Universidade Federal de Goiás
Escola de Engenharia Elétrica
Programa de Pós-Graduação em
Engenharia Elétrica e de Computação

AVALIAÇÃO DE UM AMBIENTE DE DIFERENCIAÇÃO PROPORCIONAL DE SERVIÇO COM TRÁFEGO TCP

por
Aline Carneiro Viana

Goiânia, abril de 2001

UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA
MESTRADO EM ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

AVALIAÇÃO DE UM AMBIENTE DE DIFERENCIAÇÃO PROPORCIONAL DE SERVIÇO COM TRÁFEGO TCP

por
Aline Carneiro Viana

Dissertação submetida à avaliação como requisito parcial para a obtenção do grau de Mestre em Ciências: Engenharia Elétrica e de Computação

Área de Concentração: Engenharia de Computação
Linha de Pesquisa: Telecomunicações, Redes de Computadores

Prof. José Ferreira de Rezende - COPPE/UFRJ
Prof. Otto Carlos Muniz Bandeira Duarte - COPPE/UFRJ
Prof. Leonardo Guerra de Rezende Guedes - UFG
Orientadores

Goiânia, abril de 2001.

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Viana, Aline Carneiro

AVALIAÇÃO DE UM AMBIENTE DE DIFERENCIAÇÃO PROPORCIONAL DE SERVIÇO COM TRÁFEGO TCP

Aline Carneiro Viana. - Goiânia: CPGEEC da UFG, 2001. 95 p.: il.

Dissertação (mestrado) - Universidade Federal de Goiás. Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Goiânia, GO, 2001.

Orientadores: José Ferreira de Rezende
Otto Carlos Muniz Bandeira Duarte
Leonardo Guerra de Rezende Guedes

UNIVERSIDADE FEDERAL DE GOIÁS

Reitora: Profa. Dra. Milca Severino Pereira

Pró-Reitor de Pesquisa e Pós-Graduação: Prof. Dr. José Luis Domingues

Coordenador de Pós-Graduação e Pesquisa EEE/UFG: Prof. Dr. Antônio César Baleeiro Alves

A minha família

Archimedes, Sonia, Ottilia, Alan e Junior

Agradecimentos

Aos meus pais, Archimedes e Sonia, e demais familiares pelo apoio e incentivo ao longo de toda a minha vida.

Ao professor Leonardo Guerra de Rezende Guedes pela orientação oportuna e sincera. Reservo um agradecimento especial aos professores José Ferreira de Rezende e Otto Carlos Muniz Bandeira Duarte, os quais não foram apenas orientadores acadêmicos, mas também verdadeiros mentores. Hoje possuo uma sincera admiração e um grande respeito por estes profissionais. Ao professor Humberto Abdalla Júnior por ter aceito participar da banca examinadora.

Aos amigos Anibal, Artur, Cristiane, Díbio, Dulce, Kleber, Larissa, Leizza, Leonardo Rezende, Luis Marcio, Mauros, Paulo, Rodrigo, Stone, Vivian e Willian, pela sincera amizade e pelos bons momentos de convívio. Em especial, agradeço ao Artur, à Cristiane, ao Kleber, ao Saulo, à Vivian e ao Willian pelo apoio que me deram na redação final desta tese. Agradeço aos amigos do Mestrado em Engenharia Elétrica e de Computação da EEE/UFG, do ano de 1998 e 1999 e aos amigos do Grupo de Teleinformática e Automação (GTA) da COPPE/UFRJ, meus companheiros de caminhada no mundo da pesquisa.

Aos professores do Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFG e do Programa de Engenharia Elétrica da COPPE/UFRJ. Agradeço também à Dulce e demais funcionários da EEE/UFG e à Solange COPPE/PEE pela presteza no atendimento.

À UFG pelo apoio financeiro que possibilitou a realização deste trabalho.

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

AVALIAÇÃO DE UM AMBIENTE DE DIFERENCIAÇÃO PROPORCIONAL DE SERVIÇO COM TRÁFEGO TCP

Aline Carneiro Viana

Abril de 2001

Orientadores:	José Ferreira de Rezende
	Otto Carlos Muniz Bandeira Duarte
	Leonardo Guerra de Rezende Guedes
Programa:	Engenharia Elétrica e de Computação

Esta tese avalia o desempenho do modelo de diferenciação proporcional com fontes de tráfego TCP. Pesquisas relacionadas ao protocolo TCP mostram que seu mecanismo de controle de congestionamento é fortemente influenciado por atrasos nas filas e perdas de pacotes. Esta característica do protocolo TCP torna difícil prever o seu comportamento diante de um modelo de diferenciação relativa de serviço, onde são consideradas restrições proporcionais de atraso e taxa de perda entre classes de serviço. A primeira seqüência de simulações realizadas, valida a implementação de um ambiente de encaminhamento de pacotes que segue o modelo de diferenciação proporcional de serviço, onde são utilizadas tanto fontes com taxa de transmissão constante (*Constant Bit Rate* - CBR) quanto fontes do tipo atividade-silêncio e que seguem uma distribuição Pareto. A segunda seqüência de simulações avalia o desempenho desse ambiente quando são empregadas fontes de tráfego TCP, e diferentes restrições de diferenciação proporcional de atraso e de taxa de perda.

Os resultados das simulações validam o comportamento do esquema de encaminhamento proporcional implementado e comprovam o funcionamento apropriado do modelo proporcional para tráfego TCP, no que se refere às perdas e atrasos para conexões de longa duração.

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

EVALUATION OF A SERVICE PROPORTIONAL DIFFERENTIATION ENVIRONMENT WITH TCP TRAFFIC

Aline Carneiro Viana

April of 2001

Advisors:	José Ferreira de Rezende
	Otto Carlos Muniz Bandeira Duarte
	Leonardo Guerra de Rezende Guedes
Department:	Electrical and Computing Engineering

This dissertation evaluates the performance of the proportional differentiation model with TCP sources. Research related to TCP protocol shows that its congestion control mechanism is strongly influenced by queueing delays and packet drops. Such a feature of the TCP protocol brings difficulties to foresee its behavior in a model of service relative differentiation, where delay and loss rate proportional restrictions are considered between classes of service. The first sequence of performed simulations validates the implementation of an packet forwarding environment that follows the model of service proportional differentiation, while using Constant Bit Rate (CBR) sources and on-off sources that follow a Pareto distribution. The second sequence of simulations analyzes the performance of such environment when TCP sources under different restrictions for delay and loss rate proportional differentiation are used.

The results of the simulations validate the behavior of the proportional forwarding scheme implemented. Results also confirm the appropriate functioning of the proportional model for TCP traffic, while taking into account both delay and loss rate proportional restrictions in long term connections.

Palavras-chave

Qualidade de Serviço

Diferenciação Relativa de Serviços

Tráfego TCP

Internet

Glossário

ACK : Reconhecimento positivo (*positive acknowledgement*).

AF : Encaminhamento Assegurado (*Assured Forwarding*).

BPR : *Backlog-Proportional Rate*.

DDP : *Delay Differentiation Parameters*.

DS : Serviços Diferenciados (*Differentiated Services*).

EF : Encaminhamento Expresso (*Expedited Forwarding*).

IETF : *Internet Engineering Task Force*.

IIS : *Internet Integrated Services*.

IP : *Internet Protocol*.

LDP : *Loss rate Differentiation Parameters*.

MDP : *Mean Delay Proportional*.

MPLS : *MultiProtocol Label Switching*.

MSS : *Maximum Segment Size*.

NS : *Network Simulator*.

PHB : Comportamento por nó (*Per-Hop Behavior*).

PLR : *Proportional Loss Rate*.

PMP : *Paris Metro Pricing*.

PQCM : *Proportional Queue Control Mechanism*.

QoS : Qualidade de Serviço (*Quality of Service*).

RTT : Tempo de ida-e-volta (*Round-Trip Time*).

SDP : *Scheduler Differentiation Parameters*.

SLA : Acordo de Nível de Serviço (*Service Level Agreement*).

TCA : Acordo de Condicionamento de Tráfego (*Traffic Conditioning Agreement*).

TCP :	*Transmission Control Protocol.*
UDP :	*User Datagram Protocol.*
WFQ :	*Weighted-Fair-Queueing.*
WTP :	*Waiting-Time Priority.*

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

ATUALMENTE, a Internet é a segunda maior rede de informação do mundo em número de equipamentos conectados, perdendo somente para a rede telefônica. É a maior rede do mundo em volume de informação trafegada. Tais características motivam o surgimento de propostas para melhorar seu desempenho e agregar novos valores. Por outro lado, a Internet, após mais de 30 anos de existência, manteve praticamente as mesmas características do projeto inicial. Isso criou uma significativa resistência às propostas que modifiquem alguma dessas características, como a simplicidade da camada de rede. No entanto, pesquisas [1, 2] tem mostrado que algumas limitações da Internet atual justificam a realização de alterações em sua arquitetura.

O padrão de tráfego da Internet vem sofrendo várias mudanças nos últimos anos, conseqüência da diversidade de serviços que usuários e empresas vem exigindo da infra-estrutura de rede. As atuais aplicações, por trafegarem uma variedade de mídias tais como voz, áudio, vídeo e dados, vêm exigindo diversos requisitos de Qualidade de Serviço (*Quality of Service* - QoS) [3, 4, 5]. Para atender esta demanda é necessário trocar o atual paradigma de mesmo-serviço-para-todos (*best-effort*), onde os pacotes transmitidos são tratados sem distinção ao longo de suas rotas pelo interior da rede, por um modelo no qual os usuários, as aplicações ou os pacotes individuais, são tratados diferenciadamente, baseado em seus requisitos de QoS. Dentro desta perspectiva, uma das principais tendências é a implantação de uma infra-estrutura de rede que suporte a integração e a diferenciação de serviços, provendo desta forma a QoS exigida pelos usuários/aplicações.

Fornecer suporte à QoS é um problema de complexidade razoável em se tratando de redes corporativas, porém quando se trata da Internet como um todo, a complexidade se torna ainda mais alta. Esse fato deve-se ao crescimento vertiginoso que a Internet obteve nos últimos anos em relação ao número de estações conectadas, quantidade de informações disponíveis e, conseqüentemente, no tráfego transportado. Logo, as novas propostas que ofereçam garantias de QoS devem possuir mecanismos escaláveis a fim de permitir uma adoção de forma abrangente na Internet.

Uma das opções de projeto da Internet foi manter os nós intermediários o mais simples possível, sendo qualquer complexidade adicional colocada nos nós terminais. Baseando-se nesse paradigma, um ponto polêmico é se mecanismos de gerenciamento de tráfego para a provisão de QoS são realmente necessários ou não. Alguns argumentam que sempre que a rede não tiver um desempenho satisfatório, a solução é realizar um super-provisionamento de recursos, aumentando a banda passante disponível [6]. Tal argumentação se baseia no fato de que a utilização de fibras óticas em conjunto com a tecnologia de Multiplexação por Divisão de Comprimentos de Onda (*Wavelength Division Multiplexing* - WDM) tornará a oferta de banda passante tão abundante e barata que a QoS para qualquer aplicação estará automaticamente fornecida [2]. Porém, a proposta de simplesmente super-provisionar a rede possui alguns problemas. Primeiro, um super-provisionamento efetivo é difícil de alcançar, pois teria que acomodar o somatório das taxas de picos de todo o tráfego que passa em um dado enlace e deveria ser atualizado sempre que o tráfego aumentasse. Segundo, uma rede que suporte QoS, não necessita apenas encaminhar informações mais rapidamente, ela necessita também encaminhar diferentes tipos de informação que exigem outras garantias, como baixo retardo e *jitter*. Assim, uma grande banda passante não significa obrigatoriamente garantia de baixo retardo, pois o fator de maior influência neste parâmetro é o tempo de espera em filas que decorre da carga momentânea na rede. Ou seja, uma maior banda passante faz aumentar a dinâmica das filas, porém não garante que o tempo de permanência diminuirá caso muitos fluxos também utilizem esta fila. Caso a questão de provisão de QoS seja enfrentada somente com abundância de banda passante, existe ainda a obrigação de diminuir o retardo de todos os fluxos para o atendimento dos requisitos de apenas um fluxo.

Por outro lado, uma outra proposta é utilizar mecanismos de gerenciamento de tráfego na Internet. Os defensores da adoção desses mecanismos argumentam que independentemente da quantidade de banda passante que uma rede possa ofertar, novas aplicações serão desenvolvidas e demandarão mais banda [6, 2]. Uma maior banda passante disponível torna-se inclusive um incentivo a esse desenvolvimento. Desta forma, o gerenciamento de tráfego consiste em basicamente em políticas e mecanismos que permitem que uma rede satisfaça, de forma eficiente, uma grande quantidade de serviços, ou seja, atenda a tráfegos com diferentes necessidades. É em torno do gerenciamento de tráfego que as pesquisas e a indústria vem concentrando seus esforços para tornar a Internet uma rede de integração e de diferenciação de serviços que garanta a provisão de QoS.

1.1 Qualidade de Serviço na Internet

Vários modelos de serviço e mecanismos foram propostos por grupos de pesquisa do *Internet Engineering Task Force* (IETF) a fim de tratar a questão de provisão de Qualidade de Serviço na Internet. Dentre eles, destacam-se o *Multiprotocol Label Switching* (MPLS) [7], o Roteamento Baseado em Restrições (*Constraint-Based Routing*) [8], a Integração de Serviços [9] e a Diferenciação de Serviços [10]. A seguir serão descritas suas principais características.

1.1.1 MPLS

A função de encaminhamento de um roteador convencional envolve um procedimento de análise do cabeçalho da camada de rede do pacote e a busca em uma tabela de roteamento do próximo nó para onde ele deve ser encaminhado. Essa tabela é construída baseada em informações que protocolos de roteamento distribuem pela rede. Essa função de encaminhamento demanda uma certa capacidade de processamento, e é executada para cada pacote em cada roteador da rede. Assim, à medida que a velocidade dos meios de comunicação cresce, a função de encaminhamento pode tornar-se um gargalo. A tecnologia denominada "MultiProtocol Label Switching"(MPLS) [7] tem uma abordagem diferente, simplificando a função de encaminhamento em roteadores do núcleo da rede.

O MPLS propõe uma forma padronizada de comutação de pacotes baseada na troca de rótulos, podendo ser aplicada a qualquer protocolo da camada de rede. Substitui-se o mecanismo padrão de encaminhamento nó-a-nó baseado no destino por um conceito de troca de rótulo. Isto traz o benefício de simplificar o encaminhamento de pacotes, permitindo ser escalável para taxas de comunicação bastante elevadas. Além disso, o MPLS desacopla as funções de encaminhamento e de roteamento em dois planos distintos.

Como no roteamento tradicional, a camada de rede mantém informações de protocolos de roteamento para determinar como os pacotes devem ser roteados. Baseado nessas informações, os pacotes são classificados em Classes Equivalentes de Encaminhamento (*Forwarding Equivalence Classes* - FECs). Pacotes seguindo o mesmo caminho, pertencentes à mesma FEC, são encaminhados de maneira similar. A cada FEC é atribuída um rótulo pequeno, de comprimento fixo e de significado local a um dado ponto de encaminhamento.

Um conjunto de nós contínuos que implementem a proposta de MPLS e estejam sob a mesma administração é definido como um domínio MPLS. A atribuição de um dado pacote a uma determinada FEC é feita apenas uma vez, quando o pacote entra no domínio MPLS. Ao entrar no domínio MPLS, além de se fazer a análise convencional de encaminhamento da camada de rede em busca do próximo nó, também se faz a busca para descobrir a FEC à qual o pacote pertence e o respectivo rótulo associado. Quando o pacote é enviado ao próximo nó, o rótulo é enviado junto. Em roteadores convencionais suportando MPLS, um pacote recebe um rótulo entre seus cabeçalhos de nível 2 (enlace) e nível 3 (rede). Nos nós subseqüentes, não há mais análise do cabeçalho de camada de rede do pacote. Ao invés disso, o rótulo é usado como um índice para uma tabela a qual especifica o próximo nó e um novo rótulo. O rótulo velho é substituído pelo novo rótulo, e o pacote é encaminhado a seu próximo nó. Este processo de troca de rótulos é similar ao processamento de VCI/VPI em redes ATM [11].

Assim, o plano de encaminhamento realizado pelo MPLS fica baseado fundamentalmente em dois mecanismos:

- na busca em tabelas de rótulos, as quais definem o encaminhamento a ser dado ao datagrama.
- e a subseqüente substituição do rótulo de entrada por um rótulo de saída e o encaminhamento do datagrama pela interface de saída.

