêneos.

Tabela 5.3: Taxas reservada e excedente para cada conexão em kbps.

Cenário	10%	30%	50%	70%	90%
TCP het.	25,0/225,0	75,0/175,0	125,0/125,0	175,0/75,0	225,0/25,0
TCP hom.	22,7/204,5	68,2/159,1	113,6/113,6	159,1/68,2	204,5/22,7

5.2.1 Cenário TCP Heterogêneos sem CBR/UDP

A tabela 5.4 mostra os resultados obtidos para o cenário TCP heterogêneos para as vinte e nove configurações, numeradas conforme a sua primeira coluna. Os valores dos parâmetros estão descritos nas três próximas colunas. A seguir estão os valores médios do índice de justiça no compartilhamento da largura de faixa assegurada para cada valor de CIR[12], seguidos pelo total considerando estes cinco conjuntos de simulações. As configurações aparecem em ordem decrescente de acordo com este total[13]. Os resultados para o TBM se encontram no final da tabela (configuração de número 30).

Tendo em vista que quase na totalidade dos casos os índices de justiça para o compartilhamento da largura de faixa assegurada se situam entre 0,9 e 1[14], poderia-se dizer a princípio que este cenário não apresentou uma diferenciação significativa

[12]Os resultados estão arredondados para 4 casas decimais e os intervalos de confiança aperecem nos gráficos.

[13]Apenas uma forma intuitiva de ordenar as configurações pelo desempenho geral nos cinco casos, já que elas serão comparadas pelos resultados numéricos, gráficos e análise do comportamento do algoritmo FRED do marcador.

[14]Exceto para o TBM, $CIR = 10\%, 50\%$.

Tabela 5.4: Índice de justiça no compartilhamento da largura de faixa assegurada. Cenário TCP heterogêneos sem CBR/UDP.

#	min_q	max_q	max_{th}	10%	30%	50%	70%	90%	Total
2	2	10	50	0,9999	0,9994	0,9982	0,9964	0,9973	4,9912
12	4	10	50	0,9999	0,9994	0,9983	0,9965	0,9972	4,9912
7	2	25	100	0,9988	0,9997	0,9986	0,9989	0,9951	4,9911
23	15	25	100	0,9984	0,9998	0,9984	0,9982	0,9961	4,9910
17	4	25	100	0,9980	0,9997	0,9987	0,9986	0,9948	4,9898
3	2	10	75	0,9999	0,9994	0,9982	0,9958	0,9948	4,9881
13	4	10	75	0,9999	0,9994	0,9981	0,9954	0,9951	4,9879
14	4	10	100	0,9999	0,9994	0,9985	0,9955	0,9938	4,9871
4	2	10	100	0,9999	0,9994	0,9984	0,9954	0,9940	4,9870
15	4	25	50	0,9996	0,9983	0,9950	0,9979	0,9961	4,9869
5	2	25	50	0,9995	0,9981	0,9953	0,9979	0,9961	4,9868
6	2	25	75	0,9999	0,9997	0,9985	0,9912	0,9971	4,9864
16	4	25	75	0,9999	0,9997	0,9985	0,9908	0,9964	4,9853
22	15	25	75	0,9999	0,9997	0,9986	0,9908	0,9929	4,9819
24	15	50	75	0,9998	0,9997	0,9971	0,9853	0,9908	4,9728
18	4	50	75	0,9998	0,9996	0,9956	0,9834	0,9935	4,9720
8	2	50	75	0,9999	0,9996	0,9963	0,9815	0,9929	4,9701
11	4	10	25	0,9895	0,9972	0,9969	0,9907	0,9830	4,9572
1	2	10	25	0,9892	0,9972	0,9967	0,9906	0,9791	4,9528
21	15	25	50	0,9996	0,9985	0,9967	0,9969	0,9208	4,9125
27	25	50	75	0,9998	0,9996	0,9964	0,9844	0,9229	4,9031
25	15	50	100	0,9278	0,9350	0,9277	0,9622	0,9940	4,7467
26	15	75	100	0,9365	0,9277	0,9259	0,9622	0,9937	4,7460
9	2	50	100	0,9302	0,9351	0,9291	0,9509	0,9825	4,7278
28	25	50	100	0,9189	0,9364	0,9295	0,9472	0,9718	4,7037
19	4	50	100	0,9109	0,9292	0,9196	0,9532	0,9833	4,6962
10	2	75	100	0,9240	0,9235	0,9134	0,9506	0,9827	4,6941
29	25	75	100	0,9274	0,9231	0,9197	0,9465	0,9690	4,6857
20	4	75	100	0,9200	0,9251	0,9045	0,9493	0,9816	4,6805
30	TBM			0,8985	0,9106	0,8821	0,9174	0,9355	4,5440

tanto entre as configurações do FM como entre o FM e o TBM. Porém, a partir da utilização de gráficos em escalas menores para realçar as diferenças entre os resultados, isto é, com ordenadas entre 0,9 e 1, é possível identificar padrões de comportamento do FM em função do ajuste de seus parâmetros.

A figura 5.6 mostra os resultados das oito configurações de pior resultado[15]. São elas: 9, 10, 19, 20, 25, 26, 28 e 29. As "curvas" de resultado para este grupo de configurações apresentam o mesmo "formato" que a do TBM. A razão para isto é a baixa agressividade do algoritmo FRED relativamente ao tamanho médio da fila de rastros para este cenário. Isto pode ser constatado na medida em que as oito configurações podem ser subdivididas em dois grupos, ambos com quatro configurações. O primeiro subgrupo contém as configurações 9, 19, 25 e 28, as quais possuem em comum $min_{th} = max_q = 50$ e $max_{th} = 100$. O segundo subgrupo contém as configurações 10, 20, 26 e 29, as quais possuem em comum $min_{th} = max_q = 75$ e $max_{th} = 100$. Estes são os ajustes menos agressivos dentre todas as vinte e nove configurações[16].

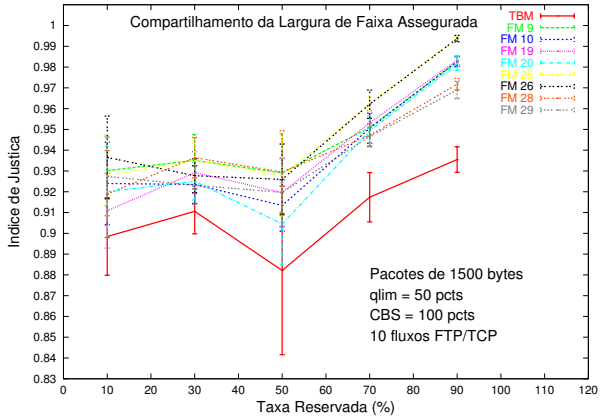


Figura 5.6: Cenário TCP heterogêneos sem CBR/UDP: piores resultados.

A falta de agressividade do algoritmo nestes oito casos também pode ser comprovada através do registro dos resultados do processo de marcação de cada pacote no

[15] A tabela está subdividida em blocos de configurações com desempenhos próximos.

[16] A agressividade é inversamente proporcional aos valores de min_{th} e max_{th}, pois quanto menor forem estes valores, mais cedo o algoritmo entra nas fases de prevenção e controle de congestionamento, nesta ordem. Além disso, a agressividade do RED e suas formas variantes também é diretamente proporcional à inclinação Θ da reta que descreve a fase de prevenção ao congestionamento, dada pela relação $tg\ \Theta = max_p/(max_{th} - min_{th})$. Sendo assim, pode-se concluir que a agressividade do RED também é inversamente proporcional à diferença entre max_{th} e min_{th}.

FM. Para isso, "grampos" foram colocados no código deste marcador para registrar quantos pacotes foram marcados como *in*, quantos foram marcados como *out* por falta de fichas e quantos foram marcados como *out* pela ação do algoritmo justo de distribuição de fichas[17]. Além disso, para o último caso foi registrado também qual trecho do algoritmo FRED levou o pacote a ser marcado como *out*[18]. A partir destes registros, foi possível constatar que para este grupo de oito configurações o percentual de pacotes marcados como *out* pela ação do FRED não ultrapassou 5.4%, comprovando portanto a baixa eficácia do controle por fluxo do marcador. Desta forma, o rendimento se torna apenas ligeiramente superior ao do TBM.

Outro grupo peculiar de configurações apresenta uma alta queda de desempenho para $CIR = 90\%$. Para a configuração 27 o motivo é o alto valor de min_q. Isto inibe o descarte aleatório para os fluxos robustos dentro da fase de prevenção ao congestionamento (grampo 3,0, $qlen_i > MAX(min_q, avgcq)$). Os registros mostram um percentual deste tipo de punição de 1,6% contra valores em torno de 33% obtidos nas configurações 2 e 12 (de melhor desempenho)[19]. Para a configuração 21, um valor mais baixo que no caso anterior porém ainda alto de min_q mantém o percentual de punição dos fluxos mais robustos ainda relativamente baixo (3,9%). Isto contribui ainda para um aumento do tamanho médio da fila. Como neste caso o valor de max_{th} está mais próximo de min_{th}, a fila de rastros chega a entrar na fase de controle de congestionamento (em torno de 10,1% de ocorrência do grampo 2,0), degenerando para uma fila que descarta todo e qualquer pacote que chega (modo *drop tail*), perdendo seletividade na distribuição das fichas.

A seguir, as configurações 1 e 11 se caracterizam pela agressividade "exagerada" do algoritmo FRED. Estas configurações possuem um valor muito baixo para max_{th} (25) assim como para a diferença entre este e min_{th}. A análise dos registros mostra um percentual nulo de pacotes marcados como *out* por falta de fichas no balde, mostrando a forte atuação do algoritmo FRED no controle da distribuição de fichas. Porém, devido ao baixo valor de max_{th}, existe um alto percentual (sempre em torno de 20%) de pacotes marcados como *out* pelo fato do tamanho médio da fila

[17] Um marcador eficaz em termos de justiça deve atingir um alto percentual de pacotes marcados como *out* pela ação do algoritmo justo de distribuição de fichas. Isto porque no caso ideal, a concessão de cada ficha deve ser feita sob a "fiscalização" deste algoritmo.

[18] A localização destes grampos dentro no código do algoritmo FRED encontra-se no apêndice B, juntamente com um número identificador.

[19] Este percentual é calculado em relação ao número total de pacotes que foram marcados como *out* pela ação do algoritmo justo de distribuição de fichas.

ultrapassar este parâmetro (modo *drop tail*).

As configurações 8, 18 e 24 já apresentam uma melhora significativa em relação aos resultados das configurações 9, 19 e 25. Isto se deve ao aumento da agressividade do algoritmo FRED com a diminuição do valor de max_{th} de 100 para 75. Porém, o valor de 50 para min_{th} e max_q ainda é relativamente alto, retardando a entrada na fase de prevenção ao congestionamento e diminuindo o poder de punição para os fluxos mais robustos (de maior RTT) quando nesta fase.

Vale notar que a maioria dos problemas acima se manifestam com maior intensidade para valores maiores de taxa contratada ($CIR > 50\%$), onde o fluxo de pacotes na fila de rastros é maior para um mesmo intervalo de tempo. A figura 5.7 mostra os resultados para as configurações 27, 21, 1, 11, 8, 18 e 24.

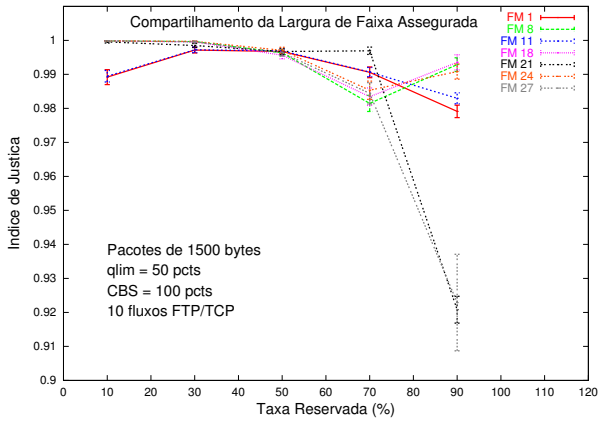
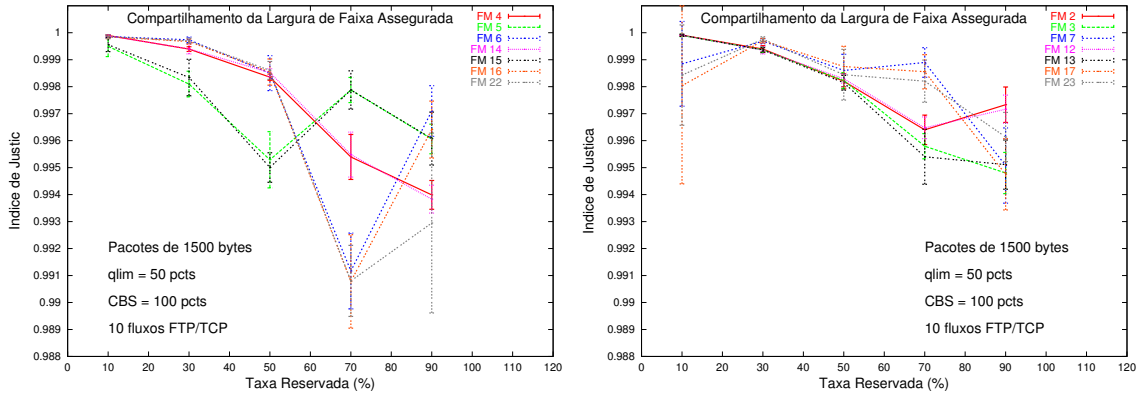


Figura 5.7: Cenário TCP heterogêneos sem CBR/UDP: resultados intermediários.

Finalmente, as quatorze configurações restantes possuem combinações de valores de min_{th} e max_{th} que proporcionam uma agressividade do FRED mais adequada para o cenário TCP heterogêneos sem CBR/UDP. Isto significa valores limitados para min_{th} (≤ 25) e min_q (≤ 15). Todos os resultados se situam acima de 0,99. São elas: 6, 16 e 22 com $min_{th} = 25$ e $max_{th} = 75$; 5 e 15 com $min_{th} = 25$ e $max_{th} = 50$, 4 e 14 com $min_{th} = 10$ e $max_{th} = 100$; 3 e 13 com $min_{th} = 10$ e $max_{th} = 75$; 7, 17 e 23 com $min_{th} = 25$ e $max_{th} = 100$; 2 e 12 com $min_{th} = 10$ e $max_{th} = 50$. As figuras 5.8a e 5.8b mostram os resultados para estas configurações. Pode-se notar que o desempenho cai com o aumento de CIR devido à dificuldade dos fluxos mais frágeis em acompanhar maiores taxas de marcação, mesmo para configurações mais adequadas do FM.

Portanto, pode-se concluir que mesmo num cenário não tão agressivo, o ajuste



(a) Configurações 4, 5, 6, 14, 15, 16 e 22.

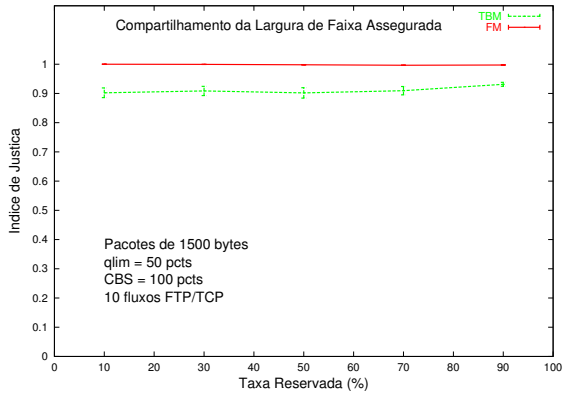
(b) Configurações 2, 3, 7, 12, 13, 17 e 23.

Figura 5.8: Cenário TCP heterogêneos sem CBR/UDP: melhores resultados.

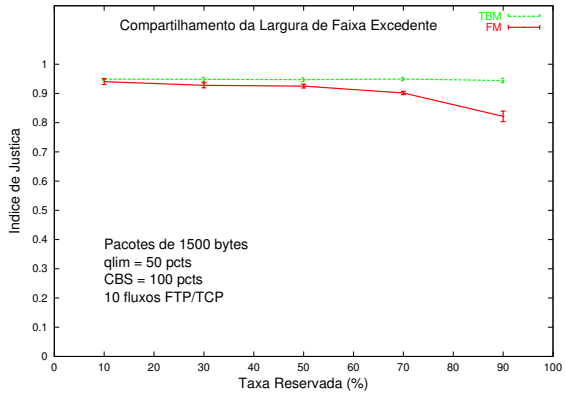
adequado dos parâmetros do FRED é necessário para obter melhores desempenhos. A tendência é que a diferença no desempenho entre as configurações se torne maior com o aumento na diferença dos RTTs dos fluxos competidores.

A título de comparação entre FM e TBM são apresentados outros resultados de ambos os marcadores (figura 5.9). A configuração escolhida para o FM é a de número de 2 (aquela com maior soma dos resultados para todos os valores de CIR).

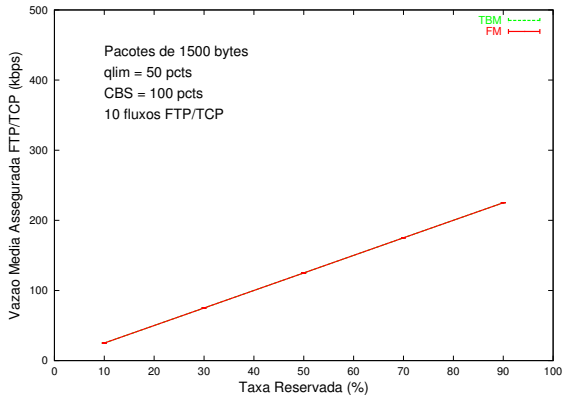
A figura 5.9a mostra o índice de justiça no compartilhamento da faixa assegurada. Confome visto anteriormente, a estratégia MAF, representada pelo FM, apresenta desempenho superior à estratégia MA, representada pelo TBM. O ganho de desempenho é justificado pelo controle por fluxo da distribuição das fichas. A figura 5.9b mostra que os desempenhos se equivalem para a justiça na largura de faixa de excedente, pois nenhum dos marcadores apresenta qualquer mecanismo para atuar no compartilhamento deste recurso. Esta figura também mostra a degradação de desempenho do FM para valores mais altos de CIR. Isto ocorre porque o FM faz com que o tráfego total dos fluxos mais frágeis tenham um maior percentual de tráfego assegurado em relação ao que no caso onde é utilizado o TBM. Em compensação, como não há um mecanismo adicional para a largura de faixa excedente, esta vantagem é "paga" através da concessão de parte de sua largura de faixa excedente para os fluxos mais robustos. A figura 5.9c mostra a vazão média assegurada dos fluxos TCP, evidenciando a eficiência da arquitetura do serviço assegurado em proteger o tráfego assegurado. O gráfico mostra ainda que a vazão média assegurada vale $CIR/10$, isto é, o valor contratado dividido pelo número de fluxos. Finalmente, a



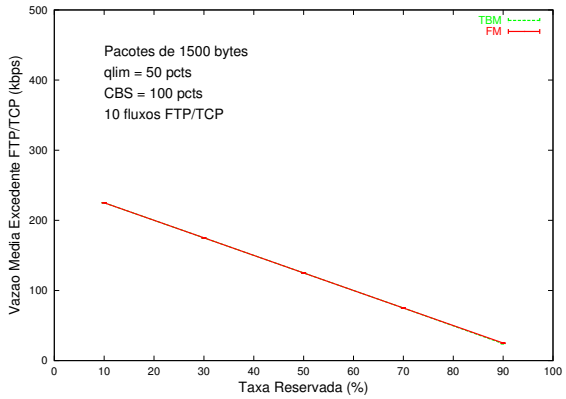
(a) Índice de justiça na largura de faixa assegurada.



(b) Índice de justiça na largura de faixa excedente.



(c) Vazão média assegurada dos fluxos TCP.



(d) Vazão média excedente dos fluxos TCP.

Figura 5.9: Cenário TCP heterogêneos sem CBR/UDP: FM x TBM.

vazão média excedente dos fluxos TCP (figura 5.9d) vale o que falta para completar a utilização completa do enlace gargalo.

5.2.2 Cenário TCP Homogêneos com CBR/UDP

A tabela 5.5 mostra os resultados obtidos para o cenário TCP homogêneos seguindo a mesma organização da tabela 5.4.