O MPLS define protocolos próprios para a distribuição de rótulos associados a FEC's. Uma vez realizada essa distribuição dos rótulos, o mecanismo de encaminhamento desempenhado pelo MPLS é completamente desacoplado do plano de roteamento.

Com base nos rótulos distribuídos, são estabelecidos Caminhos de Comutação de Rótulos (*Label Switched Paths* – LSPs). Um LSP é similar a um Circuito Virtual (*Virtual Circuit* - VC) de ATM e é unidirecional do transmissor para o receptor. Um LSP pode ser visto como um túnel. Quando um pacote chega ao ponto de entrada desse túnel, sua trajetória até o final do túnel já está completamente determinada. O pacote é encapsulado na estrutura proposta pela arquitetura MPLS e somente é desencapsulado ao final do túnel.

O roteador compatível com a proposta de MPLS é chamado de Roteador de Comutação de Rótulos (*Label Switch Router* – LSR). Na inicialização de um domínio MPLS, os LSRs de ingresso, egresso e interiores utilizam algoritmos convencionais de roteamento para determinar as rotas existentes no interior do domínio. É associado um rótulo às rotas entre qualquer par de componentes. A cada conjunto de rótulos que representa uma rota explícita no interior do domínio MPLS é associado um FEC que agrega um conjunto de pacotes com características comuns. Após o estabelecimento dos FECs, quando um pacote chega a um LSR de ingresso na fronteira de um domínio MPLS, este é classificado e recebe o rótulo correspondente ao FEC adequado.

O processo de encaminhamento de pacotes em uma arquitetura MPLS, pode ser ilustrado pela Figura 1.1. Nessa figura, um nó com endereço X desejando enviar pacotes para um nó de endereço Y, colocaria os pacotes no meio compartilhado de sua LAN com o respectivo endereço destino Y. O roteador LSR_1 faria a análise e determinaria em sua tabela de encaminhamento que o destino Y corresponde a um prefixo associado a uma FEC F_y com um rótulo distribuído igual a L_1, tendo como próximo salto o roteador LSR_2. Este pacote é rotulado com L_1 e encaminhado na interface de saída que dá acesso a LSR_2. Em LSR_2 não é mais necessário fazer a análise de prefixo. Basta LSR_2 consultar sua tabela de rótulos através de uma indexação simples e verificar a qual interface o pacote deverá ser encaminhado e com qual rótulo. Supõe-se que para LSR_2 tenha sido distribuído um rótulo L_2 para a mesma FEC F_y. Então LSR_2 substituirá L_1 por L_2 e encaminhará o pacote a LSR_3. O processo se repete nos demais roteadores, até que o pacote atinge o roteador destino, sendo finalmente entregue na LAN onde se encontra o nó com endereço Y. Neste exemplo, LSR_1 e LSR_6 são denominados dispositivos de borda, uma vez que fazem a interface entre domínios não MPLS e o domínio MPLS.

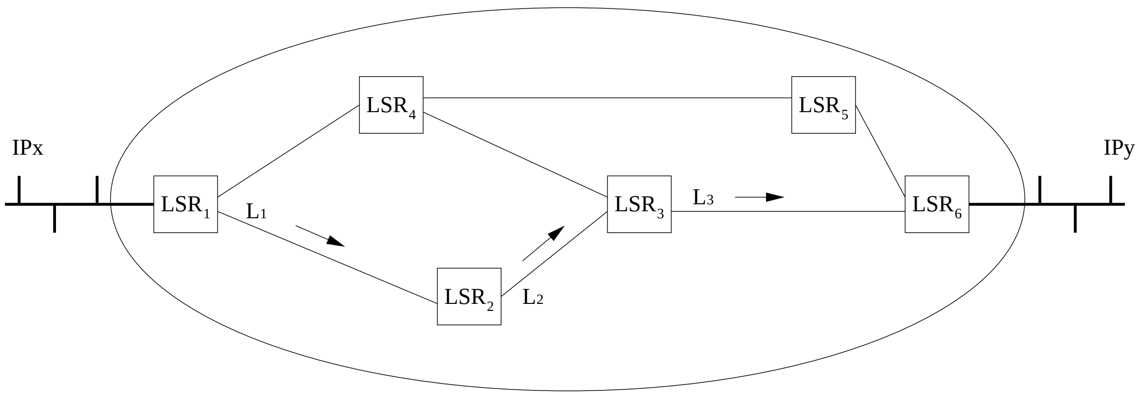


Figura 1.1: Encaminhamento de um pacote em MPLS.

Dentro de um domínio MPLS, o encaminhamento de pacotes, a classificação e a Qualidade de Serviço (QoS) são determinados pelos rótulos atribuídos a cada pacote.

Com esta estrutura, no interior de um domínio que implemente MPLS, os pacotes podem ser comutados ao longo de sua trajetória com base no rótulo associado a cada um deles, dispensando a análise do cabeçalho da camada de rede do pacote. Esta alternativa contrasta com a abordagem convencional onde há uma decisão de roteamento por pacote a cada salto. Desta forma, a classificação e o encaminhamento de pacotes em um domínio MPLS é agilizado. A comutação baseada em rótulos é mais rápida do que o roteamento a cada salto que baseia-se na procura completa em tabelas de roteamento pela entrada que mais se aproxime do endereço de destino do pacote. No entanto, a maior velocidade não é a maior motivação para a adoção de uma arquitetura baseada em comutação de rótulos [12]. O paradigma da comutação baseada em rótulos permite a integração de tecnologias, provendo meios para o transporte de pacotes multiprotocolo de tamanhos variados através de um núcleo de rede que combina roteador e ATM. A comutação baseada em rótulos possibilita o aproveitamento das noções de roteamento escalável de pacotes com tamanho variado baseado na topologia da rede de comutação de pacotes em combinação com a QoS, com a rotulação e engenharia de tráfego do mundo ATM.

1.1.2 Roteamento Baseado em Restrições

O congestionamento em uma rede pode ser causado pela falta de recursos na rede ou por uma distribuição desbalanceada de tráfego. No primeiro caso, todos os roteadores e enlaces estão sobrecarregados e a única solução é prover mais recursos atualizando a infra-estrutura de rede. No segundo caso, algumas partes da rede estão sobrecarregadas, enquanto outras não estão. Esta distribuição desbalanceada de tráfego pode ser causada por um protocolo de roteamento dinâmico, tais como RIP (*Routing Information Protocol*) e OSPF (*Open Shortest Path First*) , pois eles sempre selecionam os caminhos mais curtos para encaminhar os pacotes. Como resultado, roteadores e enlaces ao longo de uma caminho mais curto entre dois nós, podem tornar-se congestionados enquanto roteadores e enlaces de um caminho mais longo estão vazios. A opção de *Equal-Cost Multi-Path* (ECMP) do protocolo OSPF [13] é útil em distribuir carga a vários caminhos mais curtos. Mas se existe apenas um caminho mais curto, o ECMP não ajudará. Em redes simples, pode ser possível a configuração manual do custo dos enlaces por administradores de rede e, assim, distribuir o tráfego de uma forma mais equilibrada. Porém, para redes complexas isto será quase impossível.

A Engenharia de Tráfego é o processo de controlar a maneira pela qual o tráfego flui através das redes a fim de que o congestionamento causado pela utilização ineficiente da rede, possa ser evitado. O Roteamento Baseado em Restrições é o componente mais importante da Engenharia de Tráfego.

Evitar congestionamentos e prover degradação suave de desempenho, no caso de congestionamento, são complementares, o que torna a Engenharia de Tráfego útil em todas as arquiteturas de serviço. O Roteamento Baseado em Restrições é usado para calcular rotas que são sujeitas a múltiplas restrições, representando uma evolução do esquema denominado Roteamento com QoS [14, 15]. Dada uma necessidade de qualidade de serviço para um fluxo ou uma agregação de fluxos, este mecanismo de roteamento encontra a rota mais provável a atender os requisitos de qualidade de serviço especificados. O Roteamento Baseado em Restrições estende o Roteamento com QoS por considerar outras restrições, tal como políticas especificadas por administradores de redes. Tais políticas podem, por exemplo, restringir o acesso a determinadas rotas somente a pacotes oriundos de um conjunto restrito de endereços.

As metas do roteamento baseado em restrições são:

- selecionar rotas que possam atender um dado requisito de QoS.
- evitar congestionamento e melhorar a satisfação do usuário.
- aumentar a utilização da rede.

Enquanto determina uma rota, o Roteamento Baseado em Restrições considera não somente os requisitos topológicos da rede, mas também os requisitos do fluxo, a disponibilidade de recursos dos enlaces e outras políticas especificadas pelos administradores de rede. Por esta razão, é possível que este método retorne um caminho mais longo porém sem congestionamento do que um caminho curto mas congestionado. Assim, o tráfego da rede pode ser distribuído de uma forma mais equilibrada.

As questões centrais do Roteamento Baseado em Restrições são a distribuição das informações de estado de enlace e o cálculo de rotas baseado nessas informações.

Para calcular rotas orientadas à qualidade de serviço, os roteadores necessitam de informações da topologia e de disponibilidade de recursos da rede. Uma possibilidade é estender as mensagens de divulgação de estado de enlace de protocolos tais como o OSPF, de maneira a trazer também a informação de disponibilidade de recursos (a informação da topologia já é nativa nestes protocolos). Deve ser estabelecido um compromisso entre a necessidade de informações precisas e a necessidade de se evitar propagações freqüentes dos anúncios de estado de enlace. Outra possibilidade é distribuir os anúncios de estado de enlace apenas quando houver uma mudança de topologia ou uma mudança significativa na disponibilidade de recursos. A fim de realizar o Roteamento Baseado em Restrições, todos os roteadores no domínio devem ser capazes de anunciar suas informações de disponibilidade de recursos.

Os algoritmos para cálculo de tabelas de roteamento no Roteamento Baseado em Restrições e a complexidade de tais algoritmos dependem das métricas escolhidas para as rotas. Métricas comuns são o custo monetário, quantidade de saltos, largura de banda, confiabilidade, atraso e *jitter*. Algoritmos de roteamento selecionam rotas que otimizam uma ou mais dessas métricas. Ao se tentar minimizar duas ou mais métricas simultaneamente, estas métricas devem ser independentes, a fim de que a otimização seja factível. Ao invés de otimizar múltiplas métricas simultaneamente, uma alternativa mais simples é otimizar múltiplas métricas seqüencialmente em ordem decrescente de importância, ou definir uma função de custo para múltiplas métricas, tais como largura de banda, número de nós, etc.

O Roteamento Baseado em Restrições apresenta como vantagens a possibilidade de uma melhor utilização da rede e de uma melhor satisfação às requisições de QoS. Por outro lado, este mecanismo introduz um aumento na comunicação, na sobrecarga de cálculos e no tamanho da tabela de roteamento, além da desvantagem de que caminhos mais longos podem consumir mais recursos. Outra questão essencial, é a granulosidade do roteamento. Roteamento com baixa granulosidade é mais flexível, sendo mais estável e mais eficiente em termos de utilização de recursos. Porém, a sobrecarga no cálculo e armazenagem das rotas é muito alto.

1.1.3 Integração de Serviços

A proposta desenvolvida pelo grupo de trabalho *Integrated Services* do IETF (*Internet Engineering Task Force*), consiste na definição de um modelo de integração de serviços, chamado modelo IIS (*Internet Integrated Services*) [9, 16], que em seu ponto mais relevante, refere-se à reserva de recursos nos nós da rede para atender os requisitos de QoS de um determinado fluxo. Assim, tal modelo consiste de um conjunto de garantias de serviços, onde, em resposta a uma requisição, a rede se compromete a fornecer algum serviço.

O modelo de Serviços Integrados se propõe a alterar o modelo de serviço original da Internet baseado no melhor-esforço, estendendo a arquitetura original adicionando novos componentes e mecanismos. Assim, um roteador deve implementar e manter uma QoS apropriada para cada fluxo. A estrutura proposta pelo modelo IIS é composta por quatro componentes: o classificador, o escalonador de pacotes, a rotina de controle de admissão e o protocolo de reservas de recursos. Os três primeiros componentes, realizam o controle de tráfego, responsável por suprir qualidades de serviço para cada fluxo no roteador, Figura 1.2. O classificador é responsável por mapear os pacotes que chegam no roteador em classes de serviço. Todos os pacotes da mesma classe obtêm o mesmo tratamento do escalonador de pacotes. O escalonador de pacotes gerencia um conjunto de filas decidindo de acordo com o modelo de serviços adotado, qual fila (classe) deve ter seu pacote encaminhado. Por sua vez, o controle de admissão implementa um algoritmo de decisão usado para determinar se pode ser garantido a QoS requisitada por um novo fluxo sem afetar garantias já estabelecidas. Por fim, o protocolo de reserva de recursos cria e mantém o estado de um fluxo específico nos nós da rede ao longo do caminho do fluxo. A arquitetura de Integração de Serviço foi analisada para tráfego IP por Rezende e Romano em [17].

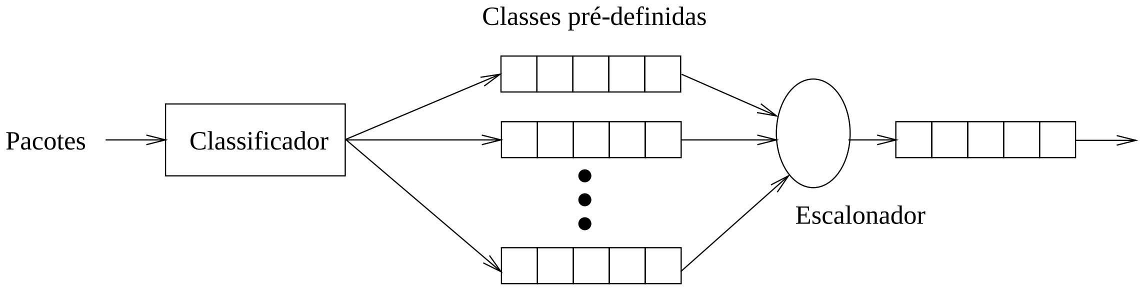


Figura 1.2: Controle de tráfego do modelo de integração de serviços.

O protocolo RSVP (*Resource ReSerVation Protocol*) [18, 19, 20, 21] foi recomendado como o protocolo de reserva de recursos a ser utilizado na arquitetura de Integração de Serviços. Esse protocolo é responsável pela sinalização dos recursos necessários ao atendimento dos requisitos de QoS de uma determinada sessão. Além disso, ele é orientado ao receptor, ou seja, os receptores decidem qual o nível de recursos a serem reservados e são responsáveis por estabelecer e manter ativa esta reserva enquanto desejarem receber dados. As reservas nos roteadores são mantidas por estados voláteis (*soft-states*) que precisam ser renovados periodicamente para permanecerem ativos. A presença de estados voláteis fornece ao RSVP tanto adaptabilidade quanto robustez, pois permite um reestabelecimento automático das reservas no caso de mudanças de rota. O RSVP define diferentes estilos de reserva através de filtros. A reserva de recursos controla a quantidade de recursos reservados, enquanto o filtro associado a esta reserva controla quais pacotes podem usar estes recursos. Por ser orientado ao receptor, o RSVP necessita assegurar que as mensagens de reserva de recursos do receptor sigam exatamente a mesma rota dos fluxos de dados provenientes das fontes. Para tanto, o transmissor envia mensagens de caminho (*path messages*) para os receptores. Essas mensagens armazenam informações sobre o caminho percorrido até os receptores. Por sua vez, os receptores enviam mensagens de reserva (*reservation messages*) com o pedido de reserva pelo caminho inverso até o transmissor. A passagem dessas mensagens renovam os estados voláteis nos roteadores intermediários.

Em adição ao serviço de melhor-esforço, o modelo de Serviços Integrados define dois novos tipos de serviços: o serviço Garantido (*Guaranteed Service*-GS) [22] e o serviço de Carga Controlada(*Controlled-Load Service*-CL) [23]. O serviço Garantido oferece à aplicação contratante, uma garantia determinística do limite máximo de atraso fim-a-fim de cada pacote, de acordo com as características de tráfego contratadas. Assim, este serviço objetiva atender os requisitos de atraso máximo associados aos fluxos de dados de aplicações de tempo-real, como as de áudio e vídeo com *buffer* de apresentação (*playback*), que podem considerar inútil um pacote que chegue após o ponto de apresentação. Cada roteador fornece o Serviço Garantido para um determinado fluxo pela alocação de uma banda passante R e um espaço em *buffer* B que este fluxo pode consumir. O objetivo é aproximar-se de um modelo de fluido de serviço [24, 25], de forma que para o fluxo, a rede se apresente como uma linha dedicada de banda passante R. Para tanto, uma mensagem enviada do transmissor para o receptor leva consigo as características do tráfego e, ao longo deste caminho, informações sobre desvios do modelo de fluido são acumuladas. A partir dessas informações sobre os desvios acumulados e das características do tráfego, o receptor pode calcular a banda passante e o espaço em *buffer* [24, 25] a serem reservados em cada roteador para garantir um atraso máximo e nenhuma perda de pacotes. Após o cálculo, o receptor envia o seu pedido de reserva contendo os parâmetros R e B aos roteadores intermediários para que estes reservem os recursos necessários ao Serviço Garantido.

O serviço de Carga Controlada oferece uma qualidade de serviço equivalente ao serviço de melhor-esforço quando a rede não está congestionada. Este tipo de serviço atende adequadamente às aplicações ditas de tempo-real tolerantes, isto porque o serviço garante uma banda ao receptor sem, entretanto, garantir limites com relação ao atraso. Os principais objetivos do serviço são que uma porcentagem elevada de pacotes seja entregue ao destino com sucesso e que o atraso experimentado pela maioria dos pacotes não exceda em muito o atraso mínimo experimentado na média por todos os pacotes do fluxo. Aplicações de tempo real adaptativas encontram-se nessa categoria.

Dois aspectos importantes devem ainda ser destacados relativamente à arquitetura de Integração de Serviços. Primeiramente, a complexidade envolvida na sua implementação e gerenciamento. Esta complexidade traduz-se na utilização de um protocolo de sinalização (RSVP) e na necessidade de gerenciar um volume de fluxo de dados com os respectivos estados (caminho e reserva) em todos os dispositivos de rede. A manutenção de um estado por fluxo e o processamento a eles associado é inviável em roteadores com grande volume de tráfego em uma rede de escala global como a Internet. A impossibilidade de ser escalável limitou a adoção da proposta de Integração de Serviços de forma intensiva na Internet. Devido a esse problema, surgiram algumas propostas de agregação de estados [26] visando a redução da quantidade de estados no classificador, no escalonador de pacotes e no protocolo de sinalização (RSVP). Essa agregação de estados acarreta a perda parcial do isolamento entre os fluxos e impede o fornecimento de garantias determinísticas como as do Serviço Garantido, descaracterizando este serviço. Devido a esses problemas, o uso de RSVP e Integração de Serviços é recomendado somente para redes pequenas e confinadas [27].

O segundo aspecto diz respeito à utilização de tecnologias de enlace que ofereçam naturalmente meios para o controle de QoS, permitindo que certas funcionalidades de QoS da arquitetura, possam ser mapeadas nas facilidade de QoS da camada de enlace. Um exemplo típico nesse caso pode ser ilustrado através da integração da arquitetura IIS com a tecnologia ATM.

1.1.4 Diferenciação de Serviços

Por causa das dificuldades em implementar e empregar a Integração de Serviços, foi especificado pelo IETF (*Internet Engineering Task Force*) a proposta de Serviços Diferenciados (*Differentiated Services*-DS) [28, 29, 10]. A principal meta do IETF é a especificação de uma arquitetura mais escalável e gerenciável para a diferenciação de serviços em redes IP. A premissa inicial para alcançar esta meta foi o tratamento de agregações de tráfego ao invés de fluxos individuais de pacotes nos roteadores, onde um agregado de tráfego é constituido por um conjunto de fluxos com requisições de serviço similares.

O serviço diferenciado possui alta escalabilidade a custa de uma forte agregação de fluxos em umas poucas classes de serviço. Os 6 bits do campo TOS (*Type Of Service*) do IPv4 [30, 31] ou do campo Classe de Tráfego (IPv6) [32] são utilizados pelo serviço diferenciado na realização da marcação de pacotes de fluxos individuais, sendo chamado de campo DS [28]. De acordo com essas marcações, estes fluxos individuais são classificados em agregações de fluxos pré-estabelecidas, as quais são encaminhadas por cada nó, segundo um Comportamento por Nó (*Per-Hop Behavior*- PHB). Um PHB define o comportamento padrão para o encaminhamento de agregações de fluxos gerados por diferentes aplicações em uma classe de serviço. Nós DS-capazes alocam recursos para cada PHB suportado e empregam uma política de escalonamento de forma que o comportamento associado ao PHB seja satisfeito. Os pacotes que durante a classificação não pertencerem a nenhuma agregação pré-estabelecida, recebem o serviço padrão que equivale ao serviço de melhor-esforço atualmente oferecido pela Internet.

A classificação e marcação de pacotes ocorre nas fronteiras de um domínio de Diferenciação de Serviços. Um domínio de Diferenciação de Serviços (domínio DS) é composto por um grupo de nós contíguos que operam um conjunto comum de serviços diferenciados. Deve-se ressaltar que a arquitetura de Diferenciação de Serviços traz as operações mais complexas para as fronteiras do domínio DS, simplificando os mecanismos no interior para aumentar o seu potencial de escalabilidade. Resumidamente, há um estado por fluxo nas fronteiras do domínio DS e um estado por classe de serviço oferecida nos nós localizados no interior deste domínio. A atribuição de PHBs à agregações de fluxos requer que um Contrato de Nível de Serviço (*Service Level Agreement*-SLA) seja estabelecido entre domínios, ou entre usuários e provedores de serviço. Um SLA basicamente especifica as classes de serviço contratadas e a quantidade de tráfego permitida em cada classe.

Além da definição da arquitetura funcional, duas abordagens distintas derivaram das pesquisas em Diferenciação de Serviços: o Serviço Diferenciado Absoluto [33, 34] e o Serviço Diferenciado Relativo [35]. A primeira abordagem tenta satisfazer as mesmas metas que o Serviço Integrado, ou seja, garantir níveis absolutos de desempenho. Porém, sem a manutenção de estados por fluxos em todos os roteadores e com apenas algumas reservas de recursos semi-estáticas, ao invés de um protocolo dinâmico de reserva de recursos. Esta abordagem é baseada em sofisticados mecanismos de controle de admissão por pacote e provisionamento de recursos, a fim de prover garantias para medidas de desempenho absolutas tais como uma taxa de serviço mínima ou um atraso máximo fim-a-fim. A segunda abordagem envolve modelos de serviço que garantem uma ordenação relativa da qualidade de serviço entre classes, ao invés de prover garantias de serviço para uma classe individualmente. Assim, a rede apenas assegura que as classes mais altas receberão melhores serviços do que as classes mais baixas, ficando a cargo do usuário/aplicação selecionar a classe que melhor traduza suas necessidades. Esta abordagem é mais simples em termos de implementação, desenvolvimento e gerenciabilidade da rede, e não necessita de mecanismos para controle de admissão por pacote e provisionamento de recursos.

Vários fatores contribuem para uma significativa diferença entre Serviços Diferenciados e Serviços Integrados. Primeiramente, existe um número limitado de classes de serviço indicadas pelo campo DS. Assim, por ser o serviço alocado na granulosidade de uma classe, a quantidade de informação de estado é proporcional ao número de classes, ao invés do número de fluxos, tornando o Serviço Diferenciado mais escalável. Outro fator a ser ressaltado é que sofisticadas operações como classificação, marcação, policiamento e suavizacão de tráfego, são necessárias apenas nos roteadores das bordas de um domínio. Os roteadores do interior do domínio necessitam implementar apenas a operação de classificação. Isso contribui com uma maior facilidade para implementar e desenvolver Serviços Diferenciados, além de permitir que roteadores do interior do domínio encaminhem os pacotes com uma maior rapidez. A tabela 1.1 resume as principais diferenças entre as arquiteturas DS e IIS.

Tabela 1.1: Comparação das arquiteturas DS e IIS. de tráfego TCP.

Características	**Serviço Integrado**	**Serviço Diferenciado**
Granularidade da diferenciação de serviço	fluxos individuais	agregados de fluxo
Estados em roteadores	por fluxo	por agregação
Base da classificação do tráfego	vários campos do cabeçalho	o campo DS (6 bits) do cabeçalho IP
Tipo da diferenciação de serviço	garantia determinística ou estatística	garantia absoluta ou relativa
Controle de Admissão	requerido (por fluxo)	requerido somente para a diferenciação absoluta (por pacote)
Coordenação da diferenciação de serviço	fim-a-fim	local (por nó)
Escalabilidade	limitada pelo número de fluxos	limitada pelo número de classes de serviço
Gerenciamento da rede	similar a redes comutadas a circuitos	similar a redes IP existentes
Protocolo de sinalização	requerido (RSVP)	não é requerido para esquemas relativos. Esquemas absolutos necessitam de reservas semi-estáticas

1.2 Objetivos e Organização do Texto

A arquitetura de Diferenciação de Serviços (DS) é considerada uma forma escalável e gerenciável de prover suporte a uma gama de aplicações, cujos requisitos de QoS exigem mais da rede do que o serviço de melhor-esforço pode prover. A abordagem de Serviços Diferenciados Relativo constitui uma das derivações das pesquisas em DS e destaca-se por uma maior simplicidade em sua arquitetura. Esta abordagem envolve modelos que garantem uma ordenação da qualidade de serviço entre classes, ao invés de prover garantias de serviço para um classe individualmente. Além disso, provê a diferentes aplicações e usuários a flexibilidade de selecionar a classe de encaminhamento que melhor corresponda a suas restrições de qualidade/custo, considerando a adaptação nos sistemas terminais e aplicações [36]. Com o crescimento em popularidade das aplicações adaptativas, tal como *Real Player* e produtos de Telefonia IP, a abordagem de diferenciação relativa de serviço tem conquistado adeptos interessados em usufruir de suas vantagens e em avançar em pesquisas relacionadas.

Várias formas de se prover diferenciação relativa de serviço para usuários e operadores de rede foram propostas e avaliadas em [36, 37, 38, 39, 40]. As medidas de atraso nas filas e taxa perda de pacotes constituem parâmetros importantes na avaliação de desempenho de um serviço de encaminhamento de pacotes em cada nó. Assim, em [36, 37, 38] foram analisados mecanismos que realizam diferenciação relativa de serviço para essas medidas de desempenho, utilizando-se tráfego de tempo-real insensível ao congestionamento. Porém, nenhum desses trabalhos leva em consideração as características da maior parte do tráfego encontrado na Internet, o tráfego TCP. Pesquisas em [41, 42, 43, 44] mostram que o mecanismo de controle de congestionamento deste protocolo é fortemente influenciado por métricos de desempenho como atrasos nas filas e perdas de pacotes. Esta característica do protocolo TCP torna difícil prever o seu comportamento diante de um modelo de diferenciação relativa de serviço.

O protocolo TCP é responsável pela geração de 85% a 95% dos *bytes* que trafegam hoje pela Internet. Dada sua abrangência, é de fundamental importância que pesquisas relacionadas à provisão de Qualidade de Serviço na Internet levem em consideração as características introduzidas por este protocolo à rede mundial. Desta forma, como contribuição à tais pesquisas e estando ciente da influência que o protocolo TCP exerce no dinamismo do tráfego da Internet, o objetivo principal deste trabalho é avaliar o desempenho do modelo de diferenciação relativa de serviço, quando submetido a um tráfego TCP. Foi desenvolvido no simulador *NS*-

2 [45], um esquema de encaminhamento de pacotes que fornece uma diferenciação proporcional de serviço, uma das formas de se prover Serviços Diferenciados Relativo. Nas simulações realizadas, primeiramente o esquema desenvolvido é validado, onde são utilizadas tanto fontes com taxa de transmissão constante (*Constant Bit Rate* - CBR) quanto fontes do tipo atividade-silêncio e que seguem uma distribuição Pareto (*Pareto*). As análises realizadas nesta seqüência de simulações objetivam verificar a realização da correta diferenciação proporcional pelas partes constituintes do esquema de encaminhamento desenvolvido. Uma segunda série de simulações avalia o desempenho desse esquema de encaminhamento proporcional quando aplicado fontes de tráfego TCP. O objetivo é verificar se as restrições de diferenciação proporcional de atraso e de taxa de perda são atendidas, quando o tráfego existente, reage a tais métricas de desempenho.

Este trabalho está organizado da seguinte forma. O capítulo 2 apresenta a proposta de Diferenciação de Serviços do IETF, descrevendo o modelo proporcional como uma forma de se prover diferenciação relativa de serviço controlável e previsível. Apresenta também o esquema de encaminhamento que segue esse modelo, onde são descritos os mecanismos que o compõe. No capítulo 3 é abordado o funcionamento do protocolo TCP, onde o problema de congestionamento e os algoritmos desenvolvidos para seu controle são relatados, além de apresentar um modelo de desempenho analítico para a vazão do TCP. No capítulo 4 são descritas a configuração e as topologias utilizadas nas simulações realizadas, além de serem analisados os resultados obtidos. Finalmente, o capítulo 5 apresenta as conclusões deste trabalho.

Capítulo 2

Diferenciação de Serviços na Internet

A PRINCIPAL meta da Diferenciação de Serviços, desenvolvida pelo Grupo de Trabalho DiffServ (*Differentiated Services*) do IETF [46], foi a especificação de uma arquitetura escalável e gerenciável para a provisão de QoS em redes IP. Neste capítulo são apresentados os conceitos e abordagens relacionados a esta especificação e descritos os mecanismos que compõem a arquitetura funcional proposta.

2.1 Serviços Diferenciados

A arquitetura de Serviços Diferenciados (*Differentiated Services*-DS) [28, 10] é uma das propostas para o atual estágio de introdução de qualidade de serviço na Internet, buscando o fornecimento de uma gama de serviços de forma escalável. Essa arquitetura constitui uma solução mais simples que a arquitetura de Serviços Integrados na Internet (*Internet Integrated Services*-IIS), que efetua reserva de recursos por fluxo, o que exige dos roteadores a manutenção do estado de cada fluxo. Esta característica é a principal desvantagem da arquitetura de Serviços Integrados na Internet, notadamente quando se tem milhares ou milhões de fluxos trafegando num *backbone* em um dado instante. O receio que o Serviço Integrado seria incapaz, por questões de escalabilidade, de operar em grandes *backbones* IP, motivou o surgimento do Serviço Diferenciado. A arquitetura de Serviços Diferenciados possui alta escalabilidade a custa de uma forte agregação de fluxos em umas poucas classes de serviço.

O Serviço Diferenciado emprega um mecanismo denominado Comportamento por Nó (*Per Hop Behavior*-PHB) que define comportamentos padrões para o encaminhamento diferenciado de pacotes. Caso todos os roteadores do caminho implementem a arquitetura de Serviços Diferenciados[1], o encaminhamento de um pacote marcado para ser tratado por um determinado PHB terá um perfil de encaminhamento uniforme em todo o caminho percorrido. Um PHB trata fluxos gerados por diferentes aplicações e define uma classe de serviço. Os nós capacitados para diferenciação de serviço são responsáveis por empregar uma política de escalonamento de forma que o comportamento associado a cada PHB seja satisfeito. A definição de PHBs é flexível, sendo função de grandezas tais como: recursos (banda passante, *buffer*) e parâmetros de QoS (atraso, *jitter*, taxa de perda).

O Serviço Diferenciado emprega *codepoints* para a identificação de PHBs. Os *codepoints* (DSCP) ocupam 6 bits do campo Tipo de Serviço (*Type Of Service*-TOS) do IPv4 [30, 31] ou Classe de Tráfego do IPv6 [32], agora denominados campo DS (*Differentiated Service*) [28]. A Figura 2.1 ilustra este campo, que é incompatível com a definição existente do campo TOS do IPv4 [30, 47, 31].

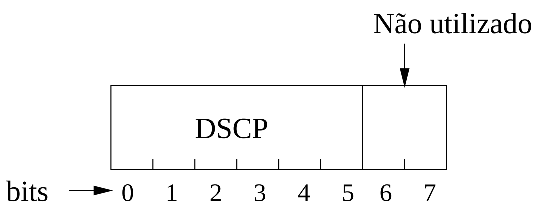


Figura 2.1: Campo DS (*Differentiated Service*).