Comparando ligeiramente as duas tabelas, pode-se notar duas diferenças principais. A primeira é que os valores para os índices de justiça são mais baixos, caracterizando este cenário como mais agressivo que o anterior. A segunda é que há uma mudança significativa na ordem das configurações, o que indica que a presen-

Tabela 5.5: Índice de justiça no compartilhamento da largura de faixa assegurada. Cenário TCP homogêneos com CBR/UDP.

#	min_q	max_q	max_{th}	10%	30%	50%	70%	90%	Total
7	2	25	100	0,9901	0,9930	0,9881	0,9881	0,9925	4,9518
17	4	25	100	0,9896	0,9928	0,9890	0,9880	0,9922	4,9516
4	2	10	100	0,9983	0,9953	0,9879	0,9801	0,9785	4,9401
14	4	10	100	0,9980	0,9953	0,9879	0,9796	0,9789	4,9397
16	4	25	75	0,9931	0,9901	0,9801	0,9874	0,9837	4,9344
6	2	25	75	0,9924	0,9897	0,9809	0,9874	0,9831	4,9336
13	4	10	75	0,9983	0,9956	0,9876	0,9804	0,9655	4,9274
3	2	10	75	0,9980	0,9954	0,9879	0,9807	0,9637	4,9258
9	2	50	100	0,9762	0,9875	0,9843	0,9856	0,9914	4,9250
23	15	25	100	0,9781	0,9886	0,9863	0,9860	0,9839	4,9228
19	4	50	100	0,9729	0,9878	0,9830	0,9851	0,9914	4,9201
25	15	50	100	0,9775	0,9794	0,9855	0,9836	0,9850	4,9111
26	15	75	100	0,9823	0,9817	0,9825	0,9745	0,9741	4,8951
12	4	10	50	0,9964	0,9930	0,9845	0,9769	0,9398	4,8907
2	2	10	50	0,9947	0,9931	0,9846	0,9769	0,9340	4,8832
20	4	75	100	0,9800	0,9849	0,9786	0,9700	0,9631	4,8767
10	2	75	100	0,9813	0,9839	0,9802	0,9640	0,9622	4,8716
18	4	50	75	0,9809	0,9878	0,9737	0,9636	0,9407	4,8467
28	25	50	100	0,9654	0,9720	0,9678	0,9631	0,9711	4,8395
8	2	50	75	0,9663	0,9893	0,9746	0,9613	0,9359	4,8275
29	25	75	100	0,9624	0,9736	0,9624	0,9628	0,9659	4,8271
22	15	25	75	0,9867	0,9920	0,9684	0,9392	0,8844	4,7706
15	4	25	50	0,9204	0,9560	0,9350	0,9827	0,9657	4,7597
5	2	25	50	0,9170	0,9576	0,9344	0,9826	0,9651	4,7567
24	15	50	75	0,8385	0,9908	0,9646	0,9368	0,8850	4,6157
27	25	50	75	0,7540	0,9777	0,9497	0,9217	0,8045	4,4077
1	2	10	25	0,8405	0,8781	0,8621	0,8896	0,9245	4,3947
11	4	10	25	0,8736	0,8749	0,8194	0,8247	0,8744	4,2669
21	15	25	50	0,8700	0,8785	0,7686	0,6662	0,6218	3,8050
30	TBM			0,1101	0,1083	0,1064	0,1039	0,0976	0,5262

ça do tráfego não responsivo (pelo menos na intensidade utilizada) exige diferentes composições para os valores dos parâmetros de forma a melhorar o desempenho. A principal mudança se deve ao fato da transmissão constante de informação por parte do fluxo UDP elevar o nível médio da fila de rastros. Isto ocorre mesmo com o controle exercido pelo algoritmo FRED, na medida em que esta conexão tem sempre dados a transmitir e vai ocupar todo o espaço que lhe for permitido. Isto faz com que configurações com os valores mais baixos de max_{th} (2, 12, 5, 15, 1, 11, 21) apareçam bem mais abaixo na tabela neste cenário, por representarem uma agressividade exagerada.

Com base na tabela 5.5, as seguintes recomendações podem ser feitas a respeito da configuração do FM no cenário TCP homogêneos com CBR/UDP:

- valores de max_{th} mais baixos (e portanto mais próximos de min_{th}) degradam o desempenho do FM, o que pode ser verificado através dos seguintes blocos de configurações: [1, 2, 3 e 4] ($min_q = 2$ e $min_{th} = 10$), [5, 6 e 7] ($min_q = 2$ e $min_{th} = 25$), [11, 12 , 13 e 14] ($min_q = 4$ e $min_{th} = 10$) e [15, 16 e 17] ($min_q = 4$ e $min_{th} = 25$). Comparando entre si as configurações de cada bloco, percebe-se claramente a queda de desempenho conforme max_{th} dimunui de 100 até 25. Quando isto acontece, o nível médio máximo de pacotes na fila de rastros *in* decresce e com ele a capacidade de armazenamento de estados. Como o fluxo CBR/UDP sempre terá seu espaço na fila (reduzido no pior caso até *avgcq*), o percentual de pacotes *in* cai para o tráfego TCP e aumenta para o UDP, diminuindo a justiça. Além disso, valores baixos de max_{th} causam o aumento do percentual de pacotes marcados com *out* pela operação da fila de rastros no modo *drop tail*, o que ocorre de forma mais intensa para as configurações 1 e 11;

- $min_{th} = max_q$ não deve ser muito alto nem muito baixo em relação à max_{th}. No primeiro caso ($max_q = 50$ ou 75), o número de pacotes que podem ser marcados como *in* num $TBFT$ aumenta em demasia, diminuindo a capacidade do algoritmo FRED em penalizar o fluxo CBR/UDP. Esta queda de desempenho pode ser comprovada através dos seguintes blocos de configurações: [7, 9 e 10] ($min_q = 2$ e $max_{th} = 100$), [17, 19 e 20] ($min_q = 4$ e $min_{th} = 100$), [6 e 8] ($min_q = 2$ e $max_{th} = 75$), [16 e 18] ($min_q = 4$ e $max_{th} = 75$) e [23, 25 e 26] ($min_q = 15$ e $min_{th} = 100$). Comparando entre si as configurações de cada bloco, percebe-se a queda de desempenho conforme max_q aumenta de 25 para 50 ou 75. No segundo caso ($max_q = min_{th} = 10$), a agressividade do

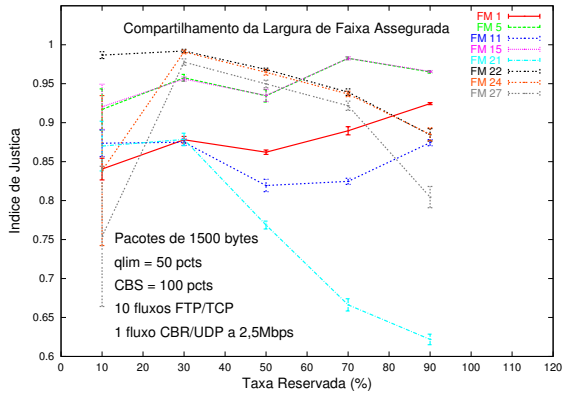
algoritmo FRED se torna relativamente alta, prejudicando também os fluxos TCP. Esta queda de desempenho pode ser comprovada através dos seguintes blocos de configurações: [4 e 7] ($min_q = 2$ e $max_{th} = 100$), [14 e 17] ($min_q = 4$ e $min_{th} = 100$), [3 e 6] ($min_q = 2$ e $max_{th} = 75$) e [13 e 16] ($min_q = 4$ e $max_{th} = 75$). Comparando entre si as configurações de cada bloco, percebe-se a queda de desempenho conforme $min_{th} = max_q$ diminui de 25 para 10;

- min_q não deve assumir valores muito altos para não permitir que o fluxo CBR/UDP possa ocupar impunimente um espaço desproporcional dentro da fila de rastros. A queda de desempenho pelo aumento do valor de min_q pode ser comprovada através dos seguintes blocos de configurações: [5, 15 e 21] ($min_{th} = 25$ e $max_{th} = 50$), [6, 16 e 22] ($min_{th} = 25$ e $max_{th} = 75$), [7, 17, 23] ($min_{th} = 25$ e $max_{th} = 100$), [8, 18, 24 e 27] ($min_{th} = 50$ e $max_{th} = 75$), [9, 19, 25 e 28] ($min_{th} = 50$ e $max_{th} = 100$), [10, 20, 26 e 29] ($min_{th} = 75$ e $max_{th} = 100$). Comparando entre si as configurações de cada bloco, percebe-se a queda de desempenho quando min_q assume os valores mais altos em cada bloco (15 ou 25).

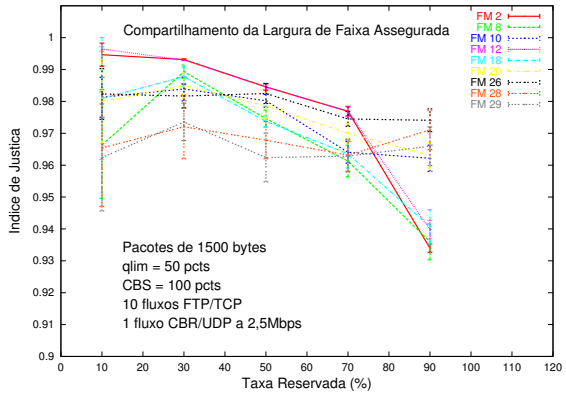
Os resultados de todas as configurações encontram-se na figura 5.10, agrupadas por desempenho.

De mesma forma que no cenário anterior, os resultados dos marcadores FM (configuração 7) e TBM são apresentados na figura 5.11.

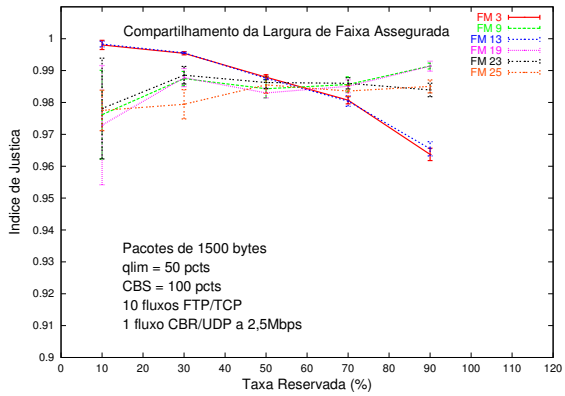
A figura 5.11a mostra o índice de justiça no compartilhamento da faixa assegurada. A estratégia MAF, representada pelo FM, obtém índices de justiça sempre acima de 0,98. Este desempenho é bastante superior ao da estratégia MA, representado pelo TBM, o qual mantém o índice pouco acima de 1/11 independentemente do valor de CIR. Isto significa que praticamente toda a largura de faixa assegurada é utilizada pelo fluxo CBR/UDP. Este ganho de desempenho é justificado pelo controle de fluxo da distribuição das fichas. A figura 5.11b mostra que os desempenhos se equivalem na medida que nenhum dos marcadores apresenta qualquer mecanismo para a divisão justa deste recurso. Novamente o FM apresenta uma queda de desempenho com o aumento de CIR devido aos mesmos motivos expostos para a figura 5.9b. As figuras 5.11c e 5.11e mostram a vazão média assegurada para os fluxos TCP e UDP, respectivamente. Estas curvas mostram que para o FM estes valores quase atingem os valores desejados (justos), o que não ocorre para o caso do TBM. A tabela 5.6 contém os valores numéricos. Finalmente, as figuras 5.11d



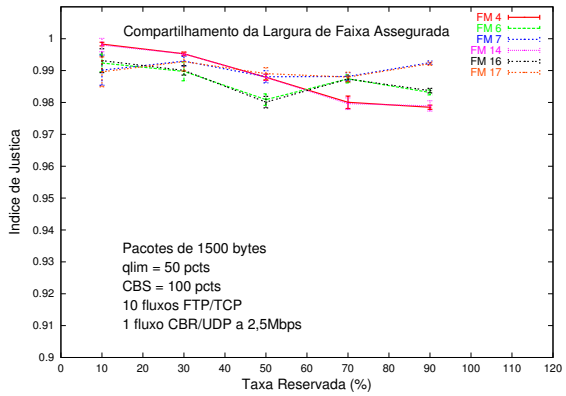
(a) Configurações 1, 5, 11, 15, 21, 22, 24 e 27.



(b) Configurações 2, 8, 10, 12, 18, 20, 26, 28 e 29.



(c) Configurações 3, 9, 13, 19, 23 e 25.

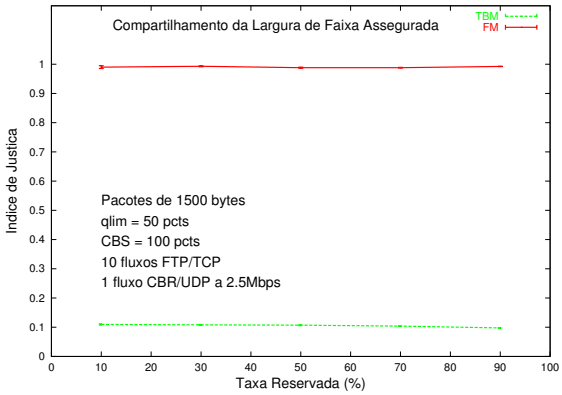


(d) Configurações 4, 6, 7, 14, 16 e 17.

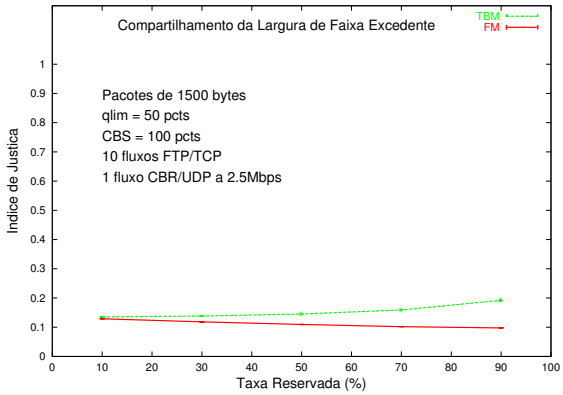
Figura 5.10: Cenário TCP homogêneos com CBR/UDP: todos os resultados.

e 5.11f mostram a vazão média excedente para os fluxos TCP e UDP, respectivamente. Estas figuras justificam os resultados da figura 5.11b, tendo em vista que para ambos os marcadores as vazões médias excedentes para cada tipo de tráfego são quase as mesmas.

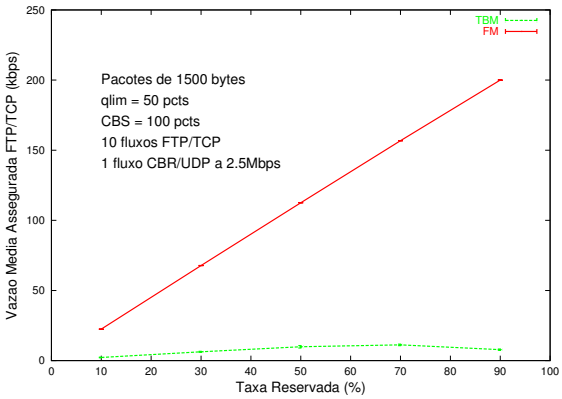
Também foram obtidos resultados para um terceiro cenário TCP heterogêneos com tráfego não responsivo, de forma a combinar as duas causas do problema da justiça. Os resultados foram bastante semelhantes aos do cenário TCP homogêneos com CBR/UDP em termos da ordenação das configurações por desempenho geral. Esta semelhança se deve ao fato da presença do tráfego não responsivo com uma taxa de 2,5Mbps ser uma causa bem mais agressiva do que a diferença de RTTs. Estes resultados encontram-se no apêndice C.



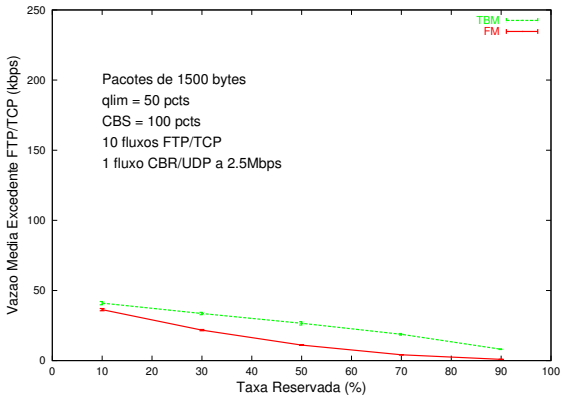
(a) Índice de justiça na largura de faixa assegurada.



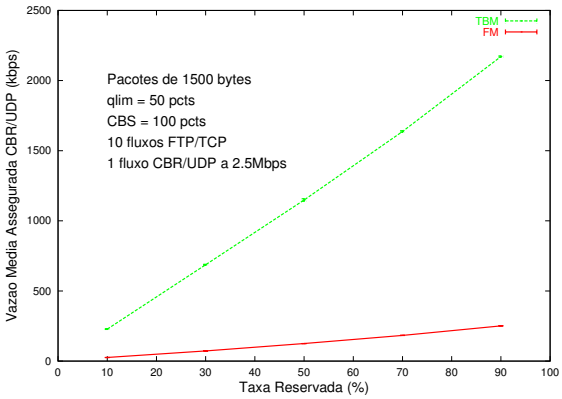
(b) Índice de justiça na largura de faixa excedente.



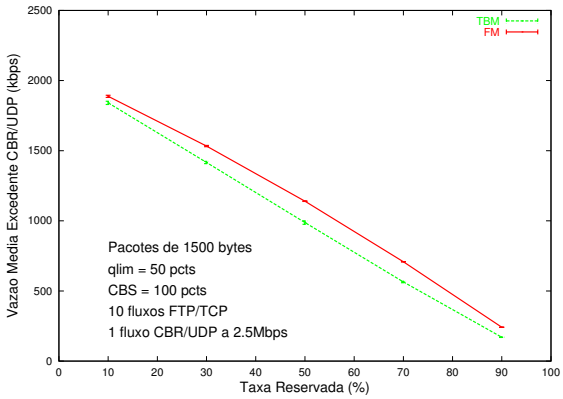
(c) Vazão média assegurada dos fluxos TCP.



(d) Vazão média excedente dos fluxos TCP.



(e) Vazão média assegurada do fluxo UDP.



(f) Vazão média excedente do fluxo UDP.

Figura 5.11: Cenário TCP homogêneos com CBR/UDP: FM x TBM.

Tabela 5.6: TCP homogêneos com CBR/UDP: resultados numéricos de vazão para a configuração 7.

Vazão média assegurada dos fluxos TCP					
	$CIR = 10\%$	$CIR = 30\%$	$CIR = 50\%$	$CIR = 70\%$	$CIR = 90\%$
TB	2,2	6,3	9,9	11,2	7,8
FM	22,5	67,8	112,6	156,8	200,0
Ideal	22,7	68,2	113,6	159,1	204,5
Vazão média assegurada dos fluxos TCP					
	$CIR = 10\%$	$CIR = 30\%$	$CIR = 50\%$	$CIR = 70\%$	$CIR = 90\%$
TB	227,8	686,5	1148,6	1636,7	2171,2
FM	25,7	72,7	125,0	183,3	251,1
Ideal	22,7	68,2	113,6	159,1	204,5

Adicionalmente, de forma a não limitar a validade do estudo a um número reduzido de 10 e 11 microfluxos, as mesmas simulações também foram geradas para os mesmos cenários com 100 e 101 conexões (10 conexões TCP de cada nó fonte F_i para cada nó destino D_i). Para tal, as filas nos roteadores assim como o tamanho do balde nos marcadores foram escalados por um fator de 10 ($qlim = 500$ e $CBS = 1000$). Quanto aos parâmetros do algoritmo FRED, os mesmos valores absolutos ($min_{th} = 2{,}4$) e percentuais em relação ao tamanho do balde foram utilizados (tabela 5.1). Os resultados encontram-se no apêndice C e levaram a duas conclusões:

- o resultado geral cai devido ao aumento do número de fluxos (outra causa de injustiça conforme discutido no capítulo 4);
- ocorre uma mudança bastante significativa na ordenação das configurações em função do desempenho total, confirmando as conclusões deste primeiro estudo onde os melhores ajustes variam de acordo com o cenário.

A principal conclusão deste primeiro estudo é que o FM apresenta um desempenho superior ao balde de fichas clássico TBM no que se refere ao compartilhamento da largura de faixa assegurada entre fluxos de um mesmo tráfego agregado. A diferença no desempenho tende a ser maior quanto mais agressivo for o cenário. Além disso, o desempenho do FM pode ser degradado em função de um ajuste inadequado dos parâmetros herdados do algoritmo FRED.