A atribuição de PHBs aos pacotes IP requer um contrato entre domínios onde são oferecidos Serviços Diferenciados. Um domínio como esse é composto por um conjunto de nós compatíveis com a proposta de Diferenciação de Serviços que compartilham uma mesma política de provisão de serviços. Nós localizados na fronteira

[1]Roteadores com esta característica são denominados nós capacitados para diferenciação de serviço

de um domínio de Diferenciação de Serviço realizam agregações de fluxos, essas por sua vez são encaminhadas pelos nós do interior deste domínio. Na arquitetura de Serviços Diferenciados, distintas diferenciações de serviços podem ser definidas em ambas as direções do fluxo de informações, sendo portanto assimétrica. Os nós de fronteira assumem, dependendo do sentido do tráfego tratado, o papel de nós de ingresso ou egresso. Seja o exemplo da figura 2.2 onde o domínio D_1 (*upstream*) é um cliente corporativo e o domínio D_2 (*downstream*) um ISP (*Internet Service Provider*). Os *codepoints* são atribuídos no roteador de egresso do domínio D_1 (roteador R_1) ou no roteador de ingresso do domínio D_2 (roteador R_2). Um contrato entre D_1 e D_2 denominado Contrato de Nível de Serviço (*Service Level Agreement*-SLA) governa a atribuição de PHBs nos pacotes que trafegam entre os dois domínios. Um contrato de nível de serviço (SLA) pode especificar regras de classificação e remarcação de pacotes, os perfis de tráfego, ações a serem tomadas para o tráfego não conforme com os termos especificados no contrato de serviço e regras de condicionamento de tráfego denominadas Contrato de Condicionamento de Tráfego (*Traffic Conditioning Agreement*-TCA).

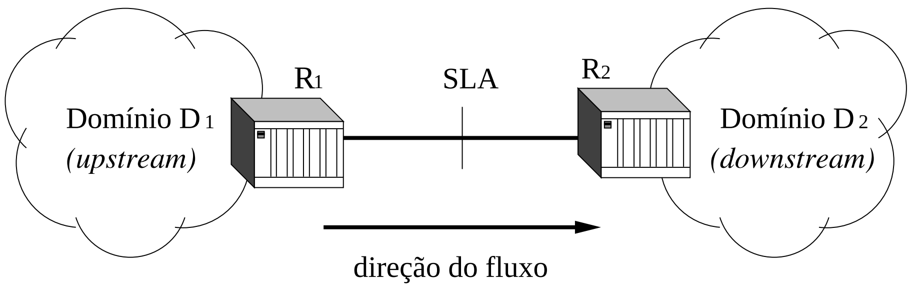


Figura 2.2: Acordo de nível de serviço entre dois domínios.

2.1.1 Arquitetura Funcional

A Figura 2.3 define a Arquitetura de Serviço Diferenciados em termos funcionais. O classificador seleciona pacotes em função de informações contidas no cabeçalho e de regras de classificação e os encaminham para processamento em função do PHB ao qual cada pacote está associado. Caso nenhum PHB seja identificado, um PHB *default* referente ao serviço de melhor-esforço é conferido ao pacote.

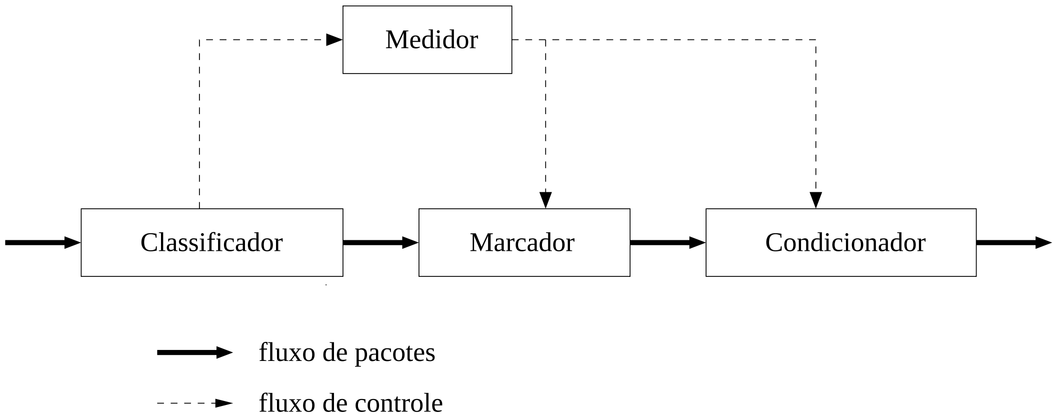


Figura 2.3: Arquitetura funcional do Serviço Diferenciado.

Os demais componentes da arquitetura são responsáveis pela medição, suavização (*shapping*), policiamento e remarcação de pacotes (alteração do valor do *codepoint* presente no pacote) . Essas funções são desempenhadas pelo medidor, suavizador, policiador e marcador, respectivamente. Tais componentes são responsáveis pelo condicionamento de tráfego e visam assegurar que o tráfego ingressante no domínio está em conformidade com os limites impostos pelo SLA/TCA. Os medidores verificam as propriedades temporais do tráfego selecionado pelo classificador, comparando com as propriedades especificadas no SLA/TCA para cada classe de serviço. Os marcadores atribuem um *codepoint* ao pacote, ou alteram um valor previamente atribuído. O *codepoint* atribuído ao pacote é função do estado informado pelo medidor. Por exemplo, o marcador de pacotes poderá atribuir aos pacotes do tráfego excedente numa determinada classe, o *codepoint* do comportamento referente ao serviço do tipo melhor esforço. O suavizador de tráfego retém um ou mais pacotes até que estes estejam em conformidade com o perfil contratado e possam ser encaminhados na rede. O *buffer* utilizado por um suavizador possui tamanho limitado. Portanto, eventualmente, pacotes podem ser descartados quando o tamanho do *buffer* é insuficiente. O policiador descarta os pacotes que excedem o perfil contratado e equivale a um suavizador cujo tamanho do *buffer* é igual a zero. Os pacotes fora do perfil contratado (*out-of-profile*) podem ser descartados, suavizados ou remarcados para uma outra agregação que possua um serviço inferior. A decisão de como tratá-los deve estar especificada no TCA.

As características e mecanismos da arquitetura de Diferenciação de Serviços vem norteando a realização de diversas pesquisas. Dentre eles, pode-se citar as realizadas por A. Ziviane [48, 49, 50, 51] com tráfego de voz sendo aplicado em um ambiente de diferenciação de serviço.

2.2 Abordagens do Serviço Diferenciado

As pesquisas em Serviços Diferenciados vêm seguindo duas direções distintas. A primeira direção é relacionada à abordagem de Diferenciação Absoluta de Serviços [33, 34], onde é garantido a um usuário um perfil de serviço absoluto da rede. Nessa abordagem, o tráfego de um usuário, após passar por um controle de admissão a nível de pacotes, terá garantias da rede do nível de desempenho previamente estabelecido em um contrato de serviço. Parte, ou a totalidade, deste tráfego será rejeitada se não estiverem de acordo com o perfil de tráfego contratado e a rede não puder prover as garantias necessárias.

Os modelos de serviços de Encaminhamento Expresso (*Expedited Forwarding* - EF) [34] e de Encaminhamento Assegurado (*Assured Forwarding* - AF) [33] são exemplos de propostas de PHB para a implementação de Serviços Diferenciados Absolutos. No modelo de serviço de Encaminhamento Expresso, garantias para uma largura de banda nominal, atrasos de fila e taxas de perda mínimas são fornecidas ao usuário ao longo de um certo caminho na rede, independente do comportamento do resto do tráfego nesse caminho. Este tipo de serviço requer o estabelecimento de reservas semi-estáticas de largura de banda e *buffers*, a serem configuradas através do domínio.

O modelo de serviço de Encaminhamento Assegurado também provê aos usuários garantias de largura de banda ao longo de certos caminhos na rede, mas sem garantias estritas de que esta largura de banda estará sempre disponível. Em outras palavras, o serviço Assegurado é mais baseado no provisionamento e em garantias estáticas do que em reservas de largura de banda para cada usuário. Quatro classes de serviço são definidas pelo PHB AF, cada uma com uma certa quantidade de recursos (memória e largura de banda) alocada. Em cada classe, os pacotes podem ser marcados com uma de três preferências para descarte. Havendo congestionamento, a preferência de descarte determina a importância relativa entre os pacotes de uma mesma classe. Pesquisas relacionadas à avaliação do Serviço Assegurado foram realizadas em [52, 53, 54]. No contexto dos modelos de serviços existentes, a Diferenciação Absoluta de Serviço fornece garantias absolutas de desempenho estabelecidas nos contratos de serviço, o que torna necessária a existência de provisionamento de recursos e de controle de admissão a nível de pacotes. Essa necessidade introduz uma considerável complexidade na infra-estrutura de tarifação empregada, nas políticas inter-domínios necessárias e em sua monitoração, e na manutenção da arquitetura de Qualidade de Serviço requerida.

A segunda direção das pesquisas em Serviços Diferenciados está relacionada à abordagem de Diferenciação Relativa de Serviços [35]. Essa abordagem envolve modelos que garantem uma ordenação da qualidade de serviço entre classes, ao invés de prover garantias de serviço para uma classe individualmente. Neste caso, a única garantia da rede é que uma classe superior receberá um melhor serviço do que uma classe inferior. Além disso, a quantidade de serviço recebida por uma classe e a qualidade percebida por um usuário dependerão da carga atual em cada classe. Assim, espera-se que usuários/aplicações adaptem suas necessidades baseados no nível de desempenho observado em suas classes, ou desloquem para uma classe melhor, se as restrições de custo permitirem. Neste contexto, aplicações e usuários não obtêm uma garantia de nível de desempenho absoluta, tal como um limite mínimo de atraso fim-a-fim ou de largura de banda, o que torna desnecessário os mecanismos de provisionamento de recursos e controle de admissão a nível de pacote. Como conseqüência, obtém-se uma considerável simplicidade em termos de implementação, desenvolvimento e gerenciabilidade da rede.

O PHB Seletor de Classe [28], padronizado pelo IETF, segue o modelo de Diferenciação Relativa de Serviço, onde é utilizado para efeitos de classificação, o subcampo Precedência que ocupa os três primeiros bits do campo TOS do protocolo IPv4. Assim, oito classes de serviço podem ser definidas por esse PHB, sendo estipulado que quanto maior o valor do *codepoint*, (correspondente a uma classe) mais prioritário é o tráfego a ele associado. Dependendo do cenário em desenvolvimento, a classificação de pacotes em diferentes classes pode também ser realizada pela própria aplicação, ficando a seu encargo a adaptação dinâmica da classificação de seus pacotes, baseada nas medidas de atraso e perdas observadas nas classes.

O modelo de Serviço Diferenciado Relativo deve ser fortemente acoplado a um esquema baseado em tarifas ou em políticas, a fim de fazer com que classes superiores custem mais ou tenham o uso mais restrito do que classes inferiores. Caso contrário, a classe mais alta será usada de forma irrestrita por qualquer usuário e a diferenciação relativa não será efetiva. A pesquisa realizada neste trabalho concentra-se na avaliação do comportamento do modelo de serviço relativo e a próxima seção define conceitualmente esse modelo.

2.3 Serviço Diferenciado Relativo

A premissa básica da diferenciação relativa é que o tráfego da rede é agrupado em N classes de serviço, onde a classe i é melhor (ou no mínimo menos pior), em termos de métricas de desempenho para encaminhamento de pacotes em cada nó, do que a classe $(i-1)$, para $1 < i \leq N$. Aplicações e usuários, neste contexto, podem selecionar dinamicamente a classe que melhor satisfaça suas restrições de preço e qualidade, sem garantias a priori para o nível de desempenho atual de cada classe.

Algumas formas de prover diferenciação relativa de serviço, como Priorização Estrita (*Strict Priorization*), Diferenciação por Capacidade (*Capacity Differentiation*) e Diferenciação por Preço (*Price Differentiation*), foram tratadas em [37, 39, 40] e são descritas a seguir.

- **Priorização Estrita:** nesta política, classes superiores são servidas antes das classes inferiores. Apesar desta abordagem manter o ordenamento de qualidade desejado entre classes, ou seja, classes superiores são sempre melhores, ela apresenta alguns problemas. Primeiro porque, se as classes superiores estão persistentemente ocupadas, isso pode resultar em longos períodos onde as classes inferiores não seriam servidas. Além disso, não apresenta uma diferenciação controlável, por que não provê qualquer meio de ajustar o espaçamento de qualidade entre classes. A habilidade de se ajustar o espaçamento entre as classes independentemente da carga das mesmas, constitui uma propriedade crucial para o modelo de diferenciação relativa.

- **Diferenciação por Preço:** também referenciada como *Paris Metro Pricing* (PMP) [39], essa política se baseia em preço, ao invés de mecanismos de encaminhamento especiais para prover uma diferenciação relativa entre classes. Essa diferenciação segue o princípio de que sendo as classes mais altas mais caras, elas terão cargas mais baixas e, por conseguinte, terão qualidades de serviço melhores. Porém, este tipo de mecanismo por preço, pode ser efetivo apenas em longas escalas de tempo, especialmente quando a tarifa das classes não pode ser modificada freqüentemente. Quando muitos usuários "ricos" tornam-se ativos ao mesmo tempo, causando sobrecarga em classes mais altas, essas passarão a oferecer uma qualidade de encaminhamento pior do que a oferecida por classes mais baixas. Essa situação geraria um caso de inconsistência, caracterizando uma diferenciação de classe não previsível.

- **Diferenciação por Capacidade:** nessa política é alocada uma quantidade maior de recursos de encaminhamento (largura de banda ou espaço no *buffer*) para classes mais altas, o que é realizado de acordo com a carga esperada em cada classe. O WFQ (*Weighted-Fair-Queueing*) [55, 24, 40] é um exemplo de escalonador que pode ser utilizado para este tipo de diferenciação. No caso de diferenciação de atraso, esse escalonador pode ser usado para distribuir a largura de banda do enlace entre as classes, a fim de que as classes mais altas tenham uma maior razão da taxa de serviço pela taxa de chegada. Porém, o atraso atual nas filas, além da largura de banda compartilhada, depende da carga e da rajada de tráfego em cada classe [56]. Assim, apesar da diferenciação da largura de banda ser controlável, a diferenciação de atraso não é, pois rajadas de tráfego e altas cargas em classes superiores podem dificultar a obtenção de atrasos de fila mais baixos para essas classes. Desta forma, é importante fornecer aos usuários a garantia de que eles sempre encontrarão melhores serviços em classes mais altas. Logo, os mecanismos de encaminhamento de pacotes devem ser capazes de prover uma diferenciação relativa consistente entre classes, independente da carga das mesmas, o que caracteriza uma diferenciação previsível.

As desvantagens dos esquemas de diferenciação apresentados anteriormente destacam duas características importantes que um modelo de diferenciação relativa de serviço deve possuir:

- ser **controlável**, o que corresponde a permitir que os operadores de rede, baseado em critérios selecionados, ajustem o espaçamento de qualidade entre as classes, e
- ser **previsível**, o que corresponde a uma diferenciação consistente onde as classes superiores são melhores que classes inferiores, independentemente das variações de carga nas classes.

Objetivando atender a esses requisitos, o Modelo de Diferenciação de Serviço Proporcional foi proposto e avaliado por Dovrolis et al. [37]. Na seção seguinte são apresentadas as características e fundamentos principais deste modelo.

2.3.1 Modelo de Diferenciação de Serviço Proporcional

O Modelo de Diferenciação Proporcional estabelece que certas métricas de desempenho para o encaminhamento de pacotes em cada nó, devem ser proporcionais a parâmetros de diferenciação de qualidade que o operador de rede escolhe. Através destes parâmetros de diferenciação, o operador de rede pode controlar o espaçamento relativo de qualidade entre as classes oferecidas, baseado em restrições de preço e políticas adotadas pelo operador.

Assim, sendo q_i uma medida de desempenho para a classe i, o modelo de diferenciação proporcional estabelece a restrição

$$q_i = \frac{c_i}{c_j} * q_j \qquad (i, j = 1, 2, ...N), \tag{2.1}$$

onde $c_1 < c_2 < ... < c_N$ são parâmetros de diferenciação de qualidade.

Pela Equação 2.1, pode-se verificar que a relação de proporcionalidade entre as mesmas permanecerá constante (o modelo é previsível) e relativamente controlada pelo operador da rede, devido aos parâmetros de diferenciação de qualidade c_i, mesmo que a carga das classes varie.

Pode-se obter um modelo de diferenciação proporcional que empregue o atraso médio ocorrido nas filas e a taxa média de perda de pacotes como métricas de desempenho de uma classe. Assim, para todo par de classes i e j, formula-se matematicamente a diferenciação proporcional de atraso de fila, como mostra a Equação 2.2.

$$\overline{d_i} = \frac{ddp_i}{ddp_j} * \overline{d_j} \qquad (i, j = 1...N) \tag{2.2}$$

onde $\overline{d_i}$ é o atraso médio sofrido pelos pacotes na fila da classe i, e os parâmetros ddp_i são Parâmetros de Diferenciação de Atraso (*Delay Differentiation Parameters*-DDP) que o operador de rede escolhe. Devido às classes superiores serem melhores, elas devem prover atrasos mais baixos, assim, os DDPs são ordenados da seguinte forma: $ddp_1 > ddp_2 > ... > ddp_N > 0$.

De forma similar, para a métrica de perda de pacotes, seja $\overline{l_i}$ a taxa média de perda de pacotes para a classe i. O modelo de diferenciação proporcional, neste caso, estabelece que as taxas médias de perda de pacotes devem ser espaçadas, como especificado na Equação 2.3.

$$\overline{l_i} = \frac{ldp_i}{ldp_j} * \overline{l_j} \qquad (i, j = 1...N) \tag{2.3}$$

onde os parâmetros ldp_i são Parâmetros de Diferenciação de Taxa de Perda (*Loss rate Differentiation Parameters*-LDP) que o operador de rede escolhe, sendo ordenados da forma: $ldp_1 > ldp_2 > ... > ldp_N > 0$, pois classes superiores possuem taxas de perda menores.

Analisando as Equações 2.2 e 2.3, pode-se constatar que as métricas de desempenho de atraso médio da fila e taxa média de perda de pacotes da classe i são proporcionais ao da classe j em relação aos parâmetros de diferenciação de qualidade ddp_i/ddp_j e ldp_i/ldp_j, respectivamente. Esses parâmetros são empregados no ajuste do espaçamento entre as mesmas.

Escalonadores e descartadores podem ser devidamente ajustados para realizarem uma diferenciação proporcional de serviço em relação às métricas de desempenho de atraso nas filas e taxa de perda. Assim, um esquema de encaminhamento proporcional pode ser obtido aplicando-se esses componentes e um gerenciador de *buffer*.

2.4 Comparação entre as Abordagens de Diferenciação de Serviço

A arquitetura de Serviços Diferenciados provê formas para estender o paradigma de encaminhamento de pacotes na Internet, com mecanismos de diferenciação de serviço escaláveis e de fácil emprego. Na abordagem de diferenciação absoluta, esses mecanismos oferecem garantias absolutas, as quais são úteis principalmente, para aplicações ditas "não elásticas" e para cenários de desenvolvimento que requerem medidas de serviço específicas, tal como Redes Privadas Virtuais (VPN - *Virtual Private Networks*). Na abordagem de diferenciação relativa, a arquitetura de Serviços Diferenciados provê às diferentes aplicações e usuários, a flexibilidade de selecionar a classe de serviço que melhor corresponda a suas restrições de qualidade/custo. Esse tipo de diferenciação é mais adequado para aplicações ditas "elásticas" que possuem a habilidade de se adaptarem ao nível de desempenho das classes de serviço ofertadas. O grande crescimento da utilização de aplicações adaptativas indicam que usuários podem tolerar variações na qualidade, o que vem impulsionando o emprego da diferenciação relativa de serviço na Internet.

Desta forma, a diferenciação absoluta e relativa são capazes de suprir aplicações com exigências diferentes: aplicações "elásticas" e "não elásticas". Portanto, é previsto a existência de ambos esses tipos de aplicações no futuro da Internet, o que torna os mecanismos de diferenciação relativa e absoluta igualmente importantes para a viabilidade comercial das inter-redes do futuro.

Serviços diferenciados absolutos são oferecidos por Modelos de Encaminhamento por Nó denominados de PHB EF e de PHB AF. Por outro lado, o Modelo de Diferenciação Proporcional constitui uma forma controlável e previsível de se fornecer serviços diferenciados relativos. A Tabela 2.1 apresenta, de uma forma sucinta, as caracterísitcas desses modelos.

Tabela 2.1: Características de três modelos DS.

Características	**PHB EF**	**PHB AF**	**Diferenciação Proporcional**
Tipo de serviço	estritamente absoluto	depende do provisionamento da rede	estritamente relativo
Operação no ingresso da rede	policiamento do tráfego	marcação (IN/OUT) ou precedência de descarte	avaliação do uso da classe
Adaptação dos pontos terminais	não requerida	pode ajudar	importante ou requerida
Agente *bandwidth-broker* [a]	requerido	não requerido	não requerido

[a]Decide se os recursos disponíveis no domínio são suficientes para atender a demanda da aplicação e realiza a configuração das regras de condicionamento de tráfego nos roteadores.

Recentemente foi proposto por Jagannathan et al. [57] um novo PHB para Serviços Diferenciados Relativos em redes IP, o *Proportional Forwarding PHB*. Neste PHB, cada fluxo ou usuário da rede recebe uma alocação de largura de banda proporcional a um parâmetro de serviço que varia de um usuário para outro. Esse parâmetro é denominado de taxa de informação contratada (*Subscribed Information Rate - SIR*). Desta forma, um mecanismo de marcação no roteador de ingresso do domínio de Diferenciação de Serviços calcula a razão da taxa de envio da fonte pela sua taxa de informação contratada e configura o *Differentiated Services CodePoint* (DSCP) de cada pacote. Usuários com SIR mais altos pagam mais e, por conseguinte, receberão uma alocação de largura de banda mais alta do que os usuários que pagam por SIR menores. Nos roteadores do núcleo do domínio DS, um esquema de gerenciamento ativo de fila baseado em vários níveis de *thresholds* (n-RED por exemplo), usará os DSCPs para a diferenciação entre pacotes durante o congestionamento. Assim, quando necessário, o gerenciador ativo de fila selecionará pacotes a serem descartados de acordo com a marcação hierárquica dos DSCPs nos pacotes.

2.4.1 Esquema de Encaminhamento Proporcional

O esquema de encaminhamento de pacotes da Figura 2.4 foi proposto em [38] e pode ser interpretado como uma forma de se empregar o Modelo de Diferenciação de Serviço Proporcional. Nesse esquema, N classes de serviço são oferecidas, estando uma fila lógica associada a cada classe. Os pacotes, a medida que chegam e após a sua classificação, são marcados e encaminhados para as filas correspondentes. O esquema de encaminhamento proporcional é composto basicamente de três componentes:

- o escalonador de pacotes que, de acordo com as restrições descritas na Equação 2.2 e visando garantir a proporcionalidade de atraso estabelecidas entre as classes, seleciona um pacote a ser enviado. Assim, o escalonador procura distribuir dinamicamente e proporcionalmente a largura de banda do enlace de saída entre as classes existentes. Os algoritmos *Proportional Queue Control Mechanism* (PQCM) [58], *Mean Delay Proportional* (MDP) [59], *Backlog-Proportional Rate* (BPR) e *Waiting-Time Priority* (WTP) [37], são exemplos de escalonadores que realizam diferenciação proporcional de atraso;

- o gerenciador de *buffer* que monitora a ocupação das agregações de pacotes esperando nas filas e, conforme o mecanismo de gerenciamento empregado, decide quando um descarte deve ocorrer. Um exemplo simples de gerenciador é o *DropTail*, no qual um pacote é descartado quando não existe mais espaço disponível no *buffer*. O algoritmo *DropTail-CS* (*DropTail-Complete Sharing*) se comporta da mesma forma que o *DropTail*, porém emprega a política de organização de *buffer* de compartilhamento completo, onde o espaço total disponível para o *buffer* é compartilhado por qualquer pacote, independente da classe a qual ele pertença. Outro exemplo de gerenciador, mais sofisticado, é o RED [60];

- o descartador de pacotes que, ao receber a sinalização do gerenciador de *buffer* de que um descarte deve ocorrer, seleciona a classe que terá um pacote descartado. Tal seleção seguirá um algoritmo de descarte que procurará manter as restrições descritas na Equação 2.3 e garantir a proporcionalidade de taxa de perda estabelecida entre as classes. Os algoritmos *Proportional Loss Rate* com memória infinita (PLR(∞)), e com memória M (PLR(M)) [38], são exemplos de descartadores que realizam diferenciação proporcional de taxa de perda.

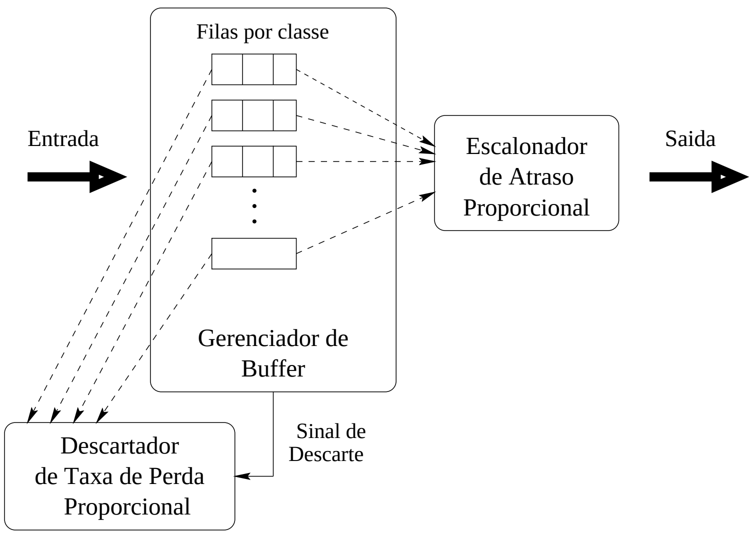


Figura 2.4: Esquema de Encaminhamento Proporcional.

Nas seções seguintes são apresentadas as propostas de escalonadores e descartadores que procuram se aproximar do modelo de diferenciação proporcional de atraso e de taxa de perda. Vale ressaltar que todos os escalonadores apresentados são conservativos, ou seja, se uma classe escolhida para ser escalonada encontra-se vazia, o escalonador nunca ficará parado se existe pacotes em outras classes a serem escalonados, e realizará a seleção de outra classe a ser servida.

Escalonador PQCM

A semântica de serviço deste escalonador assegura que o atraso de transferência de classes diferenciadas sejam proporcionais. Para tanto, o desempenho de uma classe mais prioritária é proporcional a de uma classe menos prioritária, possibilitando um fácil gerenciamento e controle da ocupação das filas.

O objetivo do PQCM (*Proportional Queue Control Mechanism*) [58] é prover um serviço onde o atraso de espera na fila mais prioritária ($d^{+}(t)$) seja sempre menor do que o da fila menos prioritária ($d^{-}(t)$), ou seja:

$$d^{+}(t) = \alpha * d^{-}(t) \qquad (0 < \alpha < 1), \tag{2.4}$$

onde o parâmetro α é usado para ajustar o nível de diferenciação.

Para que esse objetivo seja satisfeito, a taxa de saída das filas mais e menos prioritárias no tempo t, é reajustada a cada Δ segundos, de acordo com as Equações 2.5 e 2.6:

$$C^{+}(t) = \frac{C * e^{+}(t)}{e^{+}(t) + \alpha * e^{-}(t)}; \tag{2.5}$$

$$C^{-}(t) = C - C^{+}(t), \tag{2.6}$$

onde C corresponde à capacidade total do enlace de saída e os parâmetros $e^{+}(t)$ e $e^{-}(t)$ correspondem à ocupação das filas mais e menos prioritária, respectivamente.

Além do reajuste da taxa de saída a cada Δ segundos, o escalonamento é realizado de forma que a fila mais prioritária será servida primeiro se não estiver vazia e se, até o momento, ela tiver sido servida menos vezes do que a fila menos prioritária, ou seja, $s^{+} < s^{-}$. Mais detalhes sobre a implementação deste escalonador, bem como análises de seu desempenho, podem ser obtidas em [58].

As simulações realizadas por Moret e Fdida [58] demonstraram ser o mecanismo de escalonamento PQCM satisfatório para controlar proporcionalmente o atraso de uma classe mais prioritária e melhorar significativamente o serviço provido para o usuário final. Porém, esse mecanismo traz como desvantagem uma grande complexidade envolvida em sua implementação, resultado da necessidade de atualização das taxas de saída de cada classe a cada Δ segundos. As análises apresentadas por Moret, com o propósito de facilitar a compreensão do mecanismo de escalonamento proposto, consideraram apenas a diferenciação entre dois fluxos agregados. A diferenciação entre mais de duas classes de serviço seja possível, porém é provável que a complexidade envolvida na manutenção das taxas de saída de todas as classes, dificulte em muito a implementação deste mecanismo de escalonamento.

Escalonador MDP

O escalonador MDP (*Mean Delay Proportional*) [59] visa garantir uma diferenciação relativa de atraso entre duas ou mais classes de serviço. "Pesos de atraso" são associados às classes de serviço e é assegurado que os atrasos médios percebidos por pacotes em quaisquer classes serão inversamente proporcionais aos pesos de atraso associados às classes as quais fazem parte. Assim,

$$\frac{\overline{d_i(t)}}{\overline{d_j(t)}} = \frac{\Delta_j}{\Delta_i} \qquad (t \to \infty), \tag{2.7}$$

onde $\overline{d_i(t)}$ é o atraso médio experimentado por pacotes da classe i no tempo t e o parâmetro Δ_i é o "peso de atraso" atribuído a essa classe.

Quando um pacote necessita ser selecionado para a transmissão no tempo t, o escalonador MDP selecionará uma fila j para realizar esta transmissão satisfazendo a restrição

$$j \leftarrow max_i \frac{\overline{d_i(t)} * \Delta_i}{s_i(t)}, \tag{2.8}$$

onde $s_i(t)$ é o número de pacotes servidos da classe i até o tempo t.

Em outras palavras, o pacote da cabeça da fila a ser selecionado para transmissão, será aquele que pertencer à fila que possuir um máximo atraso médio normalizado no tempo t, baseado nas chegadas de pacotes até o tempo presente (Equação 2.8). Satisfazendo essa restrição de escalonamento, todas as filas, não vazias, observarão o mesmo atraso médio normalizado, onde o atraso médio percebido por uma classe será inversamente proporcional ao seu peso de atraso (Equação 2.7).

As análises realizadas por Nandagopal et al. [59] indicam que o algoritmo MDP garante a diferenciação proporcional de atraso desejada entre as classes de serviço. Além disso, permite uma convergência mais precisa (com uma margem de erro inferior a 5%) para essa diferenciação de atraso em condições de baixa e alta carga de tráfego e em curtas escalas de tempo. Isso é possível pois os pacotes na cabeça das filas refletem a carga do sistema até o instante que eles são servidos, ao invés de refletirem a carga do sistema somente a partir do momento que chegaram na fila [2].

Escalonador BPR

A idéia básica do escalonador BPR (*Backlog-Proportional Rate*) [37] é ajustar dinamicamente a taxa de serviço de uma classe baseada na carga dessa classe (Equação 2.9) a fim de permitir uma diferenciação proporcional de atraso entre as classes de serviço ofertadas. Assim,

[2]Conseqüência do emprego de $s_i(t)$ no cálculo de seleção da fila j.

$$\frac{r_i(t)}{r_j(t)} = \frac{sdp_i(t)}{sdp_j(t)} * \frac{q_i(t)}{q_j(t)}, \tag{2.9}$$

onde $r_i(t)$ é a taxa de serviço que é associada à fila da classe i no tempo t e $q_i(t)$ é a ocupação dessa fila nesse mesmo tempo t. As variáveis sdp_i são os Parâmetros de Diferenciação do Escalonador (*Scheduler Differentiation Parameters* - SDP) que o operador de rede escolhe, onde as razões SDPs tendem para o inverso das razões DDPs (*Delay Differentiation Parameters*), $\frac{ddp_i}{ddp_j} = \frac{sdp_j}{sdp_i}$. Desta forma, o SDP determina o quanto que a taxa de serviço de uma certa classe aumenta com o tempo em relação a ocupação de sua fila, sendo ordenado como $sdp_1 < sdp_2 < ... < sdp_N$, pois as classes superiores devem ter taxas de serviço maiores, conduzindo a atrasos de fila menores.

Assim, quando a ocupação relativa de uma fila é bastante pequena, a taxa relativa de serviço dada para essa fila também será pequena. Como resultado, os últimos pacotes em uma fila podem experimentar um atraso muito maior do que o experimentado por seus antecessores.

Algumas simulações levando em consideração várias condições da rede, foram realizadas por Dovrolis et al. [37]. Tais simulações objetivaram avaliar as condições sobre as quais este escalonador é infactível, ou seja, em que condições esse escalonador desvia-se do modelo proporcional de atraso discutido na Seção 2.3.1. Resultados obtidos indicam que o escalonador BPR se aproxima do modelo de diferenciação proporcional de atraso durante condições de alta carga de tráfego (quando a utilização do enlace tende a 100%), em longas escalas de tempo. Em curtas escalas de tempo também ocorre essa aproximação, porém esta é menos precisa do que a apresentada pelos escalonadores MDP e WTP.