Finalmente, este estudo também permitiu observar que as melhores configurações não foram as mesmas em cada cenário e nem quando o número de fluxos foi aumentado. Isto favorece a pesquisa de um modelo adaptativo para a configuração dinâmica dos parâmetros do algoritmo FRED. Com isso poderia-se evitar ou diminuir a degradação do desempenho do FM com a mudança do número e da natureza das conexões ativas ao longo do tempo, o que acontece com frequência na Internet.

5.3 Segundo Estudo - Avaliação de Desempenho do TCFM

O primeiro estudo mostrou que os parâmetros do algoritmo FRED podem influenciar em muito o desempenho do FM, bem como comprovou a eficácia da estratégia MAF perante à MA na obtenção de justiça no compartilhamento da largura de faixa assegurada.

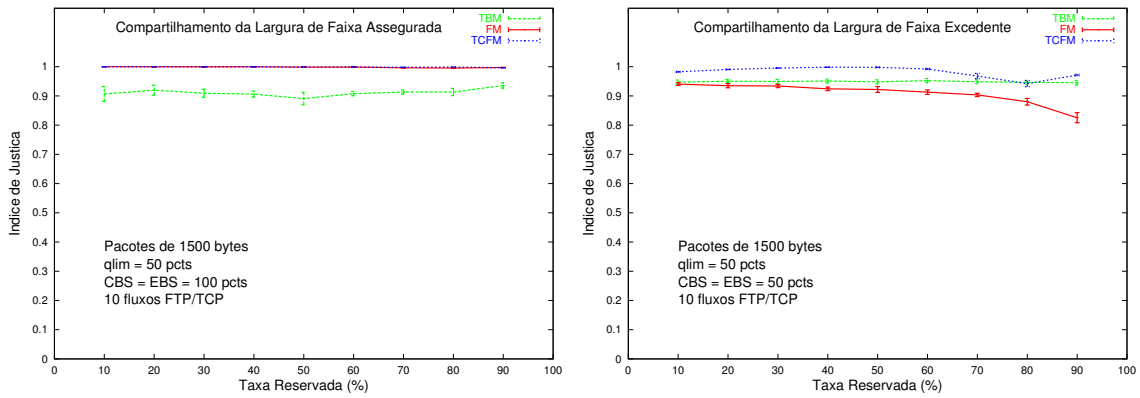
O objetivo deste segundo estudo é, através de alguns cenários, avaliar o desempenho do TCFM frente à proposta original FM, principalmente quanto à justiça no compartilhamento da largura de faixa excedente.

5.3.1 Influência do RTT

Este conjunto de simulações utiliza o mesmo cenário TCP heterogêneos sem CBR/UDP do primeiro estudo. Desta vez os valores de CIR variam de 10% até 90% do enlace gargalo, em incrementos de 10%. Os parâmetros do algoritmo FRED para os marcadores FM e TCFM são configurados de acordo com a configuração 2 (min_q = 2, $min_{th} = max_q = 10$, $max_{th} = 50$). Os baldes C (verde) e E (amarelo) possuem o mesmo tamanho dos baldes dos marcadores FM e TBM, isto é, 100 pacotes. A taxa de preenchimento do balde E, EIR, vale sempre 2,5Mbps (capacidade do enlace gargalo) - CIR, correspondendo portanto à fração do enlace gargalo não contratada pelo serviço assegurado.

A figura 5.12a mostra que o TCFM e o FM proporcionam o mesmo nível de justiça, ambos acima do TBM. Este resultado é esperado, na medida em que o FM e o TCFM possuem o mesmo mecanismo para proporcionar o compartilhamento justo da largura de faixa assegurada. Porém, a figura 5.12b mostra que o TCFM proporciona melhores resultados para o compartilhamento da largura de faixa excedente, devido ao mecanismo adicional de controle por fluxo da distribuição das

fichas amarelas no balde E. Vale lembrar que o índice de justiça na largura de faixa excedente é calculado utilizando as vazões de pacotes amarelos e vermelhos.



(a) Índice de justiça na largura de faixa assegurada.

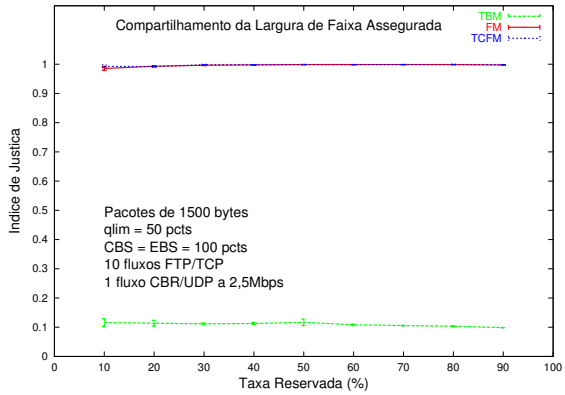
(b) Índice de justiça na largura de faixa excedente.

Figura 5.12: Cenário TCP heterogêneos sem CBR/UDP: TCFM x FM x TBM.

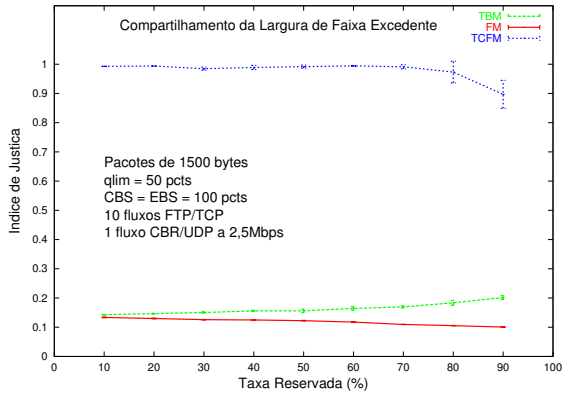
5.3.2 Influência do Tráfego Não Responsivo

Este cenário utiliza o mesmo cenário TCP homogêneos com CBR/UDP do primeiro estudo. Os parâmetros do algoritmo FRED para os marcadores FM e TCFM são configurados de acordo com a configuração 7 ($min_q = 2$, $min_{th} = max_q = 25$, $max_{th} = 100$).

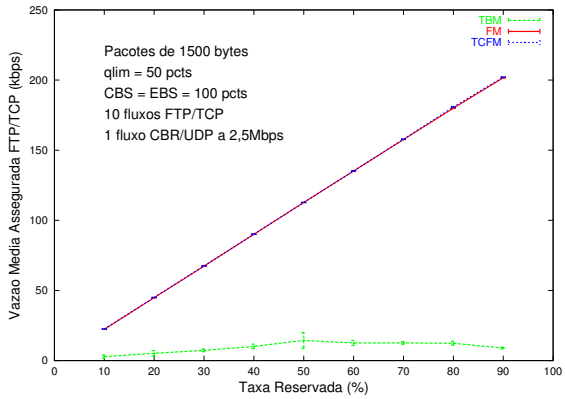
A figura 5.13a mostra que o TCFM e o FM proporcionam o mesmo nível de justiça, ambos muito acima do TBM devido à maior agressividade deste cenário. As figuras 5.13c e 5.13e evidenciam a proteção ao tráfego responsivo por parte do FM e TCFM. Estes marcadores garantem maior vazão de tráfego assegurado para os fluxos TCP ao mesmo tempo que restringem a vazão do fluxo UDP. Porém, a figura 5.13b mostra que o TCFM proporciona resultados muito superiores ao FM para o compartilhamento da largura de faixa excedente, tal como acontece entre o FM e o TBM para o compartilhamento da largura de faixa assegurada. As figuras 5.13d e 5.13f mostram que apenas o TCFM protege o tráfego excedente dos fluxos responsivos em detrimento do fluxo UDP.



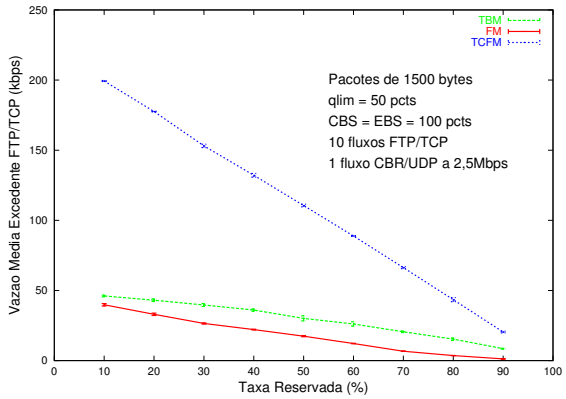
(a) Índice de justiça na largura de faixa assegurada.



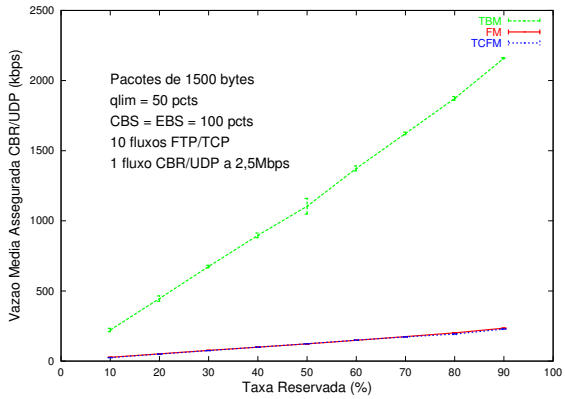
(b) Índice de justiça na largura de faixa excedente.



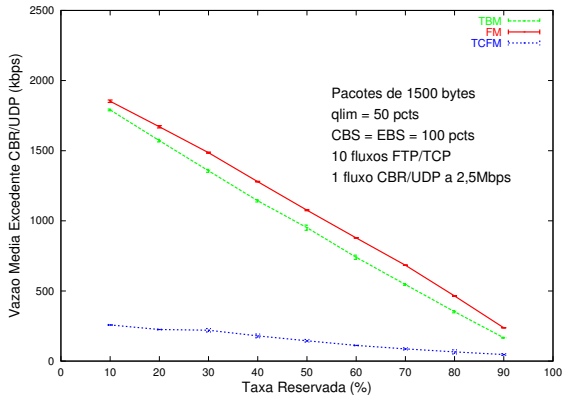
(c) Vazão média assegurada dos fluxos TCP.



(d) Vazão média excedente dos fluxos TCP.



(e) Vazão média assegurada do fluxo UDP.



(f) Vazão média excedente do fluxo UDP.

Figura 5.13: Cenário TCP homogêneos com CBR/UDP: TCFM x FM x TBM.

5.3.3 Influência do Número de Fluxos Ativos

Nesta subseção é analisada a influência do aumento do número de fluxos na justiça entre fluxos de um mesmo tráfego agregado. Conforme visto na subseção 4.2.3, o compartilhamento da largura de faixa obtida por um fluxo agregado por entre os seus microfluxos tende a ser menos uniforme quanto maior o número de fluxos.

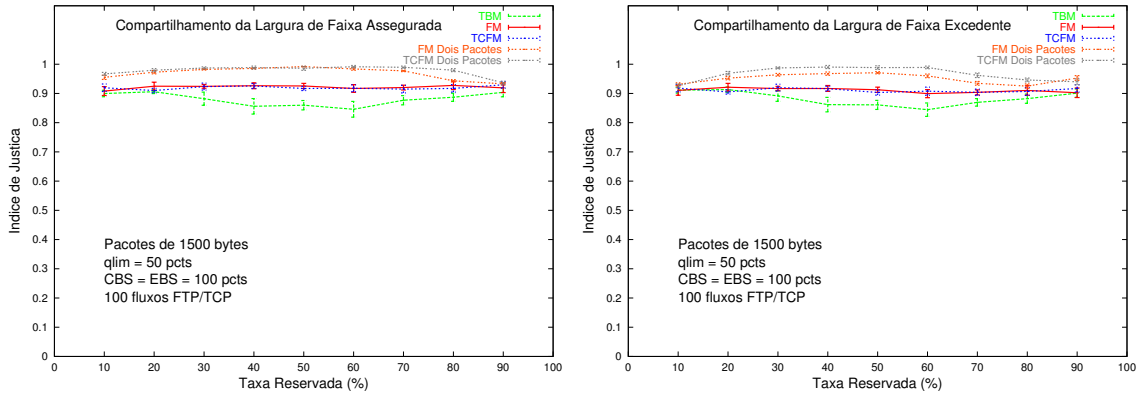
Para este estudo, são utilizados os mesmos cenários, porém com 100 (TCP heterogêneos sem CBR/UDP) e 101 (TCP homogêneos com CBR/UDP) conexões, 10 conexões TCP em cada nó. As filas nos roteadores, assim como o tamanho do balde nos marcadores, foram mantidas constantes a fim de caracterizar uma situação onde o número de fluxos é excessivo[20].

A figuras 5.14a e 5.14b mostram os índices de justiça nas larguras de faixa assegurada e excedente para o cenário TCP heterogêneos sem CBR/UDP. Comparando estes resultados com os das figuras 5.12a e 5.12b, nota-se que os desempenhos dos marcadores TCFM e FM caem ao nível do TBM. Isto ocorre porque cada fluxo tenta manter pelo menos um pacote em trânsito, aumentando o tamanho médio das filas RIO e RED com três níveis, fazendo-as operar na fase de controle de congestionamento. Com isso, os fluxos se revezam em *time-outs* de retransmissão, o que explica também a maior instabilidade do cenário evidenciada pelos maiores intervalos de confiança. Com relação aos marcadores justos, o mesmo acontece para a fila de rastros, fazendo com que o controle de distribuição das fichas perca a sua eficácia.

De modo a aliviar este efeito, o algoritmo FRED possui uma opção denominada modo de dois pacotes (*two-packet mode*) [98]. Neste modo, toda vez que o tamanho médio da fila avg ultrapassa max_{th}, cada fluxo passa a poder abrigar no máximo dois pacotes na fila. Os resultados mostram uma melhora para os marcadores FM e TCFM. Vale notar que a melhora no FM para o compartilhamento da largura de faixa assegurada se reflete também na faixa excedente, na medida em que os *time-outs* ficam distribuídos de forma mais uniforme, e assim mais fluxos passam ter acesso à largura de faixa excedente. Porém, o mecanismo adicional para o TCFM proporciona um desempenho ainda maior.

A figuras 5.15a e 5.15b mostram os índices de justiça nas larguras de faixa asse-

[20]Segundo MORRIS [121], este efeito começa a aparecer quando o número de fluxos supera o número de pacotes que cabe na memória da rede. Considerando o retardo médio em torno de dezenas de milisegundos, a memória da rede dos cenários utilizados fica em torno de 10 pacotes (13,125 pacotes utilizando uma latência de 50ms), sem contar o armazenamento extra fornecido pela fila do enlace gargalo (50 pacotes). Portanto, 100 fluxos é um número excessivo de fluxos, mesmo considerando o tamanho da fila do enlace gargalo.



(a) Índice de justiça na largura de faixa assegurada.

(b) Índice de justiça na largura de faixa excedente.

Figura 5.14: Cenário TCP heterogêneos sem CBR/UDP com 100 fluxos.

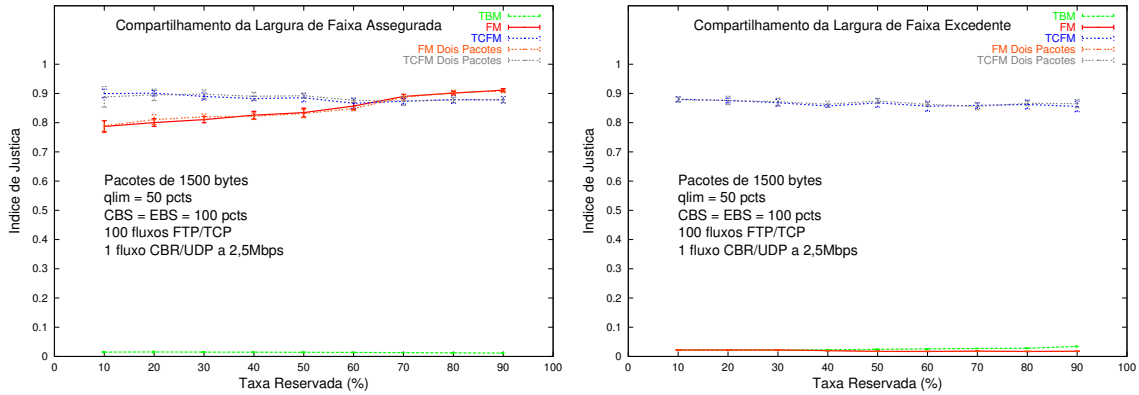
gurada e excedente para o cenário TCP homogêneos com CBR/UDP. Comparando estes resultados com os das figuras 5.13a e 5.13b, nota-se que os desempenhos dos marcadores TCFM e FM caem, da mesma forma como ocorreu para o cenário anterior. Porém, as formas variantes no modo de dois pacotes não proporcionam melhoria de desempenho, pois o tráfego não responsivo já é inibido pela operação normal do algoritmo FRED. Devido à sua alta agressividade, o tráfego tenta ultrapassar max_q. Toda vez que isto acontece, este fluxo fica limitado a $avgcq$ pacotes na fila. Devido ao alto número de fluxos, o controle por fluxo convencional chega a exercer o mesmo nível de inibição para o tráfego não responsivo.

Para encerrar este estudo, mais algumas simulações foram geradas para mostrar como o aumento do número de fluxos influencia a justiça negativamente. O valor de CIR foi fixado em 1,25Mbps (50% do enlace gargalo). Os número de fluxos foi variado de 50 até 950, de 50 em 50. Em cada cenário os fluxos foram divididos igualmente entre os 10 nós fonte.

Para o cenário TCP heterogêneos sem CBR/UDP (figura 5.16), o desempenho tanto para o índice de justiça no compartilhamento da largura de faixa assegurada como para a excedente, cai com o aumento do número de fluxos. Além disso, também em ambos os casos, o modo de dois pacotes proporciona uma melhora até 400 fluxos.

Para o cenário TCP homogêneos com CBR/UDP (figura 5.17), o desempenho também cai com o aumento do número de fluxos. Porém, tanto para o FM quanto para o TCFM, o modo de dois pacotes não apresenta nenhuma melhora.

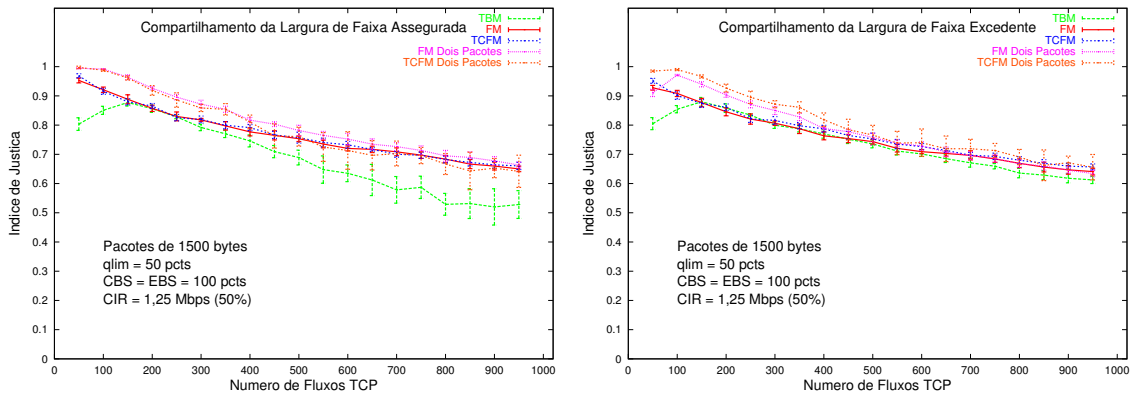
Este segundo segundo mostrou que o TCFM proporciona melhores índices de



(a) Índice de justiça na largura de faixa assegurada.

(b) Índice de justiça na largura de faixa excedente.

Figura 5.15: Cenário TCP homogêneos com CBR/UDP com 100 fluxos.



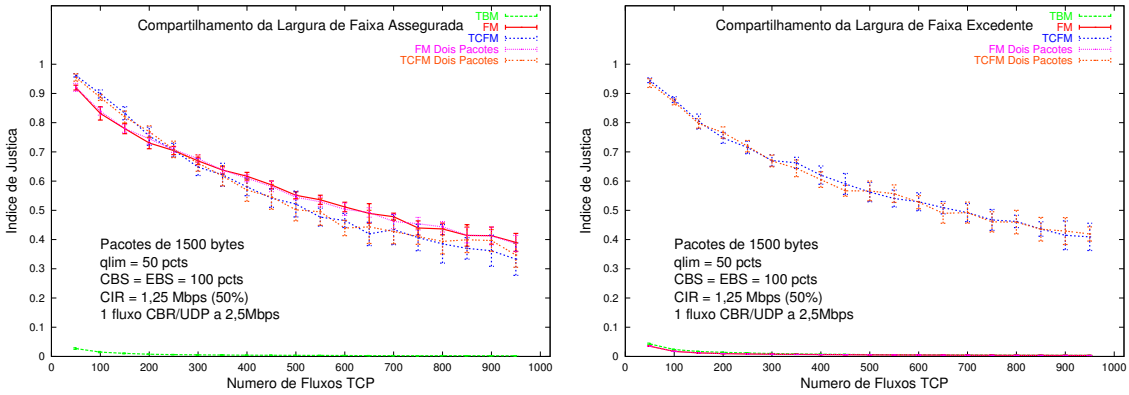
(a) Índice de justiça na largura de faixa assegurada.

(b) Índice de justiça na largura de faixa excedente.

Figura 5.16: Cenário TCP heterogêneos sem CBR/UDP: variação do número de fluxos.

justiça do que o FM para o compartilhamento da largura de faixa excedente, em cenários com fluxos TCP de diferentes RTTs e também na presença de tráfego responsivo. O aumento do desempenho se deve ao mecanismo de marcação adicional para dividir de forma justa a largura de faixa excedente, correspondente à taxa EIR que preenche o balde E do marcador TCFM.

No entanto, à medida que o número de fluxos aumenta, o algoritmo FRED vai se tornando ineficiente no combate à injustiça. Os resultados mostram também que



(a) Índice de justiça na largura de faixa assegurada.

(b) Índice de justiça na largura de faixa excedente.

Figura 5.17: Cenário TCP homogêneos com CBR/UDP: variação do número de fluxos

o modo de dois pacotes consegue aliviar estes efeitos apenas na ausência do tráfego não-responsivo. Mesmo assim, a eficácia deste modo de operação alternativo também decresce com o aumento do número de fluxos. A razão para a queda de desempenho é que o modo de dois pacotes força o TCP à trabalhar com uma janela de congestionamento pequena. Para as implementações mais comuns do TCP, tais como Tahoe, Reno e New Reno, é preciso esperar por três reconhecimentos duplicados até que um pacote perdido seja retransmitido. Sendo assim, o TCP nestes casos é forçado a entrar em *time-out* para se recuperar de descartes de pacotes quando opera com pequenas janelas. Algumas versões do TCP apresentam modificações para atacar este problema [98, 126].

Finalmente, fica evidente que o desempenho da estratégia MAF quanto à obtenção de justiça entre fluxos de um mesmo tráfego agregado será função do algoritmo escolhido. Com os cenários vistos neste trabalho e levando em conta a infinidade de possibilidades no mundo real, fica reforçada a necessidade de obtenção de algoritmos de marcação adaptativos, de forma a permitir que os condicionadores de tráfego possam ajustar-se à diferentes composições de tráfego ao longo do tempo.

Capítulo 6

Conclusão

Este capítulo final conclui este trabalho enumerando as suas principais contribuições. A seguir, são apresentadas as conclusões finais, dificuldades encontradas e limitações. Finalmente, alguns temas para trabalhos futuros são sugeridos.

6.1 Contribuições

Este trabalho permite destacar as seguintes contribuições:

- estruturação do problema da justiça no serviço assegurado, separando-o em dois problemas distintos e menores, facilitando o seu entendimento e análise;
- identificação e comparação entre as soluções propostas para o problema de justiça entre fluxos de um mesmo tráfego agregado, evidenciando as vantagens da utilização de marcadores justos;
- identificação, classificação e comparação das estratégias de marcação (MA, MF e MAF) para a obtenção de justiça no serviço assegurado;
- extensão do marcador FM para a obtenção de melhores resultados quanto à justiça no compartilhamento da largura de faixa excedente para fluxos de um mesmo tráfego agregado (TCFM). Além de sua principal contribuição, esta foi também a principal motivação deste trabalho;
- implementação de marcadores e filas no NS-2, permitindo a geração de resultados de simulações para este trabalho, assim como para outros que possam ser realizados no futuro utilizando os mesmos mecanismos ou similares;

- estudo do desempenho do algoritmo FRED no marcador FM, permitindo o entendimento do impacto de cada parâmetro na justiça entre os fluxos de um mesmo tráfego agregado;

- estudo do desempenho do marcador TCFM quanto à eficácia na obtenção de justiça entre fluxos de um mesmo agregado, principalmente no compartilhamento da largura de faixa excedente, em vários cenários distintos e comparativamente aos marcadores TBM e FM.

6.2 Conclusões

As seguintes conclusões podem ser tiradas através das argumentações e dos resultados apresentados neste trabalho:

- dentre as soluções para o problema da justiça entre fluxos de um mesmo agregado apresentadas na literatura, a utilização de estratégias de marcação que apliquem a justiça apresenta vantagens em relação às demais. A proposta TCP de duas janelas, apesar de fornecer proteção na obtenção da taxa reservada, requer modificações no protocolo, necessita de mecanismos adicionais para que o nó fonte receba informações do condicionador de tráfego à respeito da marcação de cada pacote, não garante a obtenção da taxa excedente e não endereça a questão da influência do tráfego não responsivo. O uso de disciplinas de gerenciamento ativo que protejam os fluxos TCP mais frágeis (maiores RTTs) e penalizem os fluxos não responsivos nos nós DS pode apresentar problemas de ordem escalar, pois no interior de um domínio DS o nível de agregação se torna maior. Além disso, complexidade é adicionada ao interior da rede, o que vai de encontro à filosofia DiffServ. Enquanto isso, a utilização de marcadores justos representa uma alternativa escalável na medida em que estes mecanismos atuam nas bordas da rede. Além disso, permite constantes evoluções dos algoritmos de marcação, cujas atualizações são menos impactantes do que nos casos anteriores. Finalmente, é mais flexível na medida em que soluções específicas podem ser desenvolvidas dependendo do tipo de tráfego a ser condicionado em um domínio restrito;

- a estratégia MAF apresenta vantagens em relação às estratégias MA e MF, sendo a mais indicada para a atacar o problema da justiça entre fluxos de um mesmo agregado de tráfego. Dentre as vantagens frente à estratégia MF,

destacam-se a ausência dos problemas de ineficiência no aproveitamento da largura de faixa assegurada e a capacidade de lidar com um número imprevisível de fluxos. Além disso, a estratégia de MAF permite ainda que novos algoritmos de marcação e distribuição justa de fichas sejam desenvolvidos, preservando os seus benefícios e a arquitetura da solução;

- o FM, implementado utilizando o algoritmo FRED, apresenta um desempenho superior ao balde de fichas clássico TBM no que se refere ao compartilhamento da largura de faixa assegurada entre fluxos de um mesmo tráfego agregado;

- o desempenho do FM é função do ajuste adequado dos parâmetros do algoritmo FRED e também do cenário (topologia, número de fluxos, composição do tráfego, etc.). O algoritmo FRED se mostrou eficiente em cenários com fluxos TCP de diferentes RTTs e também na presença de tráfego não responsivo;

- o TCFM proporciona melhores índices de justiça do que o FM para o compartilhamento da largura de faixa excedente em cenários com fluxos TCP de diferentes RTTs e também na presença de tráfego responsivo;

- à medida que o número de fluxos aumenta, o algoritmo FRED vai se tornando ineficiente no combate à injustiça. Além disso, a capacidade do modo de dois pacotes em aliviar estes efeitos se mostrou ineficiente em cenários com tráfego não-responsivo e à medida que o número de fluxos continua aumentando (embora este última limitação estar ligada à dinâmica do TCP);

- o desempenho da estratégia MAF quanto à obtenção de justiça entre fluxos de um mesmo tráfego agregado é função do algoritmo escolhido. A variação do desempenho apresentada pelos marcadores justos em função dos diferentes cenários favorece a pesquisa de um modelo adaptativo para a configuração dinâmica dos parâmetros do algoritmo FRED;

- finalmente, independente do mecanismo de marcação (baseado em balde de fichas ou taxa) e do algoritmo de distribuição justa de fichas ou recursos utilizado, fica evidente que para a obtenção de justiça no compartilhamento da largura de faixa excedente, há a a necessidade de um marcador que possua um mecanismo específico para atingir este objetivo.

6.3 Limitações e Dificuldades Encontradas

As principais limitações e dificuldades encontradas no trabalho foram:

- grande possibilidade de variação dos parâmetros e cenários, tornando a abrangência dos estudos realizados limitada, apesar destes terem fornecido dados e conclusões bastante representativas[1].

- a dinâmica dos algoritmos de controle de fluxo e congestionamento do TCP, do RED e do FRED, as quais contribuem para o aumento da complexidade e dificultam a análise dos resultados;

- falta de um outra técnica de avaliação de desempenho, tal como experimentos ou uso de modelos matemáticos, recomendável para complementar os resultados obtidos. Porém, além de fora do escopo proposto para o trabalho, experimentos exigiriam um grande esforço de programação para implementar os mecanismos básicos da arquitetura DiffServ em sistemas operacionais e equipamentos de rede. Já os modelos matemáticos ficariam extremamente complexos devido aos algoritmos adaptativos de controle e congestionamento do TCP, das filas RED com múltiplos níveis nos roteadores e do algoritmo FRED nos marcadores. Por esta razão, poucos trabalhos foram produzidos com modelos matemáticos em DiffServ, tendo abordados mecanismos simples de marcação tais como o TSW [39] e o balde fichas clássico [106]. Além disso, como pode ser visto no capítulo 3, apenas a interação entre o TCP e o RED é motivo de vários estudos até os dias de hoje, alguns deles com resultados conflitantes devido às diferentes simplificações feitas para viabilizar a obtenção dos modelos.

- o não monitoramento de algumas outras grandezas, tais como tamanho de filas, variáveis dos algoritmos FRED e RED, e o número de perdas para cada fluxo, as quais ilustrariam melhor as análises. Porém, além da quantidade excessiva de dados, o registro destes valores exigiria mais recursos de armazenamento e tempo de processamento de entrada e saída, aumentando em demasia o tempo necessário para as simulações.

[1] Praticamente todos os estudos nesta área utilizaram cenários bastante similares aos deste trabalho.

6.4 Sugestões para Trabalhos Futuros

Os seguintes temas ficam como recomendações para trabalhos futuros:

- propostas de modificações no algoritmo FRED, de forma a torná-lo adaptativo quanto aos ajustes do parâmetros e modos de operação, e de forma a melhorar o desempenho dos marcadores justos propostos;
- propostas de novos mecanismos de marcação e gerenciamento de ativo de filas que possam melhorar ainda mais a justiça;
- análise específica do desempenho das diferentes implementações do TCP dentro de cenário do serviço assegurado, e partir disso propostas de modificações que possam, por exemplo, melhorar a operação com janelas pequenas;
- estudo de outras métricas de desempenho no serviço assegurado, tais como retardo e jitter;
- avaliação dos mecanismos do serviço assegurado em outros cenários, como por exemplo outros tipos de tráfego TCP, outras composições de tráfego, topologias com múltiplos gargalos, etc.;
- investigação do problema da justiça entre fluxos agregados, de forma a garantir a coexistência de tráfego pertencente a diferentes contratos dentro de um mesmo domínio DS.

Referências Bibliográficas

[1] MATHY, L., EDWARDS, C., HUTCHISON, D., "The Internet, a Global Telecommunications Solution?", *IEEE Network Magazine*, v. 14, n. 4, pp. 46-57, Jul. 2000.

[2] COMER, D.E., *Internetworking with TCP/IP Volume I: Principles, Protocols and Architecture*. 2 ed. New Jersey, New York, Prentice Hall Inc., 1995.

[3] FERGUSON, P., HUSTON, G., "Quality of service in the Internet: fact, fiction or compromise?". *Eighth Annual Internet Society Conference (INET'98)*, n. 003, Geneva, Switzerland, 21-24 Jul. 1998.

[4] DUTTA-ROY, A., "The Cost of Quality in Internet-Style Networks", *IEEE Spectrum*, v. 37, n. 9, pp. 57-62, Sep. 2000.

[5] XIAO, X., NI, L.M., "Internet QoS: The Big Picture", *IEEE Network Magazine*, v. 13, n. 2, pp. 1-13, Mar. 1999.

[6] CALLON, R., DOOLAN, P., FELDMAN, N., et al., "A framework for multiprotocol label switching". *Internet Draft*, Sep. 1999. `draft-ietf-mpls-framework-05`.

[7] ROSEN, E.C., VISWANATHAN, A., CALLON, R., "Multiprotocol label switching architecture". *Internet RFC 3031*, Jan. 2001.

[8] RAJAGOPALAN, B., MA, Q., "An overlay model for constraint-based routing". *Internet Draft*, Jan. 1999. `draft-rajagopalan-CR-overlay-00`.

[9] AWDUCHE, D.O., CHIU, A., ELWALID, A., et al., "A framework for traffic engineering". *Internet Draft*, Mar. 2001. `draft-ietf-tewg-framework-03`.

[10] BRADEN, R., CLARK, D., SHENKER, S., "Integrated services in the Internet architecture: an overview". *Internet RFC 1633*, Jun. 1994.

[11] NICHOLS, K., BLAKE, S., BAKER, F., et al., "Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers". *Internet RFC 2474*, Dec. 1998.

[12] BLAKE, S., BLACK, D.L., CARLSON, M., et al., "An architecture for differentiated services". *Internet RFC 2475*, Dec. 1998.

[13] REKHTER, Y., ROSEN, E.C., "Carrying label information in BGP–4". *Internet Draft*, Jan. 2000. `draft-ietf-mpls-bgp4-mpls-04`.

[14] AWDUCHE, D.O., BERGER, L., GAN, D.-H., et al., "RSVP-TE: extensions to RSVP for LSP tunnels". *Internet Draft*, Aug. 2000. `draft-ietf-mpls-rsvp-lsp-tunnel-07`.

[15] ANDERSON, L., DOOLAN, P., FELDMAN, N., et al., "LDP specification". *Internet Draft*, Aug. 2000. `draft-ietf-mpls-ldp-11`.

[16] JAMOUSSI, B., ABOUL-MAGD, O., ANDERSON, L., et al., "Constraint-based LSP setup using LDP". *Internet Draft*, Jul. 2000. `draft-ietf-mpls-cr-ldp-04`.

[17] ALMQUIST, P., "Type of service in the Internet protocol suite". *Internet RFC 1349*, Jul. 1992.

[18] WANG, Z., CROWCROFT, J., "Quality of Service Routing for Supporting Multimedia Applications", *IEEE Journal on Selected Areas in Communications (JSAC)*, v. 14, n. 7, pp. 1228-1234, Sep. 1996.

[19] CRAWLEY, E., NAIR, R., RAJAGOPOLAN, B., et al., "A framework for QoS-based routing in the Internet". *Internet RFC 2386*, Aug. 1998.

[20] ZHANG, Z., SANCHEZ, C., SALKEWICZ, B., et al., "Quality of service extensions to OSPF or quality of service path first routing (QOSPF)". *Internet Draft*, Sep. 1997. `draft-zhang-qos-ospf-01`.

[21] APOSTOLOPOULOS, G., GUERIN, R., KAMAT, S., et al., "QoS routing mechanisms and OSPF extensions". *Internet Draft*, Dec. 1998. `draft-guerin-qos-routing-ospf-04`.

[22] MOY, J., "OSPF version 2". *Internet RFC 2178*, Jul. 1997.

[23] XIAO, X., HANNAN, A., BAILEY, B., et al., "Traffic Engineering with MPLS in the Internet", *IEEE Network Magazine*, v. 14, n. 2, pp. 28-33, Mar. 2000.

[24] MALCOLM, J., AGOGBUA, J., O'DELL, M., et al., "Requirements for traffic engineering over MPLS". *Internet RFC 2702*, Sep. 1999.

[25] LI, T., SWALLOW, G., AWDUCHE, D.O., "IGP requirements for traffic engineering with MPLS". *Internet Draft*, Feb. 1999. `draft-li-mpls-igp-te-00`.

[26] WHITE, P.P., CROWCROFT, J., "The integrated services in the Internet: state of the art". In: *Proceedings of the IEEE*, v. 85, n. 12, pp. 1934-1946, Dec. 1997.

[27] SHENKER, S., PARTRIDGE, C., GUERIN, R., "Specification of guaranteed quality of service". *Internet RFC 2212*, Sep. 1997.

[28] WROCLAWSKI, J., "Specification of the controlled-load network element service". *Internet RFC 2211*, Sep. 1997.

[29] ZHANG, L., DEERING, S., ESTRIN, D., et al., "RSVP: A New Resource Reservation protocol", *IEEE Network Magazine*, v. 7, n. 5, pp. 8-19, Sep. 1993.

[30] BRADEN, R., ZHANG, L., BERSON, S., et al., "Resource reservation protocol (RSVP) - version 1 functional specification". *Internet RFC 2205*, Sep. 1997.

[31] WROCLAWSKI, J., "The use of RSVP with IETF Integrated Services". *Internet RFC 2210*, Sep. 1997.

[32] WHITE, P.P., "RSVP and Integrated Services in the Internet: A Tutorial", *IEEE Communications Magazine*, v. 35, n. 5, pp. 100-107, May 1997.

[33] MANKIN, A., BAKER, F., BRADEN, B., et al., "Resource reservation protocol (RSVP) version 1 applicability statement some guidelines on deployment". *Internet RFC 2208*, Sep. 1997.

[34] STOICA, I., ZHANG, H., "Providing guaranteed services without per flow management". In: *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'99)*, pp. 81-94, Cambridge, Massachusetts, USA, Sep. 1999.

[35] HUSTON, G., "Next steps for the IP QoS architecture". *Internet Draft*, Aug. 2000. `draft-iab-qos-02`.

[36] RAJAN, R., VERMA, D., KAMAT, S., et al., "A Policy Framework for Integrated and Differentiated Services in the Internet", *IEEE Network Magazine*, v. 13, n. 5, pp. 36-41, Sep. 1999.

[37] DURHAM, D., BOYLE, J., COHEN, R., et al., "The COPS (common open policy service) protocol". *Internet RFC 2478*, Jan. 2000.

[38] VERMA, D., *Supporting Service Level Agreements on IP Networks*. 1 ed. Indianapolis, Indiana, Macmillan Technical Publishing, 1999.

[39] SAHU, S., TOWSLEY, D., KUROSE, J., "A quantitative study of differentiated services for the Internet". In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM'99)*, v. 3, pp. 1808-1817, Rio de Janeiro, RJ, Brasil, Dec. 1999.

[40] REN, H., PARK, K., "Towards a theory of differentiated services". In: *Proceedings of the IEEE/IFIP Eighth International Workshop on Quality of Service (IWQoS 2000)*, pp. 211-220, Pittsburgh, Pennsylvania, USA, Jun. 2000.

[41] SEMRET, N., LIAO, R.R.-F., CAMPBELL, A.T., et al., *Market Pricing of Differentiated Internet Services*. Technical Report CU/CTR/TR 503-98-37, Columbia University Center for Telecommunications Research, 1998.

[42] ODLYZKO, A., "Paris metro pricing: the minimalist differentiated services solution". In: *Proceedings of the IEEE/IFIP Seventh International Workshop on Quality of Service (IWQoS'99)*, pp. 159-161, London, England, Jun. 1999.

[43] ZIVIANI, A., *Voz sobre Serviços Diferenciados na Internet*. Tese de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1999.

[44] SHIN, J., KIM J.-W., JAY KUO, C.-C., "Content-based packet video forwarding mechanism in differentiated services networks". In: *Proceedings of the IEEE Packet Video Workshop (PV2000)*, Cagliari, Sardinia, Italy, May 2000.

[45] BLESS, R., WEHRLE, K., "IP multicast in differentiated services networks". *Internet Draft*, Sep. 1999. `draft-bless-diffserv-multicast-00`.

[46] DOVROLIS, C., STILIADIS, D., "Relative differentiated services in the Internet: issues and mechanisms". In: *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'99)*, pp. 204-205, Atlanta, Georgia, USA, May 1999.

[47] DOVROLIS, C., STILIADIS, D., RAMANATHAN, P., "Proportional differentiated services: delay differentiation and packet scheduling". In: *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'99)*, pp. 109-120, Cambridge, Massachusetts, USA, Sep. 1999.