Escalonador WTP

No escalonador WTP (*Waiting-Time Priority*) [37], a prioridade de um pacote em ser escalonado, aumenta proporcionalmente com seu tempo de espera na fila, sendo classificado como um escalonador prioritário. Quanto mais carregada encontra-se uma classe, maior sua fila, mais tempo um pacote esperará para chegar na cabeça da fila e, assim, maior será sua prioridade de ser escalonado. Desta forma, baseado na carga de cada classe, ele distribui dinamicamente a taxa de serviço entre classes, mantendo-se porém, a diferenciação consistente (classes maiores são melhores). Especificamente, a prioridade de um pacote na fila da classe i no tempo t é

$$p_i(t) = w_i(t) * sdp_i, \tag{2.10}$$

onde, $w_i(t)$ é o tempo de espera do pacote no tempo t e sdp_i são os Parâmetros de Diferenciação do Escalonador (*Scheduler Differentiation Parameters* - SDP) que o operador de rede escolhe, estando as razões SDP tendendo para o inverso das razões DDP ($\frac{ddp_i}{ddp_j} = \frac{sdp_j}{sdp_i}$). Desta forma, o SDP determina a taxa com que a prioridade dos pacotes de uma certa classe aumenta com o tempo, sendo ordenado como $sdp_1 < sdp_2 < ... < sdp_N$, pois esta taxa é maior para classes superiores.

Este escalonador foi originalmente estudado por L. Kleinrock [61] e posteriormente, Dovrolis et al. o avaliou como um escalonador proporcional de atraso em um ambiente sem perdas (*buffers* grandes) e em um ambiente com perdas (*buffers* finitos) [37, 38]. As análises realizadas mostram que o escalonador WTP:

- se aproxima bastante do modelo de diferenciação proporcional da Equação 2.2 quando a utilização do enlace tende a 100%, ou seja, em condições de alta carga - região de operação onde mecanismos de diferenciação de serviço são necessários.

- é mais preciso para curtas escalas de tempo e alta carga, do que o escalonador BPR (*Backlog-Proportional Rate*), também proposto e avaliado em [37].

Tabela 2.2: Comparação entre os algoritmos de escalonamento apresentados.

Características	PQCM	MDP	BPR	WTP
Complexidade de implementação	Baixa: 2 classes Alta: mais de 2 classes	Alta	Baixa	Baixa
Funcionamento em escala de tempo	não analisado	Escalas de tempo longas e curtas	Escalas de tempo longas	Escalas de tempo longas e curtas
Funcionamento de acordo com a carga	não analisado	Baixa e alta carga de tráfego	Alta carga de tráfego	Alta carga de tráfego
Critério de escalonamento	Quantidade acumulada de serviço	Atraso médio das filas e número de pacotes servidos	Ocupação da fila	Tempo de espera do pacote na fila

O escalonador proporcional MDP proposto em [59], apresenta uma melhor convergência para o modelo de diferenciação proporcional da Equação 2.2 do que o escalonador WTP. Este comportamento é observado em curtas escalas de tempo e em condições de baixas e altas cargas de tráfego. Por outro lado, o escalonador WTP apresenta grande simplicidade em sua implementação, o que não pode ser encontrado no escalonador MDP.

Descartador PLR(∞)

Este descartador foi originalmente proposto e avaliado por Dovrolis et al. [38]. Ele possui o princípio básico de manter uma estimativa da taxa de perda (l_i) em cada classe e de selecionar, quando necessário, a classe que possuir menor razão l_i/ldp_i para ter um pacote descartado. Agindo desta forma, reduz-se a diferença das taxas de perda normalizadas entre as classes envolvidas, tendendo a equalizá-las. Mais especificamente, a seleção da classe apropriada à realização do descarte é determinada por

$$classe = min \frac{D_i}{A_i * ldp_i} \qquad (l_i = \frac{D_i}{A_i}), \tag{2.11}$$

onde, A_i é a quantidade de pacotes que chegaram na classe i e D_i é a quantidade de pacotes descartados da classe i até aquele momento. As variáveis ldp_i são os Parâmetros de Diferenciação de Taxa de Perda (*Loss rate Differentiation Parameters* - LDP), onde $ldp_1 > ldp_2 > ... > ldp_N > 0$.

Uma questão importante neste descartador diz respeito à definição e medição da taxa de perda estimada, ou seja, de l_i. Este parâmetro é medido usando simples contadores para as chegadas e descartes de pacotes em cada classe. Tais contadores só serão reinicializados na ocorrência de *overflow* [3], armazenando assim uma longa história desses acontecimentos. Por isso, é denominado de PLR com memória infinita. Essa característica o torna menos adaptativo às mudanças na distribuição da carga das classes. Por outro lado, experimentos realizados por Dovrolis et al. [38], demonstram que o PLR(∞) é mais preciso e apresenta menores desvios em atender a proporção de taxa de perda estabelecida entre as classes de serviço. Isto é conseqüência da utilização de estimativas baseadas em uma longa seqüência de chegadas de pacotes.

Outra questão importante a se analisar refere-se à factibilidade do modelo de diferenciação proporcional, o que significa analisar em que condições este modelo não é atendido. O descartador PLR(∞) desvia-se deste modelo quando a classe com mínima taxa de perda normalizada ($min\ l_i/ldp_i$) está vazia (*idle*), forçando o descarte de pacotes de outra classe. Experimentos realizados [38] mostram que fatores como: parâmetros de diferenciação especificados, a ocupação média das filas

[3] Pelo menos quatro bilhões de chegadas de pacotes, correspondentes às variáveis com tamanho de 32-bits

e a distribuição da carga nas classes podem afetar indiretamente a factibilidade do modelo.

O descartador PLR(M), também proposto e avaliado por Dovrolis et al., possui a característica de ser mais adaptativo às mudanças na distribuição da carga das classes, conseqüência da utilização de estimativas baseadas em uma seqüência M de chegadas de pacotes, onde $M \ll \infty$ [38]. Apesar disso, o PLR(∞) foi escolhido como o descartador proporcional a ser utilizado nas simulações apresentadas neste trabalho, devido à simplicidade envolvida em sua implementação. Além disso, as conexões de longa duração empregadas nas simulações, quando atingem o estado de equilíbrio [41, 44], apresentam menos variações na distribuição da carga.

Descartador PLR(M)

O descartador PLR(M) (*Proportional Loss Rate* com memória M), também foi proposto e avaliado por Dovrolis et al. [38]. Esse descartador também possui o princípio básico de manter uma estimativa da taxa de perda (l_i) em cada classe e de selecionar, quando necessário, a classe que possuir menor razão l_i/ldp_i para ter um pacote descartado. Porém, a taxa de perda estimada de uma classe i, l_i, é baseada na quantidade de pacotes descartados nas últimas M chegadas de pacotes. Para tanto, o descartador armazena informações do número de chegadas $A_i(M)$ e do número de descartes $D_i(M)$ da classe i nas últimas M chegadas de pacotes. Mais especificamente, a seleção da classe apropriada à realização do descarte é determinada por

$$classe = min \frac{D_i(M)}{A_i(M) * ldp_i} \qquad (l_i(M) = \frac{D_i(M)}{A_i(M)}), \tag{2.12}$$

onde, ldp_i é o Parâmetro de Diferenciação de Taxa de Perda (*Loss rate Differentiation Parameters* - LDP).

O PLR(M) se diferencia do PLR(∞) por ter seus contadores de chegadas e descartes de pacotes em cada classe, inicializados a cada M chegadas de pacotes.

Assim, o PLR(M) tenta obter uma diferenciação proporcional de taxa de perda em toda janela de tempo de M chegadas, ou invés de sucessivas janelas de tempo de M chegadas, como no caso do PLR(∞). Simulações realizadas por Dovrolis et al. [38] exibem que essa característica o torna mais adaptativo às mudanças na distribuição da carga das classes, sendo mais indicado para casos em que essa distribuição é fortemente não-estacionária. Porém, o descartador PLR(M) é menos robusto do que o PLR(∞), por apresentar maiores desvios do modelo de diferenciação proporcional de taxa de perda da Equação 2.3. A razão para isso é que as estimativas da taxa de perda são mais robustas quando são baseadas em seqüências mais longas de chegadas de pacotes, desde que rajadas nas taxas de chegada e nos tempos de descarte tendem a serem suavizadas.

O Modelo de Diferenciação Proporcional apresentado neste capítulo leva em consideração as métricas de desempenho de atraso nas filas e de taxa de perda de pacotes. Pesquisas sobre o protocolo TCP [41, 42, 43, 44] mostram que tais métricas exercem forte influência no comportamento desse protocolo, responsável pela geração de 95% dos pacotes que trafegam hoje na Internet. Assim, ciente dessa realidade de tráfego, do crescimento do número de aplicações adaptativas na rede mundial e das facilidades introduzidas pelo serviço diferenciado relativo, constitui objeto de estudo deste trabalho a avaliação de um ambiente de diferenciação proporcional em tráfego TCP. No capítulo que se segue é apresentada uma revisão bibliográfica das características comportamentais do protocolo TCP.

Capítulo 3

Os Mecanismos do Protocolo TCP

O TCP (*Transmission Control Protocol*) é um protocolo de transporte que foi especialmente projetado para prover uma transmissão confiável de dados através de uma inter-rede não confiável. Esse protocolo possui a capacidade de se adaptar dinamicamente às propriedades da inter-rede e ser robusto na presença de muitos tipos de falhas. Assim, fica claro que o desempenho do protocolo TCP é de fundamental importância para o desempenho global da Internet. Atualmente, a maior parte do tráfego da Internet usa o TCP como protocolo de transporte.

3.1 O Protocolo TCP

Depois de um crescimento explosivo nos últimos anos, a Internet se tornou a maior rede mundial de comunicação de dados. O TCP (*Transmission Control Protocol*) é o protocolo de transporte utilizado por grande parte das aplicações que hoje são utilizadas nessa rede mundial e que, dentre outras características importantes, oferece uma comunicação confiável "fim-a-fim". Em [62] o TCP foi formalmente definido, sendo posteriormente mais detalhado e estendido em [47, 63].

Os protocolos TCP e IP (*Internet Protocol*) representam o essencial da arquitetura Internet, de forma que o modelo funcional desta rede, nomeado a partir deles, é conhecido como a "pilha TCP/IP". A pilha de protocolos da arquitetura TCP/IP contém dois protocolos na camada de transporte: UDP (*User Datagram Protocol*) e TCP (*Transmission Control Protocol*). Apesar das inúmeras diferenças entre os dois protocolos, pode-se afirmar que a principal consiste na confiabilidade do recebimento dos dados por parte do destino.

Os protocolos da camada de transporte, o UDP e o TCP, se utilizam dos serviços da camada inferior, no caso, os serviços providos pelo protocolo IP. O protocolo IP que por sua vez se baseia no princípio de melhor esforço (*best-effort*), fornece um serviço não confiável, onde pacotes são descartados em caso de esgotamento de recursos. O protocolo UDP não faz uso de nenhum mecanismo para prover confiabilidade na transmissão dos dados. Ao contrário do UDP, o objetivo do TCP é proporcionar, às suas aplicações clientes, uma transferência de dados confiável entre dois sistemas terminais. Assim, o TCP aplica mecanismos de controle de transporte a fim de assegurar que os pacotes sejam entregues sem erros, na ordem em que foram enviados, e sem perda ou duplicação de dados. O serviço oferecido pelo TCP, é caracterizado pelas seguintes propriedades:

- orientado a conexão: antes das duas entidades de aplicações iniciarem a transmissão, elas devem estabelecer uma conexão. Esse processo de comunicação entre duas entidades passa por três etapas. Estabelecimento de uma conexão entre os interlocutores que desejam se comunicar; seguido pela própria comunicação de dados; e, finalmente, o fechamento da conexão que indica o fim de comunicação;

- orientado a seqüência (*streams*): os dados da aplicação, camada superior, são enviados continuamente (sem divisão em pacotes). O TCP no emissor, se encarrega da repartição desta cadeia (stream) em "segmentos" assim como da reconstituição dos segmentos na cadeia original pelo receptor;

- *full-Duplex*: a transmissão dos pacotes ocorre simultaneamente em ambos os sentidos, isto é, o circuito virtual é representado por dois canais, um de ida e outro de volta (mais um outro par de canais para a entrega de dados urgentes). Assim, os interlocutores de uma conexão TCP são chamados de emissor/receptor em vez de fonte/destino;

- *graceful Release*: o TCP possui um mecanismo de sincronismo que pode ser usado a fim de garantir que todos os dados sejam enviados sucessivamente antes que a comunicação entre os dois sistemas seja encerrada;

- endereçamento da Camada Aplicação: vários circuitos virtuais podem ser simultaneamente estabelecidos com diferentes aplicações na mesma máquina. Os aplicativos da mesma máquina se comunicam com a camada de transporte através de portas;

- integridade dos Dados do Usuário: é a garantia de que os dados entregues não são corrompidos durante a transmissão. Ao contrário do UDP, a comunicação TCP prevê a recuperação de erros através da retransmissão de dados.

3.1.1 Mecanismos Aplicados pelo TCP

O TCP usa o mecanismo de Janela Deslizante, baseado na técnica de confirmação positiva com retransmissão (*Positive Acknowledgement with Retransmission*) onde a entidade emissora, se não receber a resposta confirmando que os dados chegaram corretamente no sistema remoto, deve enviá-los novamente. O módulo TCP emissor só dispensa os dados da aplicação cliente quando recebe a confirmação do correto recebimento destes dados, o que resolve os problemas de corrupção ou perda de dados. Para resolver a chegada de segmentos duplicados (algumas vezes gerada pela retransmissão de dados que atrasam ao chegar) ou fora da ordem, o módulo TCP receptor mantém um mapa de memória semelhante ao aplicado no caso de fragmentação. Os segmentos são armazenados seguindo um mapa igual ao do emissor e só são entregues à aplicação quando se forma uma seqüência correta de dados. Logo, os dados duplicados servirão para reescrever o segmento na memória e aqueles que chegam fora de ordem esperarão a chegada dos dados atrasados a fim de comporem uma seqüência correta.

Além de resolver estes problemas, o TCP usa o mecanismo da janela deslizante (Figura 3.1) para controlar o fluxo de dados entre os aplicativos (fim-a-fim). Basicamente, este mecanismo possibilita o envio de múltiplos segmentos antes que suas confirmações cheguem de volta, ou seja, em antecipação aos reconhecimentos. Assim, a utilização desta técnica permite um melhor aproveitamento da capacidade do canal, uma vez que são enviados pacotes mesmo durante a espera pela confirmação positiva de dados enviados anteriormente, o que mantém o meio ocupado de modo mais eficiente. A idéia das janelas deslizantes é permitir a transmissão de uma determinada quantia de pacotes sem esperar a confirmação individual. Só quando um ou mais pacotes da janela são reconhecidos que ela desliza abrindo a possibilidade de transmitir o(s) próximo(s) pacote(s). Isto envolve maior rapidez de transmissão e a confirmação múltipla de vários pacotes ao mesmo tempo (economizando mensagens de confirmação).

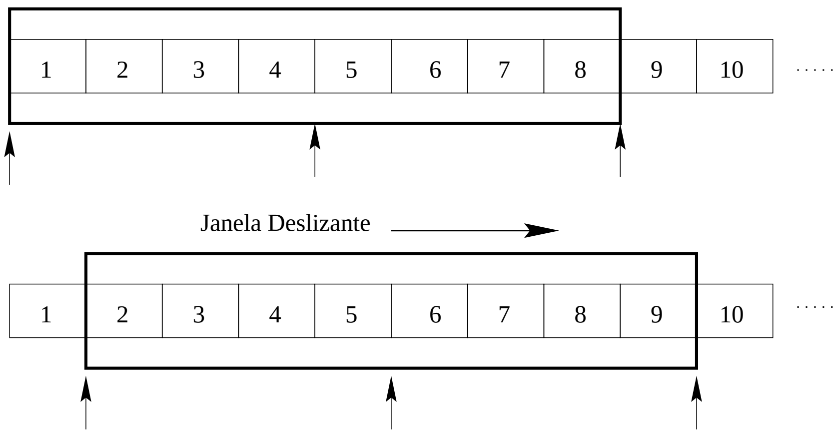


Figura 3.1: Esquema do funcionamento da janela deslizante.

Ao receber os reconhecimentos dos segmentos enviados, a janela desloca-se permitindo o envio de novos dados. No caso da Figura 3.1 poderíamos enviar o pacote de número 9 ao receber o reconhecimento do pacote de número 1, deslizando a janela de transmissão. O emissor mantém três ponteiros associados a cada conexão (Figura 3.1). O primeiro ponteiro está localizado no lado esquerdo da janela separando os pacotes que já foram reconhecidos dos que não foram, o segundo, localizado no meio da janela indica o próximo pacote que pode ser enviado; e o último ponteiro indica o final da janela, o qual separa pacotes que podem ser enviados dos que não podem.