[48] DOVROLIS, C., RAMANATHAN, P., "Proportional differentiated services, part II: loss rate differentiation and packet dropping". In: *Proceedings of the IEEE/IFIP Eighth International Workshop on Quality of Service (IWQoS 2000)*, pp 53-61, Pittsburgh, Pennsylvania, USA, Jun. 2000.

[49] JACOBSON, V., NICHOLS, K., PODURI, K., "An expedited forwarding PHB". *Internet RFC 2598*, Jun. 1999.

[50] HEINANEN, J., BAKER, F., WEISS, W., et al., "Assured forwarding PHB group". *Internet RFC 2597*, Jun. 1999.

[51] CLARK, D., WROCLAWSKI, J., "An approach to service allocation in the Internet". *Internet Draft*, Jul. 1997. `draft-clark-diff-svc-alloc-00`.

[52] CLARK, D.D., FANG, W., "Explicit Allocation of Best Effort Packet Delivery Service", *IEEE/ACM Transactions on Networking*, v. 6, n. 4, pp. 362-373, Aug. 1998.

[53] HEINANEN, J., FINLAND, T., GUERIN, R., "A single rate three color marker". *Internet RFC 2697*, Sep. 1999.

[54] HEINANEN, J., FINLAND, T., GUERIN, R., "A two rate three color marker". *Internet RFC 2698*, Sep. 1999.

[55] FANG, W., SEDDIGH, N., NANDY, B., "A time sliding window three color marker (TSWTCM)". *Internet RFC 2859*, Jun. 2000.

[56] IBANEZ, J., NICHOLS, K., "Preliminary simulation evaluation of an assured service". *Internet Draft*, Aug. 1998. `draft-ibanez-diffserv-assured-eval-00`.

[57] SEDDIGH, N., NANDY, B., PIEDA, P., et al., "An experimental study of assured services in a diffserv IP QoS network". In: *Proceedings of SPIE*, v. 3529, pp. 217-230, Dec. 1998.

[58] NANDY, B., SEDDIGH, N., PIEDA, P., "Diffserv's assured forwarding PHB: what assurance does the customer have?". In: *Proceedings of the Ninth International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'99)*, Basking Ridge, New Jersey, USA, Jun. 1999.

[59] REZENDE, J.F., "Assured service evaluation". In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM'99)*, v. 1a, pp. 100-104, Rio de Janeiro, RJ, Brasil, Dec. 1999.

[60] SEDDIGH, N., NANDY, B., PIEDA, P., "Bandwidth assurance issues for TCP flows in a differentiated services network". In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM'99)*, v. 3, pp. 1792 -1798, Rio de Janeiro, RJ, Brasil, Dec. 1999.

[61] GOYAL, M., DURRESI, A., JAIN, R., et al., "Performance analysis of assured forwarding". *Internet Draft*, Feb. 2000. `draft-goyal-diffserv-afstdy-00`.

[62] KIM, H., "A fair marker". *Internet Draft*, Apr. 1999. `draft-kim-fairmarker-diffserv-00`.

[63] ALVES, I.B.H.A., REZENDE, J.F., MORAES, L.F.M., "Evaluating fairness in aggregated traffic marking". In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM'00)*, v. 1, pp. 445-449, San Francisco, California, USA, Nov. 2000.

[64] GROSSMAN, D., "New Terminology for Diffserv". *Internet Draft*, Mar. 2001. `draft-ietf-diffserv-new-terms-04`.

[65] BERNET, Y., BLAKE, S., GROSSMAN, D., et al., "An informal management model for diffServ routers". *Internet Draft*, Feb. 2001. `draft-ietf-diffserv-model-06`.

[66] KIM, B., KIM, B.-K., "Simulation study of weighted round-robin queueing policy". In: *Proceedings of the Technical Conference on Telecommunications Research and Development*, Lowell, Massachusetts, USA, Oct. 1994.

[67] FLOYD, S., JACOBSON, V., "Link-Sharing and Resource Management Models for Packet Networks", *IEEE/ACM Transactions on Networking*, v. 3, n. 4, pp. 365-386, Aug. 1995.

[68] JACOBSON, V., "Differentiated services architecture". *Talk in the Int-Serv WG at the IETF Meeting*, Munich, Germany, Aug. 1997.

[69] ZIVIANI, A., REZENDE, J.F., DUARTE, O.C.M.B., "Towards a differentiated services support for voice traffic". In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM'99)*, v. 1a, pp. 59-63, Rio de Janeiro, RJ, Brasil, Dec. 1999.

[70] NASSER, H., LEON-GARCIA, A., ABOUL-MAGD, O., "Voice over differentiated services". *Internet Draft*, Dec. 1998. `draft-naser-voice-diffserv-eval-00`.

[71] CLARK, D., "Adding Service Discrimination to the Internet", *Telecommunications Policy*, v. 20, n. 3, pp. 169-181, Apr. 1996.

[72] NICHOLS, K., JACOBSON, V., ZHANG, L., "A two-bit differentiated services architecture for the Internet", *Internet RFC 2638*, Jul. 1999.

[73] NICHOLS, K., BLAKE, S., "Differentiated services operational model and definitions". *Internet Draft*, Feb. 1998. `draft-nichols-dsopdef-00`.

[74] WANG, Z., "User-share differentiation (USD) scalable bandwidth allocation for differentiated services". *Internet Draft*, Nov. 1997. `draft-wang-diff-serv-usd-00`.

[75] BAUMGARTNER, F., BRAUN, T., HABEGGER, P., "Differentiated services: a new approach for quality of service in the Internet". *Eighth IFIP TC6 Conference on High Performance Networking (HPN'98)*, Session T1: Next Generation Internet, Vienna, Austria, 21-25 Sep. 1998.

[76] BASU, A., WANG, Z., *A Comparative Study of Schemes for Differentiated Services*. Technical Report, Bell Laboratories, Lucent Technologies, 1998.

[77] DOVROLIS, C., RAMANATHAN, P., "A Case for Relative Differentiated Services and the Proportional Differentiation Model", *IEEE Network Magazine*, v. 13, n. 5, pp. 26-34, Sep. 1999.

[78] TEITELBAUM, B., HARES, S., DUNN, L., et al., "Internet2 QBone – Building a Testbed for Differentiated Services", *IEEE Network Magazine*, v. 13, n. 5, pp. 8-16, Sep. 1999.

[79] SEMRET, N., LIAO, R.R.-F., CAMPBELL, A.T., et al., *Peering and Provisioning of Differentiated Internet Services*. Technical Report, Columbia University Center for Telecommunications Research, 1999.

[80] YEOM, I., REDDY, A.N., "Realizing throughput guarantees in a differentiated services network". In: *Proceedings of the IEEE International Conference on Multimedia Computing and Systems (ICMCS'99)*, v. 2, pp. 372-376, Florence, Italy, Jun. 1999.

[81] GOYAL, M., DURRESI, A., MISRA, P., et al., "Effect of number of drop precedences in assured forwarding". In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM'99)*, v. 1a, pp. 188-193, Rio de Janeiro, RJ, Brasil, Dec. 1999.

[82] EELOUMI, O., CNODDER, S.D., PAUWELS, K., "Usefullness of three drop precedences in assured forwarding service". *Internet Draft*, Jul. 1999. `draft-ellomi-diffserv-threevstwo-00`.

[83] SEDDIGH, N., NANDY, B., PIEDA, P., "Study of TCP and UDP interaction for the AF-PHB". *Internet Draft*, Aug. 1999. `draft-nsbnpp-diffserv-tcpudpaf-01`.

[84] PIEDA, P., SEDDIGH, N., NANDY, B., "The dynamics of TCP and UDP interaction in IP-QoS differentiated service networks". In: *Proceedings of the Third Canadian Conference on Broadband Research (CCBR'99)*, Ottawa, Canada, Nov. 1999.

[85] FENG, W., KANDLUR, D.D., SAHA, D., et al., "Adaptive packet marking for providing differentiated services in the Internet". In: *Proceedings of the IEEE Internacional Conference on Network Protocols (ICNP'98)*, pp. 108-117, Austin, Texas, USA, Oct. 1998.

[86] TANENBAUM, S., *Computer Networks*. 3 ed. New Jersey, New York, Prentice Hall Inc., 1996.

[87] STEVENS, W., "TCP slow start, congestion avoidance, fast retransmit and fast recovery algorithms". *Internet RFC 2001*, Jan. 1997.

[88] FLOYD, S., JACOBSON, V., "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, v. 1, n. 4, pp. 397-413, Aug. 1993.

[89] JACOBSON, V., KARELS, M.J., "Congestion avoidance and control". In: *Proceedings of the ACM SIGCOMM Symposium on Communication Architectures and Protocols (SIGCOMM'88)*, pp. 314-329, Stanford, California, USA, Aug. 1988.

[90] JACOBSON, V., "Berkeley TCP evolution from 4.3-Tahoe to 4.3-Reno". In: *Proceedings of the Eighteenth Internet Engineering Task Force*, pp. 363-366, Vancouver, Canada, Jul. 1990.

[91] FLOYD, S., HENDERSON, T., "The new reno modification to TCP's fast recovery algorithm". *Internet RFC 2582*, Apr. 1999.

[92] MATHIS, O., MAHDAVI, J., FLOYD, S., et al., "TCP selective acknowlegdment options". *Internet RFC 2018*, Oct. 1996.

[93] BRADEN, B., CLARK, D., CROWCROFT, J., et al., "Recommendations on queue management and congestion avoidance in the Internet". *Internet RFC 2309*, Apr. 1998.

[94] MAY, M., BOLOT, J., DIOT, C., et al., "Reasons not to Deploy RED". In: *Proceedings of the IEEE/IFIP Seventh International Workshop on Quality of Service (IWQoS'99)*, pp. 260-262, London, England, Jun. 1999.

[95] CHRISTIANSEN, M., JEFFAY, K., OTT, D., et al., "Tuning RED for web traffic". In: *Proceedings of ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM 2000)*, pp. 139-150, Stockholm, Sweden, Sep. 2000.

[96] BONALD, T., MAY, M., BOLOT, J.-C., "Analytic evaluation of RED performance". In: *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, v. 3, pp. 1415-1424, Tel Aviv, Israel, Mar. 2000.

[97] ABOUZEID, A.A., ROY, S., "Analytic understanding of RED gateways with multiple competing TCP flows". In: *Proceedings of the IEEE Global Communications Conference (GLOBECOM'00)*, v. 1, pp. 555-560, San Francisco, California, USA, Nov. 2000.

[98] LIN, D., MORRIS, R., "Dynamics of random early detection". In: *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectu-*

res, and Protocols for Computer Communication (SIGCOMM'97), pp. 127-137, Cannes, France, Sep. 1997.

[99] ANJUM, F.M., TASSIULAS, L., *Balanced-RED: An Algorithm to Achieve Fairness in the Internet*. Technical Report CSHCN TR 99-9, Center for Satellite and Hybrid Communication Networks, 1999.

[100] OTT, T.J., LAKSHMAN, T.V., WONG, L.H., "SRED: stabilized RED". In: *Proceedings of the Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99)*, v. 3, pp. 1346-1355, New York, New York, USA, Mar. 1999.

[101] FENG, W., KANDLUR, D., SAHA, D., et al.,"A self-configuring RED gateway". In: *Proceedings of the Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'99)*, v. 3, pp. 1320-1328, New York, New York, USA, Mar. 1999.

[102] FIROIU, V., BORDEN, M., "A study of active queue management for congestion control". In: *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, v. 3, pp. 1435-1444, Tel Aviv, Israel, Mar. 2000.

[103] HOLLOT, C.V., MISRA, V., TOWSLEY, D., et al., "A control theoretic analysis of RED". In: *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2001)*, v. 3, pp. 1510-1519, Anchorage, Alaska, USA, Apr. 2001.

[104] http://www.aciri.org/floyd/REDparameters.txt

[105] ANDRIKOPOULOS, I., PAVLOU, G., "A fair traffic conditioner for the assured service in a differentiated service internet". In *Proceedings of IEEE International Conference on Communications (ICC 2000)*, v. 2, pp. 806-810, New Orleans, Louisiana, USA, Jun. 2000.

[106] SAHU, S., NAIN, P., TOWSLEY, D., et al., "On achievable service differentiation with token bucket marking for TCP". In: *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS'00)*, pp. 23-33, Santa Clara, California, USA, Jun. 2000.

[107] NANDY, B., SEDDIGH, N., PIEDA, P., et al., "Intelligent traffic conditioners for assured forwarding based differentiated services networks". In: *Proceedings of IFIP High Performance Networking Conference (HPN 2000)*, pp. 540-554, Paris, France, June 2000.

[108] YEOM, I., REDDY, A.L.N., "Impact of marking strategy on aggregated flows in a differentiated services network". In: *Proceedings of the IEEE/IFIP Seventh International Workshop on Quality of Service (IWQoS'99)*, pp. 156 -158, London, England, Jun. 1999.

[109] YEOM, I., REDDY, A.L.N., *Marking for QoS Improvement*. Technical Report, Texas A & M University, College Station, 1999.

[110] FENG, W., KANDLUR, D., SAHA, D., "Understanding and Improving TCP Performance over Networks with Minimum Rate Guarantees", *IEEE/ACM Transactions on Networking*, v. 7, n. 2, pp. 173-187, Apr. 1999.

[111] YEOM, I., REDDY, A.L.N., *Modeling TCP Behavior in a Differentiated Services Network*. Technical Report, Texas A & M University, College Station, 1999.

[112] ZHANG, L., SHENKER, S., CLARK, D.D., "Observations on the dynamics of a congestion control algorithm: the effects of two-way traffic". In: *Proceedings of the ACM SIGCOMM Conference Communications Architectures and Protocols (SIGCOMM'91)*, pp. 133-147, Zurich, Switzerland, Sep. 1991.

[113] FEROZ, A., RAO, A., KALYANARAMAN, S., "A TCP-friendly traffic marker for IP differentiated services". In: *Proceedings of the IEEE/IFIP Eighth International Workshop on Quality of Service (IWQoS 2000)*, pp 138-147, Pittsburgh, Pennsylvania, USA, Jun. 2000.

[114] BAUMGARTNER, F., BRAUN, T., SIEBEL, C., "Fairness of assured service". In: *Proceedings of the Thirteenth European Simulation Multiconference (ESM'99)*, Warsaw, Poland, Jun. 1999.

[115] BONAVENTURE, O., DE CNODDER, S., "A rate adaptive shaper for differentiated services". *Internet RFC 2963*, Oct. 2000.

[116] FLOYD, S., FALL, K., "Promoting the Use of End-to-End Congestion Control in the Internet", *IEEE/ACM Transactions on Networking*, v. 7, n. 4, pp. 458-472, Aug. 1999.

[117] DEMERS, A., KESHAV, S., SHENKER, S., "Analysis and simulation of a fair-queueing algorithm". In: *Proceedings of the ACM SIGCOMM Symposium on Communications Architectures and Protocols (SIGCOMM'89)*, pp. 1-12, Austin, Texas, USA, Sep. 1989,

[118] MCKENNEY, P.E., "Stochastic fairness queueing". In: *Proceedings of the Ninth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'90)*, v. 2, pp. 733-740, San Francisco, California, USA, Jun. 1990.

[119] JAIN, R., *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. 1 ed. New York, New York, John Wiley and Sons Inc., 1991.

[120] ALVES, I.B.H.A., REZENDE, J.F., MORAES, L.F.M., "Avaliando a justiça na marcação de tráfegos agregados". In: *Anais do XVIII Simpósio Brasileiro de Redes de Computadores (SBRC'2000)*, pp. 119-134, Belo Horizonte, MG, Brasil, May 2000.

[121] MORRIS, R., "TCP behavior with many flows". In: *Proceedings of the IEEE Internacional Conference on Network Protocols (ICNP'98)*, pp. 205-211, Atlanta, Georgia, USA, Oct. 1997.

[122] FLOYD, S., FALL, K., "Simulation-based Comparisons of Tahoe, Reno and Sack TCP", *Computer Communication Review*, v. 26, n. 3, pp. 5-21, Jul. 1996.

[123] CHOUDHURY, A., HAHNE, E., "Dynamic queue length thresholds in a shared memory ATM switch". In: *Proceedings of the Fifteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'96)*, pp. 679-687, San Francisco, California, USA, Mar. 1996.

[124] FALL, K., VARADHAN, K., *ns v2 – Notes and Documentation*, Technical Report, The VINT (Virtual InterNetwork Testbed) Project, UC Berkeley, LBL, USC/ISI & Xerox PARC, 1997.

[125] JAIN, R.K., CHIU, D.-M.W., HAWE, W.R., "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System", Research Report TR-301, DEC, 1984.

[126] BRAKMO, L.S., O'MALLEY, S.W., PETERSON, L.L., "TCP vegas: new techniques for congestion detection and avoidance". In: *Proceedings of the ACM SIGCOMM Conference on Communications Architectures, Protocols and Applications (SIGCOMM'94)*, pp. 24-35, London, England, Aug. 1994.

[127] INFORMATION SCIENSES INSTITUTE, UNIVERSITY OF SOUTHERN CALIFORNIA, "Transmission control protocol". *Internet Standard 7 (Internet RFC 793)*, Sep. 1981.

[128] KARN, P., PARTRIDGE, C. "Estimating round-trip times in reliable transport protocols". In: *Proceedings of the ACM Workshop on Frontiers in Computer Communications Technology (SIGCOMM'87)*, pp. 2-7, Stowe, Vermont, USA, Aug. 1987.

[129] BRADEN, R., "Requirements for internet hosts – communication layers". *Internet RFC 1122*, Oct. 1989.

Apêndice A

TCP: Controle de Congestionamento e Implementações

Neste apêndice, serão descritos os mecanismos de controle de congestionamento do protocolo TCP, assim como quatro de suas implementações mais comuns: Tahoe, Reno, New Reno e SACK.

A.1 Introdução

O TCP (*Transmission Control Protocol*) é um dos protocolos da camada de transporte da arquitetura TCP/IP. É orientado à conexão e transmite fluxos de dados não estruturados (*stream de bytes*), de forma simultânea e em ambos os sentidos (*full-duplex*). Suas três principais diferenças para o UDP (*User Datagram Protocol*) são a capacidade de prover confiabilidade na entrega dos dados, o controle de fluxo fim-a-fim e o controle de congestionamento.

Para prover confiabilidade, o TCP utiliza os mecanismos de reconhecimento e retransmissão. No nó fonte, para cada segmento (bloco de dados) enviado, um temporizador é disparado. No nó destino, pacotes de reconhecimento são enviados para os segmentos recebidos[1]. Quando ocorre o esgotamento de um temporizador sem que o seu reconhecimento tenha chegado (*timeout*), o segmento correspondente é retransmitido[2]. Além disso, reconhecimentos e segmentos são numerados para que

[1] Normalmente os reconhecimentos são atrasados (*delayed ACKs*) para que um pacote reconheça mais de um segmento, aumentando a eficiência da conexão.

[2] O mecanismo de reconhecimento do TCP é dito positivo, em oposição ao caso onde o destino implementa os temporizadores e envia reconhecimentos (negativos) na ocorrência de *timeouts* para segmentos não recebidos.

fonte e destino possam identificá-los corretamente. Desta forma, o TCP garante que eventualmente todos os segmentos serão recebidos.

Para evitar que a fonte transmita um segmento de cada vez após o reconhecimento do segmento anterior, o TCP utiliza a largura de faixa de forma mais eficiente através do mecanismo de janela deslizante (*sliding window*). Nesta técnica, o nó fonte transmite vários segmentos sem esperar por um reconhecimento. Conforme os segmentos iniciais vão sendo reconhecidos, o TCP desloca a sua janela transmitindo novos segmentos. O mecanismo é operado numerando-se todos os bytes do *stream* de dados, sequencialmente, de forma que os segmentos contêm o número do primeiro byte transmitido, e os reconhecimentos carregam o número do último byte recebido[3]. Sendo assim, o tamanho da janela é medido em bytes e corresponde à diferença entre último e primeiro bytes transmitidos e não reconhecidos. De forma similar, o nó destino mantém uma janela deslizante para o *buffer* de recepção.

Para prover o controle de fluxo fim-a-fim, o TCP permite ainda que o tamanho da sua janela de transmissão varie com o tempo. O nó destino, ao enviar os reconhecimentos, indica a quantidade máxima de bytes que pode receber no seu *buffer*, denominada janela anunciada (*advertised window* - *awnd*). Para que o nó fonte não sobrecarregue o destino, o tamanho da sua janela de transmissão varia de acordo com o valor anunciado, nunca podendo ultrapassá-lo. Este mecanismo elimina a possibilidade de descartes no nó destino, aumentando a eficiência da conexão.

A.2 O Controle de Congestionamento do TCP

Se o controle de fluxo fim-a-fim evita que a fonte envie mais dados do que o destino pode armazenar, não é suficiente para evitar perdas através da rede. Isto porque, mesmo que as fontes não sobrecarreguem seus destinos, nada impede que

[3]Mais precisamente, nas três primeiras implementações que serão apresentadas na seção A.3, o reconhecimento é acumulativo. Isto é, o destino anuncia o último byte de uma sequência contínua de dados recebidos. Logo, se algum segmento for perdido, podem surgir "buracos" no *buffer* de recepção. Neste caso, os bytes de número de sequência posteriores à lacuna não serão reconhecidos, ainda que possam ser armazenados. Além disso, reconhecimentos duplicados serão enviados para cada segmento recebido, até que o segmento perdido seja transmitido com sucesso, quando então todos os bytes da lacuna e após esta serão reconhecidos. Este esquema, embora simples, faz com que o nó fonte não tenha conhecimento de todas as transmissões realizadas com sucesso, o que pode levar a retransmissões desnecessárias de segmentos já recebidos. Esta deficiência é um dos motivos para a introdução de reconhecimento seletivo (SACK - *Selective ACKnowledgement*) no TCP [122].

sobrecarreguem os nós intermediários. É necessário portanto um mecanismo que previna o congestionamento (esgotamento de recursos no interior da rede), reduzindo o número de descartes e contribuindo também para a eficiência das conexões.

No entanto, a introdução do controle de congestionamento foi estimulada por um outro motivo. Conforme visto anteriormente, o TCP provê confiabilidade utilizando a técnica de reconhecimento e retransmissão. Logo, ele responde ao congestionamento retransmitindo pacotes. Sem algum mecanismo de controle, estas retransmissões podem gerar mais descartes, que por sua vez irão gerar mais retransmissões, agravando indefinidamente o congestionamento. Este quadro instável, denominado colapso de congestionamento, leva à rede a baixas taxas de utilização. Os colapsos de congestionamento começaram a ocorrer na Internet a partir de outubro de 1986 [89], ameaçando o seu crescimento. Isto motivou a modificação do TCP de forma a prevenir e controlar estas situações.

Para que este problema seja solucionado, o fluxo de dados de cada conexão TCP deve obedecer ao princípio da conservação dos pacotes [89], isto é, operar com janela de transmissão cheia e de forma estável. Nesta situação, nenhum pacote novo deve ser entregue à rede sem que outro tenha sido recebido no destino. A conservação dos pacotes pode falhar devido a três razões principais. A primeira é a incapacidade do protocolo de transporte de entrar em equilíbrio. A segunda é a injeção de um novo pacote na rede antes que outro a deixe. E a terceira é a impossibilidade de atingir o equilíbrio por limitações de recurso ao longo do caminho.

A seguir, serão descritas as técnicas fundamentais do controle de congestionamento do TCP que endereçam estas três questões. Outros mecanismos adicionais serão descritos juntamente com os tipos de TCP que os implementam.

A.2.1 Atingindo o Equilíbrio

O TCP é auto-ajustável (*self-clocking*) na medida em que o nó destino não pode gerar reconhecimentos a uma taxa maior do que a rede pode aceitar os pacotes. Além disso, conforme visto anteriormente, o nó fonte varia a janela de transmissão de acordo com o valor anunciado pelo nó destino na chegada dos reconhecimentos. Portanto, no equilíbrio, aproximadamente um pacote é gerado para cada um que deixa a rede, caso os temporizadores estejam bem regulados (subseção A.2.2).

Se a princípio manter o equilíbrio não é difícil, resta saber como atingi-lo. Com este propósito, um algoritmo denominado início lento (*slow start*) foi desenvolvido para iniciar ou reiniciar uma transmissão. A idéia consiste em elevar gradualmente

a taxa de injeção de tráfego na rede até que uma posição de equilíbrio seja atingida. Quatro modificações foram necessárias no código do TCP:

- criar uma nova variável chamada janela de congestionamento (*congestion window - cwnd*);
- no início de uma transmissão ou após uma perda, atualizar o valor de *cwnd* para o equivalente a um segmento;
- a cada reconhecimento não duplicado, aumentar a janela de congestionamento em um segmento;
- para permitir novas transmissões, considerar o tamanho da janela como o mínimo entre a janela anunciada e a janela de congestionamento (equação A.1).

$$janela = min(awnd, cwnd) \tag{A.1}$$

Apesar do seu nome, o algoritmo início lento proporciona um crescimento exponencial para a janela de transmissão até que o valor de *cwnd* se iguale ou ultrapasse *awnd*. JACOBSON e KARELS [89] mostraram a eficácia do início lento em reduzir o caráter oscilatório do TCP, fazendo com que a conexão entre em equilíbrio de forma suave.

A.2.2 Conservando o Equilíbrio

Para uma conexão em equilíbrio, um pacote será transmitido para cada um que tenha sido recebido caso os temporizadores estejam bem calibrados, ou seja, não se esgotem antes dos reconhecimentos chegarem.

A forma de atingir este objetivo é fazer boas estimativas do RTT, e utilizá-las para calcular o *timeout* de retransmissão ou RTO (*Retransmission Timeout*). Esta não é uma tarefa fácil na Internet devido a vários fatores tais como a heterogeneidade das redes interligadas, as mudanças nas tabelas de roteamento e as variações de retardo nos enlaces atravessados.

O TCP se acomoda às variações de retardo da Internet utilizando um algoritmo de retransmissão adaptativo, monitorando as conexões e calculando valores para o RTO que variam de acordo com o comportamento da rede. Primeiro, ele registra o instante de envio de um segmento e o instante de recebimento do respectivo reconhecimento. Em seguida, o TCP obtém uma nova amostra para o RTT através

da diferença entre estes valores. O cálculo final do RTT é feito então através de uma soma ponderada entre a nova amostra e o valor utilizado anteriormente. A equação A.2 mostra este cálculo onde RTT_M é o RTT médio estimado, RTT é o novo valor amostrado e α um fator suavizador entre 0.8 e 0.9 por exemplo [127].

$$RTT_M = \alpha * RTT_M + (1 - \alpha) * RTT \tag{A.2}$$

O valor do RTO é obtido do RTT através da equação A.3, onde UBOUND é um limite superior para o *timeout*, LBOUND um limite inferior e β um fator de variação do retardo. A princípio, um valor de β próximo de 1 favorece o aumento da vazão porque o RTO se aproxima do RTT. Porém, deve haver uma margem de segurança para evitar retransmissões desnecessárias devido a repentinos aumentos no RTT. A especificação do protocolo TCP [127] exemplifica o ajuste deste parâmetros da seguinte forma: β entre 1.3 to 2.0, UBOUND em 1 minuto e LBOUND em 1 segundo.

$$RTO = min(UBOUND, max(LBOUND, \beta.RTT_M)) \tag{A.3}$$

A princípio, todos estes cálculos parecem simples. Porém, o TCP utiliza um mecanismo de reconhecimentos acumulativos da sequência de bytes e não dos segmentos. Além disso, os reconhecimentos podem ser atrasados, não havendo portanto uma correspondência biunívoca com cada segmento transmitido. Outra dificuldade para a medição do RTT surge nas retransmissões, quando não há meios de saber se o reconhecimento que chega se refere ao segmento original ou ao retransmitido. Este fenômeno é denominado ambiguidade de reconhecimentos.

Assumir que o reconhecimento se refere tanto à primeira transmissão quanto à mais recente pode trazer problemas. Com a primeira opção, o RTT pode ser superestimado. Já a segunda opção pode causar problemas nos casos em que o RTT aumenta devido a um crescimento repentino no volume de tráfego. Ao enviar um segmento antes deste aumento do tráfego, o TCP utilizará um valor de RTT menor, causando *timeout* e retransmissão. Caso o reconhecimento referente ao primeiro pacote não seja perdido, ele pode chegar logo após a retransmissão fazendo com que a nova estimativa para o RTT decresça mais ainda. Implementações de TCP com este critério puderam ser observadas com um comportamento estável, no qual o RTT tem um valor ligeiramente menor do que a metade do RTT real. Assim, todos os pacotes são enviados duas vezes mesmo não havendo perdas [89].

Para solucionar o problema, o primeiro passo consiste em fazer com que o TCP ignore reconhecimentos referentes a pacotes retransmitidos. Mas somente esta mo-

dificação não é suficiente. No mesmo exemplo anterior em que uma transmissão é feita antes de um aumento no retardo da rede, o protocolo TCP utilizará um valor mais baixo de RTT e uma retransmissão poderá ocorrer. Isto fará com que o TCP continue trabalhando com estimativas reduzidas, levando a novas retransmissões enquanto o aumento no retardo persistir.

Portanto, deve haver um mecanismo para fazer com que o TCP consiga regular a estimativa do RTT em situações de retransmissão, sem ser levado a cometer erros. Este segundo passo consiste na introdução de uma estratégia de recuo (*backoff*) para o cálculo do valor do RTO, conforme descrito a seguir:

- inicialmente o valor para o RTO é calculado em função da estimativa do RTT, conforme descrito anteriormente;
- cada vez que o TCP retransmitir um pacote, ele aumenta o valor do RTO[4] conforme a equação A.4, onde o fator multiplicativo γ tipicamente vale 2 [2];

$$RTO' = \gamma.RTO \tag{A.4}$$

- o valor do RTO calculado pela estratégia de *backoff* é mantido para as transmissões subsequentes;
- quando um reconhecimento referente a um pacote transmitido uma única vez for recebido, o TCP passa novamente a calcular o RTO em função do RTT.

As modificações acima são conhecidas como algoritmo de Karn por causa de um dos seus autores [128].

Mesmo após estes aperfeiçoamentos o resultado não é satisfatório, pois os procedimentos descritos até o momento não permitem que as estimativas para o *timeout* de retransmissão acompanhem grandes variações de retardo na rede. Pode ser mostrado que com o valor sugerido de 2 para β na equação A.3, a estimativa para o valor do RTT poderá se adaptar bem com valores de utilização iguais ou inferiores a 30% [2].

JACOBSON e KARELS [89] propuseram modificações para atacar este problema. São estimados os valores médios tanto do RTT quanto da sua variação, sendo que o último substitui o parâmetro β no cálculo do RTO. As equações A.5, A.6 e A.7

[4]Conforme visto anteriormente, deve haver um teto para prevenir que o RTO cresça indefinidamente no caso de retransmissões sucessivas.

descrevem estes cálculos, onde $desvio_M$ é o desvio médio estimado, δ e ρ controlam a influência de uma nova amostra no RTT e desvio médios, respectivamente, e η controla a influência do desvio médio no RTO.

$$RTT_M = (1 - \delta).RTT_M + \delta.RTT \tag{A.5}$$

$$desvio_M = (1 - \rho)desvio_M + \rho.|RTT - RTT_M| \tag{A.6}$$

$$RTO = RTT_M + \eta.desvio_M \tag{A.7}$$

Os valores de δ e ρ são preferencialmente dados em inversos de potências de dois para tornar a computação eficiente. Pesquisas na área sugerem $\delta = 1/2^3$ e $\rho = 1/2^2$ [2]. O valor de η mudou de 2 no UNIX BSD 4.3 para 4 no UNIX BSD 4.4. Este algoritmo se chama Jacobson/Karels também devido aos seus autores [89], e melhorou o desempenho do TCP quanto às estimativas para o *timeout* de retransmissão. A especificação de 1989 do protocolo TCP [129] inclui os algoritmos de Karn e Jacobson.

A.2.3 Prevenção ao Congestionamento

Tendo atacado as duas primeiras condições de obediência à lei de conservação, resta fazer com que o equilíbrio não deixe de ser atingido por limitações de recursos ao longo do caminho. Para isso é necessário que o TCP se adapte a situações de congestionamento, reduzindo a geração de tráfego apropriadamente.

Para montar uma estratégia de prevenção ao congestionamento, dois elementos são necessários:

- um sinal para indicar o congestionamento;
- uma forma eficiente de diminuir o tráfego quando este sinal for recebido e de aumentá-lo na sua ausência.

Assumindo que os temporizadores estão bem calibrados, cada *timeout* pode ser considerado como uma indicação de descarte. Como a imensa maioria das perdas na Internet são devido ao esgotamento de recursos em algum roteador, um *timeout* também serve para indicar situações de congestionamento.

Nestes casos, o tamanho das filas tende a crescer exponencialmente. Logo, só haverá estabilidade caso o tráfego seja reduzido de forma tão ou mais rápida. Já

que o TCP pode controlar o tráfego através do tamanho da janela de transmissão, uma opção é reduzi-la de forma exponencial através da equação A.8. Este método é denominado decréscimo multiplicativo (*multiplicative decrease*).

$$cwnd = d.cwmd, 0 \leq d \leq 1 \tag{A.8}$$

Na ausência de congestionamento, o aumento do tráfego poderia ser feito através de um processo inverso. Contudo, isto faria com que o ponto de equilíbrio fosse ultrapassado no aumento e no decréscimo da janela de congestionamento, gerando um comportamento oscilatório[5]. Uma solução então é aumentar o tráfego de forma linear conforme a equação A.9. Este procedimento é denominado aumento aditivo (*additive increase*).

$$cwnd = cwmd + u \tag{A.9}$$

Os dois algoritmos combinados podem ser resumidos em três passos:

- a cada estouro de um temporizador, o valor da janela de congestionamento *cwnd* cai à metade do seu valor atual (decréscimo multiplicativo);
- a cada reconhecimento de um dado novo, o valor da janela de congestionamento é aumentado em $1/cwnd$ (aumento aditivo);
- a cada envio, considerar o tamanho da janela o mínimo entre a janela anunciada *awnd* e a janela de congestionamento *cwnd*.

Este algoritmo não inclui o início lento pois possui objetivos diferentes. O início lento visa fazer com que o equilíbrio possa ser atingido enquanto que o aumento aditivo e o decréscimo multiplicativo combinados visam fazer com que condições de congestionamento não consigam perdurar por muito tempo. No entanto, conforme será visto a seguir, estes algoritmos podem ser implementados em conjunto.

[5]De certa forma isto é intuitivo pois é muito mais fácil gerar um congestionamento do que saná-lo. Este fenômeno é conhecido como o efeito da hora do *rush* devido à analogia com o caso do tráfego de automóveis.

A.3 Algumas Implementações Comuns do TCP

A.3.1 TCP Tahoe

O TCP Tahoe acrescentou alguns algoritmos e refinamentos em relação às implementações anteriores. O início lento é combinado com um algoritmo de prevenção ao congestionamento, assim como é introduzido um outro algoritmo denominado retransmissão rápida (*fast retransmit*) [87]. Para os dois primeiros, duas variáveis são utilizadas: a janela de congestionamento *cwnd* e um limiar para início lento *ssthresh* (*slow start threshold*). O algoritmo combinado está descrito abaixo:

- no início de uma conexão: $cwnd = 1$ *segmento* e $ssthresh = 65536$ *bytes*;
- a janela de transmissão é igual ao mínimo entre as janelas anunciada e de congestionamento;
- quando o congestionamento ocorre sinalizado por um *timeout* ou pelo recebimento de três reconhecimentos duplicados, a variável *ssthresh* é atualizada com a metade do valor atual da janela de transmissão ou dois segmentos, o que for maior;
- quando algum reconhecimento novo é recebido, a janela de congestionamento *cwnd* é aumentada de duas maneiras distintas:

  - se o valor de *cwnd* for menor ou igual a *ssthresh*, a janela de congestionamento é aumentada conforme o algoritmo de início lento, isto é, de um segmento a cada reconhecimento recebido;

  - quando o valor de *cwnd* for maior do que *ssthresh*, prevalece o algoritmo de prevenção de congestionamento onde o incremento vale $MSS.(MSS/cwnd)$, sendo MSS e *cwnd* medidos em bytes.

Enquanto que o início lento provoca um crescimento exponencial da janela de congestionamento, a prevenção de congestionamento corresponde a um crescimento linear de no máximo um segmento por RTT. Para compreender este fato deve-se inicialmente notar que em uma janela há n segmentos, sendo $n = cwnd/MSS$. O valor novo de *cwnd* será então $cwnd' = cwnd + A.MSS.(MSS/cwnd)$, onde A é o números de reconhecimentos recebidos. Se toda a janela de dados enviada for recebida com sucesso, então o valor de A será o mesmo valor de n, o número de

segmentos enviados na janela. Sendo assim, o novo valor da janela aumentará em no máximo MSS bytes em um RTT, conforme mostra o desenvolvimento abaixo.

$$cwnd' = cwnd + n.\frac{MSS.MSS}{cwnd} = cwnd + \frac{cwnd}{MSS}.\frac{MSS.MSS}{cwnd} = cwnd + MSS \tag{A.10}$$

Além da utilização dos algoritmos de início lento e prevenção de congestionamento em conjunto, uma nova modificação foi introduzida no TCP Tahoe. O mecanismo de retransmissão rápida visa apressar a retransmissão de segmentos quando a rede se encontra em uma situação de congestionamento moderado.

No TCP, um reconhecimento de número n tem a conotação de que todos os segmentos até $n-1$ foram recebidos e aguarda-se a transmissão do segmento n. Se o transmissor já recebeu este reconhecimento e recebe outro de mesmo número n, isto significa que um segmento de número maior do que n foi recebido, mas não n. Estes reconhecimentos duplicados informam ao transmissor que um segmento foi recebido fora de ordem ou que houve uma perda. Como cada pacote é roteado de forma independente, é possível que alguns pacotes cheguem fora da ordem em que foram enviados. Logo, o TCP não sabe se o reconhecimento duplicado foi causado por um pacote perdido ou por uma simples reordenação. Assume-se que no caso de pacotes enviados fora de ordem, serão recebidos no máximo dois reconhecimentos duplicados, até que o segmento fora de ordem seja processado e um novo reconhecimento seja gerado. Sendo assim, um número de reconhecimentos duplicados maior do que este limite pode ser interpretado pelo TCP como um indicativo de perda de pacote.

A partir deste raciocínio, foi introduzido o algoritmo de retransmissão rápida, o qual apareceu pela primeira vez no UNIX BSD versão 4.3 Tahoe em 1998. Ao receber três reconhecimentos duplicados, o segmento correspondente é retransmitido imediatamente sem que se espere pelo estouro do seu temporizador, aumentando a vazão da conexão.

A.3.2 TCP Reno

A implementação Reno retém todos os melhoramentos introduzidos no TCP Tahoe. Porém, um novo algoritmo denominado recuperação rápida (*fast recovery*) é incorporado ao de retransmissão rápida. O objetivo é impedir que a janela de congestionamento caia a um segmento nos casos de retransmissão devido a três reconhecidos duplicados.

A premissa para introduzir esta modificação é que se os reconhecimentos continuam chegando, qualquer situação de congestionamento que causou a perda de pacotes já deve ter se esgotado. Portanto, não há necessidade de se reduzir o fluxo de transmissão abruptamente através da execução do algoritmo de início lento.

Os algoritmos de retransmissão rápida e recuperação rápida combinados correspondem aos seguintes passos:

- ao receber três reconhecimentos duplicados, $ssthresh$ recebe $max(cwnd/2, 2)$;
- o segmento supostamente perdido é retransmitido;
- $cwnd$ recebe $ssthresh + 3.MSS$. Este acréscimo infla a janela de um valor correspondente aos pacotes que deixaram a rede e originaram os três reconhecimentos duplicados;
- a cada novo reconhecimento duplicado, incrementa-se $cwnd$ de um segmento (MSS). Se e a janela de congestionamento permitir, um novo segmento é transmitido;
- quando um reconhecimento novo é recebido, $cwnd$ recebe $ssthresh$ e o algoritmo de prevenção de congestionamento é executado.

O algoritmo de retransmissão rápida corresponde ao segundo passo, enquanto que o de recuperação rápida corresponde ao terceiro e quarto passos.

O TCP Reno pode conseguir o aumento da vazão em situações de congestionamento moderado [122] e apareceu pela primeira vez no UNIX BSD versão 4.3 Reno em 1990.

A.3.3 TCP New Reno

O TCP New Reno é uma otimização do TCP Reno para casos de múltiplas perdas de pacotes em uma única janela de congestionamento. O TCP New Reno inclui uma modificação no algoritmo de recuperação rápida para reduzir as chances do TCP Reno ter que esperar por um estouro do temporizador nestas ocasiões [122].