No caso de não poder enviar diversos pacotes em avanço aos reconhecimentos, ou seja, só poder enviar um pacote por vez, a vazão máxima obtida seria a indicada na Equação 3.1, onde MSS (*Maximum Segmentation Size*) representa o tamanho máximo do pacote em bytes e RTT (*Round Trip Time*) representa o tempo que um pacote necessita para ir e voltar da fonte de dados ao destino.

$$Vaz\tilde{a}o_{max} = \frac{MSS * 8}{RTT} bits/s \tag{3.1}$$

Já com a utilização de janelas deslizantes, nota-se que o limite teórico para a vazão pode ser aumentado, ficando dependente do tamanho da janela de transmissão como,

$$Vaz\tilde{a}o_{max} = \frac{tamanho_d a_j anela * 8}{RTT} bits/s, \tag{3.2}$$

onde, o tamanho máximo da janela de transmissão é dado em Bytes.

O valor de $tamanho_d a_j anela$ é determinado de forma dinâmica, sendo fornecido pelo receptor, indicando o tamanho disponível do seu *buffer*. No TCP, a variável *awnd* (de *advertised window*, a janela anunciada pelo receptor) é que representa este valor.

A janela anunciada pelo receptor (*awnd*), visa realizar um controle de fluxo no TCP. Como o protocolo pode ser utilizado em ambientes heterogêneos, onde os sistemas terminais da conexão podem possuir recursos como *buffers* e capacidade de processamento diferentes, podem ocorrer casos onde o emissor envie os segmentos fazendo com que estes cheguem ao receptor numa taxa superior à capacidade de processamento deste último. Neste caso é necessário que o receptor disponha de meios de conter a taxa de envio do emissor, realizando um ajuste de transmissão de modo à evitar o esgotamento de seus *buffers*. Assim, em todos os sinais de reconhecimento, o receptor informa ao transmissor a quantidade de memória livre nos *buffers* de recepção. Desta forma a janela de transmissão fica limitada, no máximo, à quantidade de bytes disponíveis nos *buffers* do receptor.

Para que ocorra uma ocupação eficiente do canal de transmissão, o valor da janela de transmissão deve ser ajustado cuidadosamente. Deve-se evitar uma possível sobrecarga da rede, com o uso de janelas demasiadamente grandes, como também a sub-utilização do meio, através do emprego de janelas pequenas.

3.2 Controle de Congestionamento do TCP

Desde a sua especificação original, no início dos anos 80, o protocolo TCP tem sofrido diversas modificações, gerando melhorias que permitem a sua utilização em ambientes cada vez mais heterogêneos. Dentre as diversas modificações sofridas por este protocolo, a de maior importância foi o acréscimo de mecanismos de controle de congestionamento realizada em 1988 por Van Jacobson e Michael J. Karels [41]. Nesta época, a Internet vinha sofrendo de graves problemas de desempenho, ocorridos devido à ausência de tais mecanismos. Os usuários competiam de maneira extremamente agressiva pelos recursos da rede, gerando instabilidade e colapsos de congestionamento.

Pode-se definir congestionamento como uma situação em que a demanda pelos recursos da rede se aproxima ou ultrapassa a sua capacidade [64]. A função básica do controle de congestionamento é a alocação de recursos de modo que a rede opere da forma mais eficiente possível nessas situações. Entre os recursos de uma rede, podem ser citados: a largura de banda dos enlaces, o espaço em *buffers* e o tempo de processamento nos nós intermediários da rede.

O controle de congestionamento do TCP tem como princípio básico o fato de que o fluxo de pacotes na rede deve ser conservativo. Isto implica em que um novo pacote só pode ser colocado na rede depois que um pacote antigo já tenha sido retirado [41]. O recebimento de um ACK (confirmação do recebimento de um ou mais segmentos já enviados) é o que indica ao transmissor, que um pacote deixou a rede. Como o transmissor só pode enviar mais segmentos ao receber ACKs, o protocolo ajusta automaticamente sua taxa de transmissão ao enlace mais lento da rede. Essa característica é essencial para lidar com a heterogenidade da Internet. Assim, o controle de congestionamento realizado pelo TCP envolve detectar e consertar pontos que violam a conservação de pacotes, sendo o tamanho da janela deslizante do emissor dinamicamente ajustada na tentativa de alcançar essa conservação.

O protocolo TCP detecta a ocorrência de um congestionamento através do parâmetro perda de pacotes ao qual o transmissor tem acesso. Como as redes atuais apresentam baixas taxas de erro binário, pode-se supor com segurança que quando ocorre uma perda, ela é causada por congestionamento na rede. O principal indicador que o TCP usa para inferir que houve uma perda é o tempo entre a transmissão de um pacote e a chegada do reconhecimento relativo àquele pacote, o que corresponde ao RTT (*Round Trip Time*) da conexão. Como o tempo necessário para o pacote chegar ao receptor não é conhecido a priori, ele tem de ser estimado no início e no decorrer da conexão. Uma conexão TCP pode envolver desde sistemas terminais em uma mesma rede local ou separados por enlaces de satélite, portanto a estimativa tem de abranger uma grande faixa de valores mantendo uma boa precisão.

Desta forma, um aspecto importante do controle de congestionamento do TCP é a estimativa do RTT, que adquire importância na medida em que a partir dela se estabelece o valor do temporizador que determina as retransmissões de dados. A precisão da estimativa do RTT é importante porque se ele for subestimado, o TCP fará retransmissões desnecessárias. Se por outro lado ele for superestimado, ele levará muito tempo para realizar retransmissões, fazendo com que o protocolo perca em eficiência. Como já definido anteriormente, o RTT é o tempo que um pacote necessita para ir da fonte de dados ao destino e voltar. Esta variável depende do tempo de propagação do sinal no meio de transmissão e do tempo gasto pelos dados nas filas dos roteadores. Desta forma, o RTT varia de acordo com a carga a qual a rede está sendo submetida. Quanto maior o tráfego na rede, maiores serão as filas nos roteadores e maiores serão os tempos de espera. Em geral, para redes de pacotes, a medida do RTT possui altos desvios padrões. Estas grandes variações nos valores do RTT tornam a estimação acurada do temporizador de retransmissão uma tarefa não trivial. A estimativa do RTT é um fator importante no cálculo da temporização, sendo enfaticamente abordado em [41].

Uma considerável quantidade de esforços foi aplicada no desenvolvimento e de algoritmos que procurassem conservar uma situação estável por parte da rede e realizassem o controle de congestionamento do TCP. Esses algoritmos encontram-se formalmente definidos em [41, 43, 44] e correspondem a: *Slow Start, Congestion Avoidance, Fast Retransmit* e *Fast Recovery*. Apesar de serem independentes, esses algoritmos são geralmente implementados de forma conjunta. Os mecanismos de controle de congestionamento empregados pelo TCP também foram avaliados em [65].

3.2.1 *Slow Start*

As implementações mais antigas do TCP iniciavam suas conexões enviando uma rajada de segmentos do tamanho da janela anunciada pelo receptor. A perda de pacotes era percebida pelo TCP através do estouro de uma temporização, que ativava a verificação do último segmento reconhecido pelo receptor. O protocolo enviava outra rajada, partindo do segmento seguinte a ser enviado. Assim, o transmissor continuava a colocar rajadas de pacotes em uma rede que já estava congestionada, piorando a situação. Este comportamento, insensível às condições da rede, tendia a causar sobrecarga nas filas dos roteadores, que com isso perdiam pacotes. Além disso, tornava as redes altamente instáveis, pois não havia um mecanismo que limitasse o envio de pacotes pelo transmissor em função destas temporizações. Jacobson et al. [41] mostrou como essa simples abordagem pode reduzir drasticamente a vazão do TCP.

Para evitar essas situações de instabilidade, deve existir algum mecanismo que controle o comportamento transitório da conexão levando-a ao ponto de equilíbrio de forma rápida, que entretanto não cause congestionamento na rede. A solução proposta [41, 43] para este problema foi iniciar-se a conexão com uma janela pequena, aumentando-a de forma gradual até que se atinja a capacidade da rede. Isto se aplica tanto a uma conexão que se inicia como ao reinício após um estouro do temporizador, quando o congestionamento já existe e se deseja esvaziar o meio e começar novamente a manipulação da janela de transmissão.

A implementação desta solução é feita da seguinte forma:

1. adiciona-se a variável *congestion window (cwnd)*;
2. no início ou reinício (após a ocorrência de perda) da conexão, *cwnd* recebe o valor 1*MSS;
3. a cada novo ACK (não duplicado) recebido pelo transmissor, *cwnd* é incrementada de 1*MSS;
4. o tamanho da janela de transmissão é configurado para o menor valor entre a janela anunciada pelo receptor (*awnd*[1]) e *cwnd* ([2])

O procedimento de aumentar *cwnd* de um pacote para cada ACK recebido resulta em um crescimento exponencial da janela. Assim, no início, quando um ACK é recebido, a janela de transmissão é incrementada de um para dois. Quando cada um desses dois pacotes é reconhecido, a janela é aumentada para quatro. Isso faz com que o tamanho da janela aumente exponencialmente. Porém, vale ressaltar que devido ao fato do receptor poder atrasar o envio de ACKs e enviar um ACK para cada dois pacotes que ele recebe, esse crescimento pode não ser exatamente exponencial.

Agindo dessa forma, em um dado momento se atinge a capacidade da rede, pois rajadas cada vez maiores exigem espaço nas filas dos roteadores. Do ponto de vista do *slow-start*, a única ação a ser tomada com relação a isto é que na ocorrência de uma temporização, o valor de *cwnd* volta a 1*MSS e o algoritmo é reiniciado. Outros algoritmos descritos a seguir, tratam da adaptação do transmissor às limitações de banda-passante na rede. O objetivo do *slow start* é permitir que uma conexão iniciada ou reiniciada após uma temporização atinja rapidamente o seu ponto de equilíbrio.

[1] A *advertised window* corresponde ao controle de fluxo imposto pelo receptor

[2] A *congestion window* corresponde ao controle de fluxo imposto pelo emissor

3.2.2 Congestion Avoidance

Conforme foi dito na seção anterior, o *slow start* incrementa a janela de transmissão exponencialmente. Em um determinado ponto, a capacidade da rede será ultrapassada, provocando perdas de pacotes. Isto indica que a janela de congestionamento ultrapassou o ponto de equilíbrio. A partir daí, o protocolo passa a buscar novamente uma condição de equilíbrio. O algoritmo que implementa esta etapa do controle de congestionamento chama-se *congestion avoidance.*

O *congestion avoidance* manipula a janela de congestionamento seguindo a regra de aumento aditivo e diminuição multiplicativa (*additive increase–multiplicative decrease*). Em situações de congestionamento (sinalizadas por perda de pacotes) a janela é reduzida de forma exponencial. Do contrário, a janela aumenta linearmente.

Isto se justifica pelo seguinte: pode-se demonstrar que quando a rede está congestionada, as filas nos roteadores aumentam exponencialmente. Portanto, a redução na taxa de envio também deve ser exponencial de modo a permitir que as filas se estabilizem. Por outro lado, a rede não indica ao transmissor que ele está sub-utilizando a banda passante disponível, logo deve haver uma maneira de se aumentar a taxa de transmissão, de modo que se possa descobrir o limite atual de banda passante. Um aumento exponencial, por mais lenta que fosse a constante de tempo, levaria a janela a aumentar muito rapidamente, causando instabilidade na rede.

A taxa adotada para a diminuição multiplicativa é de 0,5. Este número foi escolhido de acordo com o seguinte raciocínio: quando se verifica uma situação de congestionamento, o transmissor pode estar iniciando a conexão (em *slow start*), ou estar no estado estacionário. No primeiro caso, em que um pacote é perdido no *slow start*, a redução da janela à metade significa a volta da janela para o tamanho anterior, que sabemos que não provocou perdas. A partir deste ponto o aumento da janela passa a ser mais lento, buscando o equilíbrio. Na situação de uma conexão em estado estacionário, a perda de um pacote indica que a banda passante disponível para a conexão diminuiu, provavelmente porque outra conexão foi iniciada e está disputando os recursos da rede. Supondo que a primeira conexão estivesse sozinha e que agora divida a banda com uma outra, é correto reduzir a janela à metade pois a banda passante foi reduzida nesta proporção. Este é o pior caso no estado estacionário, já que se houvesse duas conexões e uma terceira fosse iniciada, a redução de banda seria de 1/3, e assim por diante. A redução pela metade é mais conservadora nestes casos, ajudando a garantir a estabilidade da rede.

O incremento do processo de aumento aditivo (*additive increase*) é de $1/cwnd$, para cada ACK recebido. Isto se traduz em um incremento de no máximo 1 pacote por RTT. Como o TCP é orientado a bytes, e não a pacotes, este incremento será de $MSS * MSS/cwnd$, que incrementará a janela de no máximo MSS a cada RTT.

Para compreender este fato deve-se inicialmente notar que uma janela tem n segmentos, sendo $n = cwnd/MSS$. O valor novo de $cwnd$ será então $cwnd_{nova} = cwnd + A * MSS * MSS/cwnd$, onde A é o número de ACKs recebidos. Se toda a janela de dados enviada for recebida com sucesso, então o valor de A será igual ao mesmo valor de n que é o número de segmentos enviados na janela. Assim,

$$cwnd_{novo} = cwnd + n * \frac{MSS * MSS}{c} wnd = cwnd + \frac{cwnd}{MSS} * \frac{MSS * MSS}{cwnd} \tag{3.3}$$

e, portanto,

$$cwnd_{novo} = cwnd + MSS. \tag{3.4}$$

É importante observar que esta é, na realidade, uma aproximação, uma vez que os ACKs não são recebidos no mesmo instante. Desta forma a janela $cwnd$ aumenta gradualmente com o recebimento dos reconhecimentos, com uma taxa de aproximadamente 1 segmento por janela de ACKs recebida.

O algoritmo de *congestion avoidance* então é realizado da seguinte forma:

1. quando ocorrer um estouro de temporização (*time-out*), $cwnd$ recebe $cwnd/2$;
2. para cada novo ACK recebido (não duplicado), incrementa-se $cwnd$ de $1/cwnd$;

3. a janela de transmissão é a menor entre a janela do receptor (*awnd*) e *cwnd*.

Na prática, o *slow start* e o *congestion avoidance* são implementados em conjunto. A fusão dos dois algoritmos é bastante simples. Além da variável *cwnd*, que os dois usam, adiciona-se a variável *ssthresh* (*slow start threshold*, o limite do *slow start*), que determina qual dos dois estará ativo.

A implementação em conjunto fica:

1. quando ocorrer um estouro de temporização, *ssthresh* recebe $cwnd/2$;

2. se $cwnd < ssthresh$, a janela é manipulada pelo *slow start*;

3. se $cwnd \geq ssthresh$, a janela é manipulada pelo *congestion avoidance*.

Assim, o *slow start* abre a janela exponencialmente até que haja um estouro de temporização. Quando isto ocorre, a janela volta para 1 e aumenta exponencialmente até metade da janela que provocou a perda. A partir daí, o *congestion avoidance* assume o controle, aumentando a janela lentamente de modo a conseguir mais banda passante disponível na rede.

A Figura 3.2 ilustra o comportamento dos dois mecanismos operando em conjunto [66, 67]. Inicialmente o mecanismo de *slow start* faz com que a *cwnd* cresça de forma exponencial até a ocorrência de uma perda. Nesse momento o TCP nota que a janela ultrapassou a capacidade da rede. Assim a variável *ssthresh* recebe metade do valor da janela no instante da detecção da perda. Em seguida a janela é reinicializada em 1 segmento e seu crescimento, até que se atinja o limiar de *sstresh*, se dá de forma exponencial de acordo com o *slow start*. Quando o tamanho de *cwnd* atinge o valor de *sstresh* o protocolo passa a operar em *congestion avoidance*, realizando um aumento linear da janela. Se uma perda for notada já na fase de *congestion avoidance*, o comportamento do TCP será o mesmo.