Quando o TCP percebe que ocorreu a perda de um pacote através do recebimento de reconhecimentos duplicados, um reconhecimento de número diferente só será enviado quando o pacote retransmitido chegar ao destino. No caso de uma única perda, esse reconhecimento confirma o recebimento de todos os segmentos transmitidos até antes da execução do algoritmo de retransmissão rápida. Entretanto, se

ocorrem vários descartes, este reconhecimento só confirma alguns segmentos; mais precisamente, todos até a próxima perda. Por este motivo, este reconhecimento é denominado parcial [91]. Conclui-se então que reconhecimentos parciais duplicados são um indício de que um outro pacote provavelmente foi perdido dentro da mesma janela de transmissão.

Para o TCP Reno, o valor da janela de transmissão é reduzido à *ssthresh* na chegada de reconhecimentos parciais duplicados. Além disso, a execução do algoritmo de recuperação rápida é interrompida, dando início à fase de prevenção de congestionamento. O processo se repete para cada novo conjunto de reconhecimentos parciais duplicados, fazendo com que o TCP reduza a janela de transmissão pela metade seguidas vezes. Este comportamento pode levar ao *timeout*, a depender do número de perdas e do tamanho da janela de transmissão no início do algoritmo de recuperação rápida [122].

Já o TCP New Reno, ao receber reconhecimentos parciais, se mantém no algoritmo de retransmissão rápida evitando as múltiplas reduções no valor da janela de congestionamento. Cada reconhecimento parcial é tratado como uma indicação de que mais um pacote foi perdido e deve ser retransmitido. Deste modo, quando vários pacotes são perdidos em uma mesma janela de dados, o TCP New Reno é capaz de evitar o *timeout*. Para isto, ele retransmite um pacote perdido por RTT até que todos os pacotes perdidos desta janela tenham sido retransmitidos. Para sair do algoritmo de recuperação rápida, o TCP New Reno espera pelo recebimento de um reconhecimento que confirme todos os pacotes pendentes quando este algoritmo foi iniciado.

Vale notar que o TCP Reno tenta ser mais agressivo do que o TCP Tahoe. Da mesma forma, o TCP New Reno tenta ser mais agressivo do que o TCP Reno.

A.3.4 TCP SACK

As implementações mais tradicionais do TCP utilizam reconhecimentos acumulativos. Isto é, um determinado reconhecimento indica que todos os bytes de numeração inferior a dele já foram recebidos com sucesso. Isto impõe uma limitação no desempenho em situações onde há mais de uma perda em uma mesma janela de transmissão. O motivo é que os reconhecimentos são enviados referenciando apenas a primeira lacuna do *buffer* de dados no nó destino, não dando informação alguma sobre as demais perdas.

O TCP Tahoe entra em início lento e recomeça a transmitir os dados a partir

do primeiro segmento perdido. Consequentemente, os segmentos recebidos com sucesso no destino serão retransmitidos desnecessariamente. Já para TCP Reno e o TCP New Reno, a cada conjunto de reconhecimentos duplicados, o nó fonte reenvia apenas o pacote referenciado. Portanto, o TCP é obrigado a aguardar um RTT para a chegada do reconhecimento do pacote retransmitido, o qual indicará o próximo pacote perdido. Consequentemente, apenas um pacote perdido pode ser notado a cada RTT. A única diferença entre os dois é que o TCP New Reno se mantém na fase de recuperação rápida, sem precisar reduzir a janela de transmissão múltiplas vezes.

Uma solução para estes problemas seria o uso de reconhecimento seletivo (SACK) por parte do TCP. Com esta modificação, o nó fonte obtém mais informações sobre quais segmentos foram recebidos e quais não foram. Com isso, ele pode efetuar somente as retransmissões necessárias, aumentando a eficiência da conexão.

Entretanto, esta sofisticação requer modificações no funcionamento do protocolo TCP, tanto para o nó fonte quanto para o nó destino. O campo de opções do cabeçalho do TCP [2] é utilizado para incluir informações de reconhecimento seletivo. Existem dois tipos de opções que são usadas pelo TCP SACK:

- *SACK-Permitted*, utilizada para decidir se a alternativa SACK será ou não adotada e enviada no estabelecimento da conexão;
- SACK propriamente dita, a qual passa as informações sobre os segmentos reconhecidos, e cuja estrutura do seu campo está ilustrada na figura A.1 [92]. Esta opção deve ser incluída em todos os pacotes que não reconhecem o maior número de sequência no *buffer* de recepção. Nesta situação, a rede possui dados perdidos ou fora de ordem, de forma que o nó destino armazena dados não contíguos no seu *buffer* de recepção.

Como se pode ver, o nó destino envia informações sobre blocos de dados que foram recebidos e que se encontram isolados. O campo tamanho informa a quantidade de blocos. A borda esquerda de um bloco é o número de sequência do primeiro byte deste bloco, enquanto que a borda direita é o número de sequência seguinte ao último byte deste bloco. Esta convenção é coerente com a utilizada para os reconhecimentos nas implementações convencionais. Além disso, a utilidade do campo de reconhecimento permanece inalterada.

O campo de opções do TCP pode conter no máximo quarenta bytes. Um campo de SACK que especifique n blocos terá $8.n + 2$ bytes, o que dá um limite teórico

de quatro blocos que podem ser reconhecidos. Como na prática outras opções são usadas, o número de blocos, em geral, fica limitado a três.

Modificações para o Nó Destino

O objetivo da opção de reconhecimento seletivo é que o nó fonte tenha mais informações sobre quais segmentos foram recebidos, de modo a definir uma estratégia de retransmissão mais eficiente. Portanto, existem regras bem definidas para a geração dos reconhecimentos seletivos por parte do nó destino para que o nó fonte tenha o retrato mais recente e fiel do estado do seu *buffer*.

Ao gerar um reconhecimento seletivo, o nó fonte deve obedecer às seguintes regras:

1. o primeiro bloco deve conter o segmento que disparou este reconhecimento (a menos que o último segmento tenha dado origem a um reconhecimento normal), ou seja, o primeiro bloco mostra a mudança mais recente no estado do *buffer* do nó destino;

2. o nó destino deve incluir o maior número possível de blocos. Vale notar que o espaço disponível para o envio de blocos de reconhecimento seletivo pode não ser suficiente para reportar todos os blocos presentes no *buffer* de recepção;

3. os blocos de reconhecimento seletivo devem ser preenchidos do mais recente (primeiro bloco) para o mais antigo. Deste modo, um bloco de dados será incluído em no mínimo três reconhecimentos, o que agrega redundância em caso de perda de reconhecimentos.

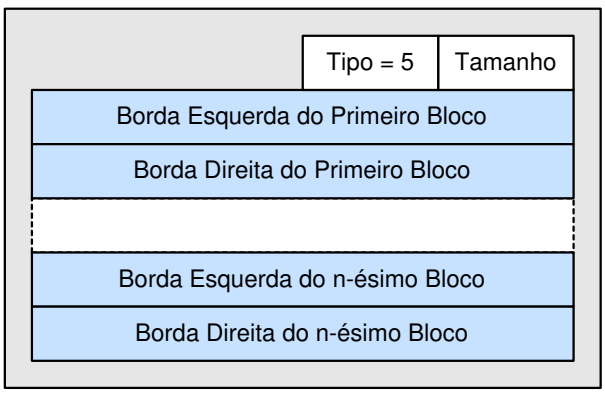


Figura A.1: Campo de reconhecimento seletivo ou de SACK.

Com relação à última regra, mesmo que sejam perdidos todos os reconhecimentos que reportam um bloco de reconhecimento seletivo, a pior consequência possível é a retransmissão desnecessária dos segmentos deste bloco. Como sem o reconhecimento seletivo estas transmissões desnecessárias seriam feitas de qualquer modo, o comportamento do TCP SACK neste pior caso é igual ao dos TCPs convencionais.

O nó destino mantém um *buffer* com os segmentos recebidos à espera de que sejam lidos pela aplicação. Quando isto acontece, os dados lidos são apagados do *buffer*. Em caso de falta de espaço de *buffer*, o nó destino pode descartar segmentos que tenham sido reconhecidos através da opção de SACK. Neste caso, o próximo reconhecimento gerado deve ter no primeiro bloco a mudança mais recente nos segmentos reconhecidos, conforme as regras já citadas.

Modificações para o Nó Fonte

No TCP tradicional, o nó fonte mantém um *buffer* de retransmissão com os segmentos que foram enviados e ainda não foram reconhecidos. Ao receber um reconhecimento, ele descarta todos os segmentos cujos números de sequência sejam menores que o número indicado.

O TCP SACK estabelece que para cada segmento em seu *buffer*, o nó fonte tenha um indicador (*SACKed flag*) que mostre que o segmento foi reportado em um bloco de reconhecimento seletivo. Em outras palavras, quando o nó fonte recebe um reconhecimento com SACK, ele deve ativar este indicador ($SACKed\ flag = 1$) para todos os segmentos que estejam dentro dos blocos especificados na opção SACK. Deste modo, quando houver retransmissões, somente os segmentos que tiverem o indicador desativado ($SACKed\ flag = 0$) são passíveis de retransmissão.

Quando houver um *timeout*, o nó fonte deve desativar todos os indicadores da fila de retransmissão, pois o nó destino pode ter descartados os segmentos anteriormente reconhecidos. Neste caso, o nó fonte deve retransmitir o segmento referenciado no último reconhecimento, independentemente do estado dos indicadores.

Os segmentos que são reconhecidos seletivamente não podem ser apagados do *buffer* de retransmissão, tendo em vista que o nó destino pode descartá-los. Um segmento só é apagado quando for reconhecido de forma convencional.

Apêndice B

Algoritmos de Gerenciamento Ativo de Filas

Este apêndice apresenta a lógica dos algoritmos RED e FRED, apresentados no capítulo 3 e utilizados nos mecanismos de implementação do serviço assegurado.

B.1 RED (*Random Early Detection*)

Em situações de congestionamento, o RED (algoritmo B.1) [88] marca ou descarta pacotes a depender do protocolo de transporte. Para o TCP, a opção é pelo descarte.

O RED também pode ser implementado no modo em bytes ou em pacotes. Os trechos, variáveis e parâmetros adicionados ao algoritmo para o funcionamento no modo em bytes estão indicados ao final de cada linha.

Algoritmo B.1: RED.

Variáveis:

avg: tamanho médio estimado da fila

$count$: número de pacotes consecutivos que não foram eliminados pelo descarte aleatório (na fase de prevenção de congestionamento)

q: tamanho instantâneo da fila

$time$: instante de tempo atual

q_{time}: instante de tempo de início de ociosidade da fila

m: estimativa do volume de informação que poderia ter sido transmitido durante um período de ociosidade da fila

---

Algoritmo B.1: RED (continuação).

Variáveis: (cont.)

p_a: probabilidade final de descarte utilizando o valor de *count*

p_b: probabilidade inicial de descarte

$PacketSize$: tamanho do pacote recebido (modo em bytes)

Funções:

$f(t)$: uma função linear do tempo t

Parâmetros:

w_q: peso do tamanho atual da fila no cálculo do tamanho médio

min_{th}: limiar para entrada na fase de prevenção de congestionamento

max_{th}: limiar para entrada na fase de controle de congestionamento

max_p: valor máximo para p_b

$MaxPacketSize$: tamanho máximo de um pacote (modo em bytes)

Inicialmente:

$avg \leftarrow 0$

$count \leftarrow -1$

À cada chegada de um pacote:

calcular novo avg:

se a fila está ocupada

$avg \leftarrow avg + w_q.(q - avg)$

senão

$m \leftarrow f(time - q_{time})$

$avg \leftarrow (1 - w_q)^m.avg$

se $min_{th} \leq avg < max_{th}$

$count \leftarrow count + 1$

calcular probabilidade p_a:

$p_b \leftarrow max_p(avg - min_{th})/(max_{th} - min_{th})$

$p_b \leftarrow p_b.(PacketSize/MaxPacketSize)$ (modo em bytes)

$p_a \leftarrow p_b/(1 - count * p_b)$

com probabilidade p_a:

marcar ou descartar o pacote

$count \leftarrow 0$

senão se $max_{th} \leq avg$

marcar ou descartar o pacote

Algoritmo B.1: RED (continuação).

À cada chegada de um pacote (cont.):

  $count \leftarrow 0$

senão $count \leftarrow -1$

Quando a fila se esvaziar:

$q_{time} \leftarrow time$

O ajuste dos parâmetros do RED não é trivial. FLOYD e JACOBSON [88] estabeleceram algumas recomendações que se encontram resumidas a seguir:

- o parâmetro w_q não deve ser muito alto pois quase não haveria filtragem. Por outro lado, um valor muito baixo faria avg reagir de forma muito lenta às variações na ocupação da fila. Através de alguns cálculos é recomendado que $w_q \geq 0.001$, sendo $w_q = 0.002$ um ajuste típico;

- a escolha dos parâmetros min_{th} e max_{th} depende do tamanho médio de fila desejado. min_{th} não pode ser muito baixo para não limitar a utilização do enlace enquanto que max_{th} não pode ser muito alto para não causar grandes retardos. Por outro lado, o RED opera de forma mais efetiva quando a diferença entre os dois é maior do que o aumento típico no valor calculado do tamanho médio da fila em um RTT. É recomendado $max_{th} \geq 2.min_{th}$ para evitar que muitos descartes ocorram num intervalo curto de tempo, causando o sincronismo global;

- o parâmetro max_p deve ser tal que a probabilidade de descarte varie lentamente com a mudança no tamanho médio da fila, evitando oscilações. Por este motivo, recomenda-se $max_p \leq 0.1$, sendo $max_p = 0.02$ um ajuste típico.

Na configuração das filas RED com dois (RIO) e três níveis para as simulações deste trabalho, estas recomendações são adotadas para a maior prioridade de descarte (pacotes *out* e vermelhos). Nos demais casos, a maioria delas é seguidas. Apesar disto, não há como considerar estes procedimentos ótimos tendo em vista a grande variedade de cenários e a discussão que ainda cerca o assunto.

B.2 FRED (*Flow Random Early Drop*)

O algoritmo B.2 descreve o FRED [98]. Os trechos, variáveis e parâmetros adicionados ao algoritmo para o funcionamento no modo em bytes estão indicados ao final de cada linha. Os "grampos" assinalados foram utilizados para a análise dos resultados do primeiro estudo do capítulo 5, onde o FRED é utilizado para a distribuição justa de fichas entre os fluxos no marcador FM. Ao final do algoritmo, são descritas as modificações necessárias para implementar a extensão do FRED para casos com grande número de fluxos [121].

Quanto ao ajuste dos parâmetros do FRED, LIN e MORRIS [98] fizeram as seguintes recomendações:

- min_{th} igual a valor mínimo entre 25% do tamanho da fila e o RTT;
- $max_p = 0.02$, $w_q = 0.002$ e $max_{th} = 2.min_{th}$ de acordo com o aconselhado para o RED;
- $min_q = 2$ para filas pequenas e $min_q = 4$ para filas grandes.

Algoritmo B.2: FRED.

Variáveis:

avg: tamanho médio estimado da fila
q_{time}: instante de tempo de início de ociosidade da fila
$count$: número de pacotes consecutivos que não foram eliminados pelo descarte aleatório (na fase de prevenção de congestionamento)
q: tamanho instantâneo da fila
$time$: instante de tempo atual
p_b: probabilidade inicial de descarte
p_a: probabilidade final de descarte utilizando o valor de $count$
m: estimativa do volume de informação que poderia ter sido transmitido durante um período de ociosidade da fila
max_q: valor máximo permitido para o tamanho da fila por fluxo
$avgcq$: valor médio por fluxo do tamanho da fila em bytes ou pacotes
$qlen_i$: tamanho da fila por fluxo
$strike_i$: número de tentativas de exceder max_q por fluxo

Algoritmo B.2: FRED (continuação).

Variáveis: (cont.)

$Nactive$: número de fluxos ativos

$PacketSize$: tamanho do pacote recebido (modo em bytes)

Funções:

$f(t)$: uma função linear do tempo t

$conn(P)$: identificador da conexão do pacote P

$MAX(a, b)$: valor máximo entre a e b

Parâmetros:

w_q: peso do tamanho atual da fila no cálculo do tamanho médio

min_{th}: limiar para entrada na fase de prevenção de congestionamento

max_{th}: limiar para entrada na fase de controle de congestionamento

max_p: valor máximo para p_b

min_q: mínimo garantido para o tamanho da fila por fluxo

$MaxPacketSize$: tamanho máximo de um pacote (modo em bytes)

$MeanPacketSize$: tamanho médio estimado de um pacote (modo em bytes)

À cada chegada de um pacote P:

se fluxo $i = conn(P)$ não tem estado criado

criar estado para o fluxo i

$qlen_i \leftarrow 0$

$strike_i \leftarrow 0$

se a fila está vazia

calcular tamanho médio da fila

$max_q \leftarrow min_{th}$

se $avg \geq max_{th}$ (início do bloco A)

$max_q \leftarrow 2$ (fim do bloco A)

$max_q \leftarrow 2.MeanPacketSize$ (fim do bloco A - modo em bytes)

identifica e gerencia fluxos não adaptativos: * grampo 1,0 *

se $(qlen_i \geq max_q)$ * grampo 1,1 *

ou $(avg \geq max_{th}$ e $qlen_i > 2 * avgcq)$ (linha B) * grampo 1,2 *

ou $(qlen_i \geq avgcq$ e $strike_i > 1)$ * grampo 1,3 *

$strike_i \leftarrow strike_i + 1$

descarta o pacote P

retornar

Algoritmo B.2: FRED (continuação).

À cada chegada de um pacote P: (cont.)
  opera no modo de descarte aleatório (prevenção de congestionamento):
  se $min_{th} \leq avg < max_{th}$
    $count \leftarrow count + 1$
    só descarta de fluxos com mais pacotes na fila (robustos):
    se $qlen_i \geq MAX(min_q, avgcq)$ * grampo 3,0 *
      $p_b \leftarrow max_p(avg - min_{th})/(max_{th} - min_{th})$
      $p_b \leftarrow p_b.(PacketSize/MaxPacketSize)$ (modo em bytes)
      $p_a \leftarrow p_b/(1 - count * p_b)$
      com probabilidade p_a:
        descartar o pacote P
      com probabilidade p_a: (cont.)
        $count \leftarrow 0$
        retornar
  opera no modo normal:
  senão se $avg < min_{th}$
    $count \leftarrow -1$
  opera no modo de controle de congestionamento:
  senão ($avg \geq max_{th}$) * grampo 2,0 *
    modo *drop-tail*: (início do bloco C)
    descartar o pacote P
    $count \leftarrow 0$
    retornar (fim do bloco C)
  se $qlen_i = 0$
    $Nactive \leftarrow Nactive + 1$
  calcular tamanho médio da fila
  aceitar o pacote P
  $qlen_i \leftarrow qlen_i + 1$
  $qlen_i \leftarrow qlen_i + PacketSize$ (modo em bytes)
  $q \leftarrow q + 1$
  $q \leftarrow q + PacketSize$ (modo em bytes)

Algoritmo B.2: FRED (continuação).

À cada transmissão de um pacote:
  calcular tamanho médio da fila
  $qlen_i \leftarrow qlen_i - 1$
  $qlen_i \leftarrow qlen_i - PacketSize$ (modo em bytes)
  $q \leftarrow q - 1$
  $q \leftarrow q - PacketSize$ (modo em bytes)
  se $qlen_i = 0$
    $Nactive \leftarrow Nactive - 1$
    apagar estado para o fluxo i
Calcular tamanho médio da fila:
  se $q > 0$ ou pacote foi transmitido
    $avg \leftarrow (1 - w_q).avg + w_q.q$
  senão
    $m \leftarrow f(time - q_time)$
    $avg \leftarrow (1 - w_q)^m.avg$
    $q_{time} \leftarrow time$
  se $Nactive > 0$
    $avgcq \leftarrow avg/Nactive$
  senão
    $avgcq \leftarrow avg$
  $avgcq \leftarrow MAX(avgcq, 1)$
  $avgcq \leftarrow MAX(avgcq, MeanPacketSize)$ (modo em bytes)
  se $q = 0$ e pacote foi transmitido
    $q_{time} \leftarrow time$

Modificações para a extensão para grande número de fluxos:
Apagar o bloco A e a linha B
Substituir o bloco C pelo trecho abaixo
    modo de dois pacotes (*two-packet mode*):
    se $qlen_i \geq 2$
    se $qlen_i \geq 2.MeanPacketSize$ (modo em bytes)
      descartar o pacote P
      $count \leftarrow 0$
      retornar

Apêndice C

Resultados Adicionais

Tabela C.1: Índice de justiça no compartilhamento da largura de faixa assegurada. Cenário TCP heterogêneos sem CBR/UDP - 10 fluxos TCP.

#	min_q	min_{th}	max_{th}	10%	30%	50%	70%	90%	Total
17	4.0	25.0	100.0	0.9886	0.9943	0.9880	0.9880	0.9918	4.9507
7	2.0	25.0	100.0	0.9890	0.9934	0.9876	0.9883	0.9920	4.9504
4	2.0	10.0	100.0	0.9970	0.9951	0.9881	0.9800	0.9789	4.9391
14	4.0	10.0	100.0	0.9986	0.9955	0.9882	0.9785	0.9781	4.9388
6	2.0	25.0	75.0	0.9935	0.9910	0.9803	0.9876	0.9832	4.9355
16	4.0	25.0	75.0	0.9925	0.9905	0.9798	0.9875	0.9837	4.9341
23	15.0	25.0	100.0	0.9869	0.9876	0.9873	0.9845	0.9849	4.9313
19	4.0	50.0	100.0	0.9835	0.9868	0.9833	0.9857	0.9920	4.9312
13	4.0	10.0	75.0	0.9972	0.9955	0.9877	0.9812	0.9643	4.9259
3	2.0	10.0	75.0	0.9975	0.9954	0.9878	0.9799	0.9645	4.9252
25	15.0	50.0	100.0	0.9847	0.9840	0.9856	0.9848	0.9850	4.9241
9	2.0	50.0	100.0	0.9705	0.9864	0.9838	0.9865	0.9912	4.9183
12	4.0	10.0	50.0	0.9967	0.9932	0.9845	0.9767	0.9416	4.8928
2	2.0	10.0	50.0	0.9946	0.9928	0.9850	0.9759	0.9329	4.8813
10	2.0	75.0	100.0	0.9837	0.9866	0.9789	0.9666	0.9604	4.8762
26	15.0	75.0	100.0	0.9618	0.9828	0.9839	0.9741	0.9736	4.8762
20	4.0	75.0	100.0	0.9808	0.9850	0.9788	0.9676	0.9630	4.8752
28	25.0	50.0	100.0	0.9707	0.9734	0.9679	0.9638	0.9701	4.8459
29	25.0	75.0	100.0	0.9679	0.9696	0.9642	0.9630	0.9655	4.8302
18	4.0	50.0	75.0	0.9564	0.9883	0.9744	0.9654	0.9404	4.8249
8	2.0	50.0	75.0	0.9335	0.9891	0.9746	0.9655	0.9398	4.8025
22	15.0	25.0	75.0	0.9892	0.9915	0.9676	0.9395	0.8814	4.7693
15	4.0	25.0	50.0	0.9165	0.9594	0.9321	0.9815	0.9649	4.7545
5	2.0	25.0	50.0	0.9122	0.9570	0.9343	0.9821	0.9653	4.7508
24	15.0	50.0	75.0	0.8547	0.9910	0.9650	0.9374	0.8897	4.6378
1	2.0	10.0	25.0	0.8799	0.8859	0.8485	0.8863	0.8318	4.3325
27	25.0	50.0	75.0	0.6821	0.9750	0.9484	0.9194	0.8020	4.3269
11	4.0	10.0	25.0	0.8761	0.8702	0.8213	0.8303	0.8731	4.2711
21	15.0	25.0	50.0	0.8420	0.8832	0.7636	0.6714	0.6201	3.7802
30	TBM			0.1091	0.1080	0.1080	0.1040	0.0977	0.5267

Tabela C.2: Índice de justiça no compartilhamento da largura de faixa assegurada. Cenário TCP heterogêneos sem CBR/UDP - 100 fluxos TCP.

#	min_q	min_{th}	max_{th}	10%	30%	50%	70%	90%	Total
2	2.0	100.0	500.0	0.9935	0.9863	0.9781	0.9833	0.9875	4.9287
12	4.0	100.0	500.0	0.9921	0.9832	0.9779	0.9848	0.9889	4.9269
15	4.0	250.0	500.0	0.9906	0.9866	0.9777	0.9813	0.9889	4.9252
1	2.0	100.0	250.0	0.9931	0.9793	0.9850	0.9846	0.9810	4.9229
5	2.0	250.0	500.0	0.9907	0.9864	0.9776	0.9807	0.9872	4.9226
11	4.0	100.0	250.0	0.9919	0.9759	0.9845	0.9841	0.9829	4.9192
3	2.0	100.0	750.0	0.9964	0.9977	0.9493	0.9542	0.9654	4.8631
13	4.0	100.0	750.0	0.9962	0.9977	0.9534	0.9485	0.9654	4.8613
21	150.0	250.0	500.0	0.9909	0.9853	0.9811	0.9652	0.9381	4.8606
6	2.0	250.0	750.0	0.9952	0.9976	0.9489	0.9469	0.9622	4.8507
16	4.0	250.0	750.0	0.9954	0.9983	0.9470	0.9465	0.9629	4.8500
14	4.0	100.0	1000.0	0.9470	0.9762	0.9456	0.9849	0.9935	4.8472
10	2.0	750.0	1000.0	0.9503	0.9748	0.9475	0.9815	0.9903	4.8445
4	2.0	100.0	1000.0	0.9461	0.9726	0.9461	0.9826	0.9888	4.8362
19	4.0	500.0	1000.0	0.9476	0.9734	0.9403	0.9820	0.9906	4.8339
7	2.0	250.0	1000.0	0.9472	0.9728	0.9401	0.9848	0.9879	4.8328
20	4.0	750.0	1000.0	0.9472	0.9721	0.9394	0.9819	0.9906	4.8312
18	4.0	500.0	750.0	0.9965	0.9967	0.9466	0.9317	0.9566	4.8281
17	4.0	250.0	1000.0	0.9439	0.9734	0.9394	0.9836	0.9876	4.8280
9	2.0	500.0	1000.0	0.9453	0.9703	0.9369	0.9820	0.9910	4.8256
8	2.0	500.0	750.0	0.9951	0.9950	0.9368	0.9313	0.9583	4.8165
22	150.0	250.0	750.0	0.9950	0.9974	0.9188	0.9388	0.9487	4.7988
27	250.0	500.0	750.0	0.9935	0.9932	0.9156	0.9273	0.9486	4.7782
30	TBM			0.9298	0.9695	0.9153	0.9779	0.9796	4.7720
25	150.0	500.0	1000.0	0.9349	0.9684	0.9104	0.9764	0.9805	4.7706
28	250.0	500.0	1000.0	0.9338	0.9685	0.9108	0.9733	0.9812	4.7676
29	250.0	750.0	1000.0	0.9259	0.9710	0.9177	0.9763	0.9757	4.7665
23	150.0	250.0	1000.0	0.9249	0.9673	0.9121	0.9784	0.9802	4.7630
26	150.0	750.0	1000.0	0.9313	0.9655	0.9137	0.9734	0.9777	4.7617
24	150.0	500.0	750.0	0.9940	0.9918	0.9106	0.9213	0.9420	4.7596

Tabela C.3: Índice de justiça no compartilhamento da largura de faixa assegurada. Cenário TCP homogêneos com CBR/UDP - 100 fluxos TCP.

#	min_q	min_{th}	max_{th}	10%	30%	50%	70%	90%	Total
3	2.0	100.0	750.0	0.9688	0.9856	0.9749	0.9855	0.9959	4.9106
13	4.0	100.0	750.0	0.9622	0.9824	0.9713	0.9847	0.9962	4.8969
12	4.0	100.0	500.0	0.9465	0.9790	0.9819	0.9902	0.9964	4.8939
6	2.0	250.0	750.0	0.9542	0.9847	0.9740	0.9830	0.9951	4.8910
16	4.0	250.0	750.0	0.9500	0.9851	0.9732	0.9833	0.9951	4.8865
2	2.0	100.0	500.0	0.9314	0.9826	0.9836	0.9884	0.9964	4.8825
18	4.0	500.0	750.0	0.9449	0.9798	0.9686	0.9771	0.9968	4.8671
8	2.0	500.0	750.0	0.9445	0.9836	0.9631	0.9768	0.9970	4.8650
5	2.0	250.0	500.0	0.9129	0.9802	0.9799	0.9877	0.9932	4.8539
15	4.0	250.0	500.0	0.9012	0.9808	0.9749	0.9906	0.9934	4.8408
7	2.0	250.0	1000.0	0.8723	0.9440	0.9725	0.9934	0.9969	4.7791
9	2.0	500.0	1000.0	0.8690	0.9421	0.9755	0.9937	0.9975	4.7778
20	4.0	750.0	1000.0	0.8644	0.9510	0.9702	0.9924	0.9958	4.7737
17	4.0	250.0	1000.0	0.8642	0.9459	0.9721	0.9938	0.9971	4.7732
14	4.0	100.0	1000.0	0.8644	0.9409	0.9720	0.9943	0.9962	4.7678
19	4.0	500.0	1000.0	0.8616	0.9467	0.9704	0.9918	0.9972	4.7677
10	2.0	750.0	1000.0	0.8659	0.9383	0.9695	0.9902	0.9957	4.7597
4	2.0	100.0	1000.0	0.8503	0.9365	0.9720	0.9949	0.9960	4.7497
22	150.0	250.0	750.0	0.9342	0.9562	0.9667	0.9288	0.9262	4.7122
27	250.0	500.0	750.0	0.9125	0.9609	0.9651	0.9250	0.9226	4.6861
24	150.0	500.0	750.0	0.9181	0.9391	0.9650	0.9254	0.9299	4.6775
11	4.0	100.0	250.0	0.8871	0.8815	0.9457	0.9654	0.9859	4.6656
29	250.0	750.0	1000.0	0.8545	0.9193	0.9339	0.9717	0.9807	4.6602
21	150.0	250.0	500.0	0.8838	0.9641	0.9626	0.9336	0.9120	4.6561
28	250.0	500.0	1000.0	0.8408	0.9139	0.9391	0.9648	0.9783	4.6368
26	150.0	750.0	1000.0	0.8381	0.9095	0.9350	0.9657	0.9791	4.6275
23	150.0	250.0	1000.0	0.8453	0.9088	0.9277	0.9649	0.9800	4.6267
25	150.0	500.0	1000.0	0.8321	0.9077	0.9378	0.9669	0.9781	4.6226
1	2.0	100.0	250.0	0.8811	0.8190	0.9389	0.9845	0.9903	4.6138
30	TBM			0.0128	0.0126	0.0127	0.0119	0.0108	0.0608

Apêndice D

Glossário

ACK :	Reconhecimento (*ACKnowledgement*).
AF :	Encaminhamento Assegurado (*Assured Forwarding*).
ANOVA :	*Analysis of Variance.*
ATM :	Modo de Transferência Assíncrono (*Asynchronous Transfer Mode*).
BA :	Agregado de Comportamento (*Behavior Aggregate*).
BB :	Corretores de Largura de Faixa (*Bandwidth Brokers*).
BRED :	*Balanced RED.*
CBQ :	*Class-Based Queuing.*
CBR :	Roteamento Baseado em Restrições (*Constraint-Based Routing*).
CBS :	*Committed Burst Size.*
CIR :	*Committed Information Rate.*
COPS :	*Common Open Policy Service.*
CoS :	Classe de Serviço (*Class of Service*).
CRC :	*Cyclic Redundancy Check.*
CTR :	*Committed Target Rate.*
DiffServ :	Serviços Diferenciados (*Differentiated Services*).
DS :	Serviços Diferenciados (*Differentiated Services*).
DSCP :	*Differentiated Services CodePoint.*
DT :	*Dynamic Threshold.*
ECMP :	*Equal-Cost MultiPath.*
EBS :	*Excess Burst Size.*
EF :	Encaminhamento Expresso (*Expedited Forwarding*).

EIR : *Excess Information Rate.*

FEC : Classes de Equivalência de Encaminhamento (*Forwarding Equivalence Classes*).

FM : Marcador Justo (*Fair Marker*).

FRED : *Flow Random Early Drop.*

FTP : Protocolo de Transferência de Arquivo (*File Transfer Protocol*).

HTTP : Protocolo de Transferência de HiperTexto (*HyperText Transfer Protocol*).

IEEE : Instituto dos Engenheiros Elétricos e Eletrônicos (*Institute of Electrical and Electronic Engineers*).

IETF : *Internet Engineering Task Force.*

IGP : *Interior Gateway Protocol.*

IRTF : *Internet Research Task Force.*

IP : Protocolo Internet (*Internet Protocol*).

IPv4 : Protocolo Internet versão 4 (*Internet Protocol version 4*).

IPv6 : Protocolo Internet versão 6 (*Internet Protocol version 6*).

IntServ : Serviços Integrados (*Integrated Services*).

IS-IS : *Intermediate System-to-Intermediate System.*

ISP : Provedor de Acesso à Internet (*Internet Service Provider*).

LDP : Protocolo de Distribuição de Rótulos (*Label Distribution Protocol*).

LSP : Caminho Comutado por Rótulo (*Label-Switched Path*).

LSR : Roteador de Comutação por Rótulo (*Label-Switching Router*).

MA : Marcação por Agregado.

MAF : Marcação por Agregado atenta a Fluxos.

MAMT : Múltiplas Médias Múltiplos Limiares (*Multiple Average Multiple Threshold*).

MAST : Múltiplas Médias Único Limiar (*Multiple Average Single Threshold*).

ME : Melhor Esforço.

MF : Marcação por Fluxo.

MF : *Multi-Field.*

MPLS : *MultiProtocol Label Switching.*

MSS : *Maximum Segment Size.*

NFS : *Network File System.*

NS-2 : *Network Simulator version 2.*

OSPF : *Open Shortest Path First.*

PBS : *Peak Burst Size.*

PDU : *Protocol Data Unit.*

PHB : Comportamento Por Enlace (*Per-Hop Behavior*).

PIR : *Peak Information Rate.*

PMP : *Paris Metro Pricing.*

PTR : *Peak Target Rate.*

QDP : Parâmetro de Diferenciação de Qualidade (*Quality Differentiation Parameters* - QDPs).

QoS : Qualidade de Serviço (*Quality of Service*).

QPS : *QBone Premium Service.*

RED : *Random Early Detection.*

RIO : *RED with In and Out.*

RIP : Protocolo de Informação de Roteamento (*Routing Information Protocol*).

RSVP : Protocolo de Reserva de Recursos (*Resource ReserVation Protocol*).

RTT : Tempo de Ida e Volta *Round-Trip Time.*

RTO : *Retransmission TimeOut.*

SACK : *Selective ACKnowledgement.*

SAST : Única Média Único Limiar (*Single Average Single Threshold*).

SAMT : Única Média Múltiplos Limiares (*Single Average Multiple Thresholds*).

SCORE : *Scalable Core.*

SFQ : *Stochastic Fair Queueing.*

SLA : Contrato de Nível de Serviço (*Service Level Agreement*).

SLS : Especificação de Nível de Serviço (*Service Level Specification*).

SRED : *Stabilized RED.*

SRRAS : *Single Rate Rate Adaptive Shaper.*

SRTCM : Marcador de Três Cores de Taxa Única (*Single Rate Three Color Marker*).

TCP : Protocolo de Controle de Transmissão (*Transmission Control Protocol*).

TBFT : *Token Bucket Fill Time*.

TBM : Marcador Balde de Fichas (*Token Bucket Marker*).

TCA : Contrato de Condicionamento de Tráfego (*Traffic Conditioning Agreement*).

TCFM : Marcador Justo de Três Cores (*Three Color Fair Marker*).

TCS : Especificação de Condicionamento de Tráfego (*Traffic Conditioning Specification*).

TE : Engenharia de Tráfego (*Traffic Engineering*).

TOS : Tipo de Serviço (*Type Of Service*).

TRRAS : *Two Rate Rate Adaptive Shaper*.

TRTCM : Marcador de Três Cores de Taxa Dupla (*Two Rate Three Color Marker*).

TSW : Marcador de Janela Deslizante no Tempo (*Time Sliding Window*).

TSWTCM : Marcador de Janela Deslizante no Tempo de Três Cores (*Time Sliding Window Three Color Marker*).

TTL : *Time To Live*.

UCAID : *University Corporation for Advanced Internet Development*.

UDP : Protocolo de Datagrama do Usuário (*User Datagram Protocol*).

USD : *User-Share Differentiation*.

vBNS : *very high performance Backbone Network Service*.

VCI : Identificador de Circuito Virtual (*Virtual Circuit Identifier*).

VLAN : Rede Local Virtual (*Virtual Local Area Network*).

VPI : Identificador de Caminho Virtual (*Virtual Path Identifier*).

WDM : Multiplexação por Divisão de Comprimento de Onda (*Wavelength Division Multiplexing*).

WRR : Rodízio Ponderado *Weighted Round-Robin*.

WWW : *World Wide Web*.

Apêndice E

Endereços Importantes na WWW

Instituições envolvidas com este trabalho:

- CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior)
  http://www.capes.gov.br/
- COPPE (Coordenação dos Programas de Pós-Graduação em Engenharia)
  http://www.coppe.ufrj.br/
- UFRJ (Universidade Federal do Rio de Janeiro)
  http://www.ufrj.br/
- RAVEL (laboratório de Redes de Alta VELocidade)
  http://www.ravel.ufrj.br/
- GTA (Grupo de Teleinformática e Automação)
  http://www.gta.ufrj.br/

Download desta tese e dos artigos publicados:

- Pelo RAVEL
  http://www.ravel.ufrj.br/sobre/publicacoes.htm
- Pelo GTA
  http://www.gta.ufrj.br/publicacoes/
  http://www.gta.ufrj.br/~rezende/publicacoes.html

Internet:

- IAB (*Internet Architecture Board*)

  http://www.iab.org/iab/

- IETF (*Internet Engineering Task Force*)

  http://www.ietf.org/

- IRTF (*Internet Research Task Force*)

  http://www.irtf.org/

- *Internet Society*

  http://www.isoc.org/

- Padrões da Internet

  http://www.faqs.org/rfcs/std/std-index.html

- RFCs da Internet

  http://www.ietf.org/rfc.html

  http://www.landfield.com/rfcs/rfc-activeV.html

- *Drafts* da Internet

  ftp://www.ietf.org/internet-drafts/

  http://www.alternic.org/drafts/

QoS na Internet:

- UCAID (*University Corporation for Advanced Internet Development*)

  http://www.ucaid.edu/

- Internet 2

  http://www.internet2.edu/

- QBone da Internet 2

  http://www.internet2.edu/qbone/

Engenharia de tráfego:

- Grupo de trabalho do IETF - TEWG (*Traffic Engineering Work Group*)

  http://www.ietf.org/html.charters/tewg-charter.html

- TEQUILA (*Traffic Engineering for QUality of service in the Internet, at LArge scale*)

  http://www.ist-tequila.org/

MPLS (*MultiProtocol Label Switching*):

- Grupo de trabalho do IETF

  http://www.ietf.org/html.charters/mpls-charter.html

Serviços integrados:

- Grupo de trabalho do IETF

  http://www.ietf.org/html.charters/intserv-charter.html

- RSVP (Resource ReServation Protocol)

  http://www.isi.edu/div7/rsvp/rsvp.html

Serviços diferenciados:

- Grupo de trabalho do IETF

  http://www.ietf.org/html.charters/diffserv-charter.html

- Serviços diferenciados na Internet2

  http://www.internet2.edu/qos/may98Workshop/html/diffserv.html

- Artigos sobre serviços diferenciados - GTA

  http://www.gta.ufrj.br/diffserv/

- Serviços diferenciados - Universidade de Tecnologia de Helsinki, Finlândia

  http://www.hut.fi/~msisomak/diffserv.html

- Serviços diferenciados - CSIRO (*Commonwealth Scientific Industrial and Research Organisation*), Austrália

  http://www.atnf.csiro.au/~msisomak/dif_serv.html

Implementações de Serviços Diferenciados em sistemas operacionais:

- Linux - Instituto Federal de Tecnologia da Suíça (EPFL)
  http://lrcwww.epfl.ch/linux-diffserv/
- FreeBSD - Universidade da Califórnia em Los Angeles (UCLA)
  http://irl.cs.ucla.edu/twotier/

Gerenciamento Ativo de Filas:

- RED (*Random Early Detection*)
  http://www.aciri.org/floyd/red.html

Ferramentas utilizadas:

- NS (*Network Simulator*)
  http://www.isi.edu/nsnam/ns/