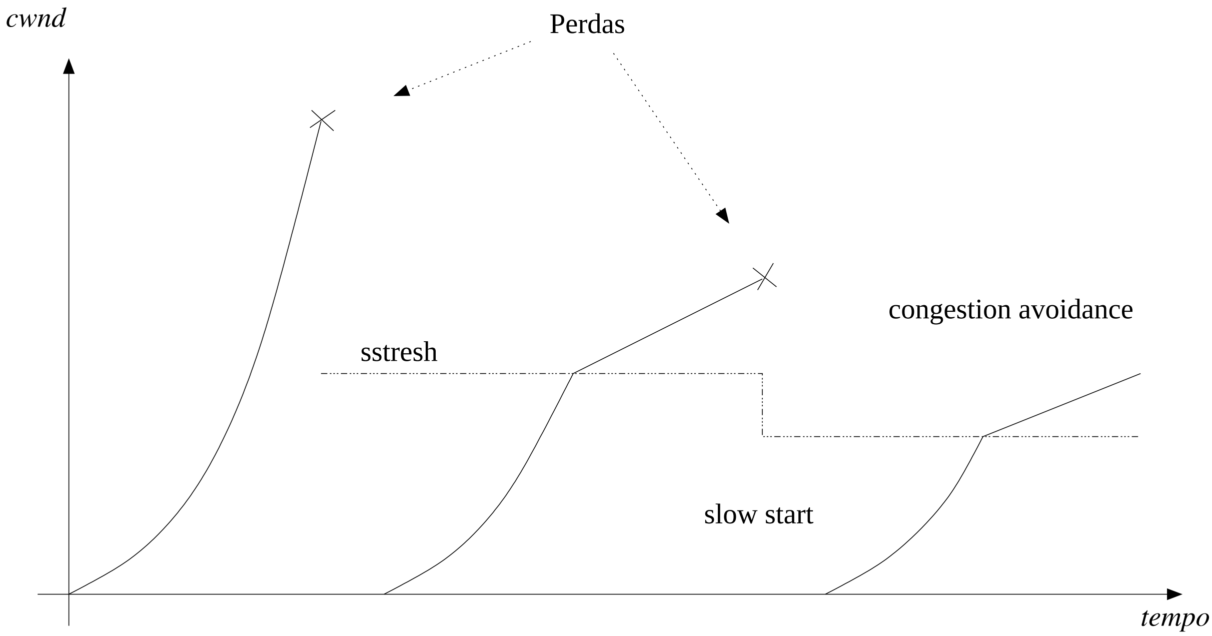


Figura 3.2: Mecanismos de *slow-start* e *congestion avoidance* atuando em conjunto.

3.2.3 Fast Retransmit

O mecanismo *fast retransmit* visa apressar a retransmissão de segmentos quando a rede se encontra em uma situação de congestionamento moderado.

É importante lembrar que um agente TCP deve gerar um ACK (chamado ACK duplicado) ao receber um segmento fora de ordem. Normalmente, ACK(n) tem a conotação de que todos os segmentos até $n - 1$ foram recebidos, e aguarda-se a transmissão de n. Se o transmissor já recebeu ACK(n) e recebe outro, isto significa que um segmento com número maior que n foi recebido, mas n não. Estes ACKs duplicados informam então ao transmissor que um segmento foi recebido fora de ordem e qual é o número de seqüência esperado.

Como a rede IP é sem conexão, para cada pacote se faz o cálculo da melhor rota e é possível que alguns pacotes cheguem fora da ordem em que foram enviados. Logo, o TCP não sabe se o ACK 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 ACKs duplicados até que o segmento fora de ordem seja processado e um novo ACK seja gerado. Um número de ACKs replicados maior do que o limite de três é interpretado pelo TCP como um indicativo de perda de pacote, e o segmento é retransmitido. Isto evita que se tenha que esperar o estouro do temporizador, e a retransmissão é feita mais rapidamente.

3.2.4 Fast Recovery

O *fast recovery* foi uma modificação ao algoritmo anterior. Seu objetivo é melhorar a eficiência de utilização do meio, evitando que o canal se esvazie em decorrência de uma perda isolada.

Ressalta-se que o controle de congestionamento do TCP usa como indicador de congestionamento na rede uma perda de pacote. A perda de pacote, por sua vez, pode ser detectada de duas formas: por um estouro de temporização ou pela recepção de um determinado número de ACKs replicados. Neste último caso, além de indicar a perda de um pacote, os ACKs duplicados carregam outra informação: que pacotes válidos deixaram a rede. Como o controle de congestionamento do TCP é baseado no princípio da conservação de pacotes na rede, o fato de um pacote ter sido retirado implica que se pode transmitir outro pacote pela rede.

Mais que isso, os ACKs replicados indicam que mesmo que um pacote tenha sido perdido, existe um fluxo de dados entre os dois sistemas terminais da conexão. Isto significa que o congestionamento que existe é moderado, e portanto não é necessário realizar novamente o *slow start* para esvaziar o canal.

O *fast recovery* é implementado em conjunto com o *fast retransmit*, da seguinte forma:

1. ao receber três ACKs replicados consecutivos, $ssthresh$ recebe $cwnd/2$;
2. retransmite-se o segmento supostamente perdido;
3. $cwnd$ recebe $ssthresh + 3 * MSS$;
4. a cada novo ACK duplicado, incrementa-se $cwnd$ de $1 * MSS$ e se for possível, transmite-se um novo segmento;

5. ao chegar um novo ACK, *cwnd* recebe *ssthresh*.

Deve-se notar que o primeiro e o último passo do algoritmo acima correspondem ao *congestion avoidance*, ou seja, a janela é reduzida à metade. No entanto, para evitar o *slow start*, o *fast recovery* "infla" a janela de congestionamento, incrementando-a de um MSS para cada ACK replicado recebido (passos 3 e 4). Deste modo, para cada pacote que deixa a rede, um novo é inserido. No passo 3, *cwnd* é incrementada de 3*MSS de uma vez por causa dos três ACKs duplicados que são necessários para disparar o *fast recovery*, indicando que três pacotes já deixaram a rede. O *fast retransmit* está implementado no passo 2.

Com a evolução dos algoritmos de controle de congestionamento descritos anteriormente, diferentes versões do protocolo TCP foram sendo implementadas. De acordo com os novos mecanismos propostos, o *software* do TCP veio sofrendo atualizações de modo a incorporar as modificações sugeridas. As versões mais conhecidas do protocolo são: TCP Tahoe, TCP Reno, TCP New Reno e TCP SACK. Uma comparação dessas versões foi apresentada por K. Fall e S. Floyd [42]. O TCP Reno é a versão do protocolo TCP que encontra-se atualmente mais difundia na Internet, sendo a versão empregada nas simulações analisadas neste trabalho.

3.3 O Desempenho do Comportamento do TCP

O dinamismo do tráfego na Internet é fortemente influenciado pela conduta do algoritmo de *congestion avoidance* do TCP [41, 43]. Motivadas por esse fato, pesquisas foram desenvolvidas a fim de investigar um modelo de desempenho analítico para esse algoritmo [44]. No modelo obtido, a vazão do TCP durante a execução do algoritmo de *congestion avoidance*, foi estimada por

$$BW = \frac{MSS * C}{RTT * \sqrt{p}}, \tag{3.5}$$

onde MSS (*Maximum Segment Size*) é o tamanho máximo do segmento em bytes, p é a taxa de perda média que a conexão sofre e RTT é o *Round-Trip Delay* no caminho da conexão. Como já descrito anteriormente, o RTT é composto pelo atraso de propagação dos enlaces (τ) e pelo atraso médio ocorrido nas filas ($\overline{d}$)durante o caminho. A constante de proporcionalidade C constitui um conjunto de termos que são tipicamente constantes para uma combinação de: implementações do TCP, estratégia de ACK (atrasados ou não) e mecanismo de perda (periódicas ou aleatórias).

A estimativa do desempenho do TCP foi possível mediante algumas simplificações, onde assumiu-se que:

- o caminho percorrido pelas conexões TCP tem perdas, onde o RTT é constante, pois possui largura de banda suficiente e carga total baixa o suficiente para nunca encher quaisquer filas;
- o enlace libera aproximadamente $1/p$ pacotes consecutivos, seguido por um descarte.

Outra forma de se estimar o desempenho do TCP foi apresentada por Floyd [68]. De acordo com Mathis [44], esse modelo pode ser usado para prever como o TCP compartilha a largura de banda na Internet. Ele pode também ser usado para prever os efeitos do TCP na Internet e representar um processo de equilíbrio entre perda, atraso e largura de banda.

Analisando a Equação 3.5 e sendo $RTT = \tau + \overline{d}$ pode-se facilmente verificar que a vazão do protocolo TCP é inversamente proporcional às perdas e atrasos médios das filas, ocorridos no caminho da conexão. Desta forma, constitui objeto de estudo deste trabalho a análise do comportamento de um modelo de diferenciação proporcional de atraso e taxa de perda, quando submetido ao tráfego TCP, cujo comportamento é fortemente influenciado por tais métricas de desempenho. Pretende-se avaliar se as proporções de atraso médio e taxa de perda estabelecidos entre as classes de serviço oferecidas, são garantidas quando o tráfego empregado reage à diferenciação proporcional realizada.

Capítulo 4

Diferenciação Proporcional com Tráfego TCP

A FIM de analisar o desempenho de um modelo de diferenciação proporcional utilizando tráfego TCP, diversas simulações foram realizadas. O comportamento desse modelo é avaliado para verificar se as proporções estabelecidas entre as classes de serviço, para atraso nas filas e taxa de perda de pacotes, são garantidas quando aplicado um tráfego cujo comportamento é fortemente influenciado por tais métricas de desempenho, como é o caso do protocolo TCP. A seção 4.1 descreve a implementação da arquitetura de Diferenciação Proporcional de Serviços na ferramenta de simulação. A seção 4.2 apresenta os resultados das simulações com fontes de tráfego UDP, realizadas com o objetivo de validar a arquitetura desenvolvida. As simulações com fontes de tráfego TCP, juntamente com os resultados obtidos e suas análises são discutidos na seção 4.3.

4.1 Infra-estrutura de Simulação

Para a realização das simulações propostas foi utilizada a ferramenta *NS-2* [69]. O *NS-2* é um simulador dirigido a eventos discretos desenvolvido para a pesquisa em redes de comunicação. Ele realiza uma abstração de nós e enlaces para a simulação de protocolos de rede. Sobre esta infra-estrutura, alguns algoritmos de roteamento utilizados na Internet são implementados. O *NS-2* implementa diferentes variações de protocolos da arquitetura TCP/IP, como o TCP e o UDP, e também fontes de tráfego, tais como CBR e as modeladas por distribuições *On-Off* e Pareto. Além disso, ele permite o acréscimo de novos módulos a sua infra-estrutura, possibilitando o desenvolvimento de extensões para áreas específicas. Esses módulos podem representar um protocolo, uma aplicação ou uma fonte de tráfego.

Uma extensão ao pacote básico do *NS-2* foi implementada para o suporte à simulação de Serviços Diferenciados. Essa extensão, na realidade, engloba a criação de diversos módulos independentes que implementam os mecanismos necessários à arquitetura de Diferenciação de Serviços (DS). Tal extensão permanece em desenvolvimento no laboratório do Grupo de Teleinformática e Automação (GTA) da COPPE/UFRJ, visando o acréscimo de novos mecanismos, e tem sido objeto de várias pesquisas nesta área. A implementação de diferentes classes de serviço no esquema de Diferenciação Absoluta de Serviços, é realizada através da combinação de uma diversidade de mecanismos de classificação, escalonamento de pacotes e condicionamento de tráfego. Assim sendo, essa extensão ao *NS-2* desenvolvida inclui mecanismos de:

- classificação de pacotes, tanto classificadores MF quanto BA (seção 2.1);
- escalonamento de pacotes, tais como filas com prioridade (PRR), *weighted round-robin* (WRR) e *weighted fair-queueing* (WFQ);
- condicionamento de tráfego, tais como suavizadores de tráfego (*shapers*) e policiadores;
- marcação de pacotes de acordo com perfis de tráfego (*profilers*);
- gerenciamento ativo de filas, tais como o RED genérico.

De forma semelhante às extensões descritas anteriormente, novas implementações foram geradas para permitir a realização de uma Diferenciação Relativa de Serviço, o que resultaram em aproximadamente três mil linhas de códigos. Com essa finalidade, foram acrescentados à arquitetura de Diferenciação de Serviços já implementada, os módulos constituintes do Esquema de Encaminhamento Proporcional da Figura 2.4, (seção 2.4.1). A implementação do Modelo Proporcional para a Diferenciação Relativa de Serviços é realizada através da combinação de mecanismos de gerenciamento de *buffer*, de encaminhamento de pacotes e descarte proporcionais. Assim sendo, os algoritmos escolhidos para desempenhar o papel desses mecanismos foram:

- O escalonador proporcional de atraso WTP (seção 2.4.1). Escolhido devido à simplicidade envolvida em sua implementação, estando de acordo com o proposto por Dovrolis et.al em [37].
- O descartador proporcional de taxa de perda PLR(∞) (seção 2.4.1). Também escolhido devido a sua simplicidade, estando de acordo com o proposto por Dovrolis et.al em [38].
- O gerenciador de *buffer* *DropTail-CS* (seção 2.4.1). Sua implementação consistiu em alterações no algoritmo *DropTail* original do *NS-2*, para que passasse a empregar a política de organização de *buffer* de compartilhamento completo. Para tanto, o tamanho máximo de cada fila é configurado para o espaço total disponível no *buffer*, compartilhado por todas as classes. A cada chegada de pacote, a somatória dos espaços ocupados por cada fila não ultrapassa o tamanho total do *buffer*. Caso isso ocorra, uma sinalização de necessidade de descarte é emitida para o descartador PLR(∞).

Todos estes mecanismos foram desenvolvidos e implementados em uma arquitetura de Diferenciação de Serviços de forma genérica, para que ela possa ser usada em diferentes configurações e topologias de rede. A combinação adequada desses mecanismos, com alguns módulos anteriormente implementados para a arquitetura DS, e com os já existentes no original *NS-2*, permite a simulação de um ambiente de Diferenciação Relativa de Serviços capaz de suportar o transporte de qualquer tipo de tráfego, incluindo TCP. Antes da avaliação do comportamento desse ambiente, quando sujeito ao tráfego TCP, foi necessária a validação do esquema de encaminhamento proporcional implementado. A próxima seção discute as simulações realizadas com esse objetivo.

4.2 Validação do Esquema Implementado

As etapas de validação consistiram na realização de algumas simulações descritas por Dovrolis et.al. em [37, 38] e na comparação dos resultados obtidos com os apresentados nessas citações. Assim, as etapas executadas objetivaram:

- primeiramente, validar isoladamente o funcionamento do mecanismo de escalonamento empregado pelo WTP em um ambiente sem perdas (*buffer* grande), aplicando-se tanto fontes com taxa de transmissão constante (*Constant Bit Rate* - CBR) quanto fontes do tipo atividade-silêncio com distribuição *Pareto* que apresentam dependência de longa duração;
- em seguida, validar o funcionamento do mecanismo de descarte empregado pelo PLR(∞) em alta carga, também aplicando-se tanto fontes CBR quanto fontes Pareto;
- por fim, o esquema de encaminhamento proporcional foi completamente validado em condições variadas, onde foi aplicado unicamente fontes de tráfego Pareto.

Em todas as simulações, os resultados obtidos com tráfego Pareto mostraram-se equivalentes aos originalmente apresentados nas citações. Simulações com fontes de tráfego CBR não constavam nessas citações e foram realizadas objetivando um melhor aprendizado e a obtenção de um maior conhecimento sobre o comportamento dos mecanismos empregados.

Todas as etapas de validação apresentadas nesta seção utilizam a topologia apresentada na Figura 4.1. Nessa topologia, todos os nós são interligados por enlaces de 10Mbps e 1ms de atraso de propagação, com exceção do enlace de gargalo entre os nós r_1 e r_2 com capacidade de 5Mbps e atraso de propagação de 10ms, o que resulta em um atraso de propagação ida e volta (τ) de 22ms. Os nós s_1 e s_2 possuem fontes responsáveis pela geração de tráfego de duas classes de serviço, as quais devem ser tratadas diferenciadamente no enlace de gargalo. Os pacotes gerados por esses nós são enviados para o nó de destino r_2.

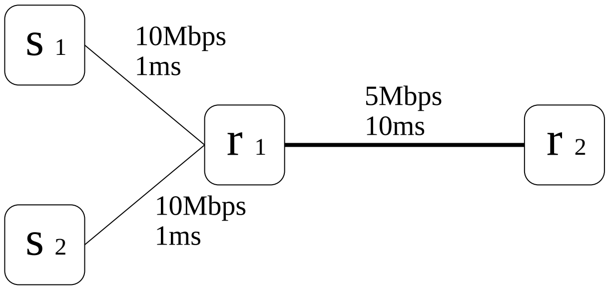


Figura 4.1: Topologia de simulação empregada nas validações.

O enlace de gargalo é compartilhado por pacotes pertencentes a duas classes diferentes e armazenados em duas filas lógicas. A primeira fila recebe tráfego gerado pelo nó s_1 e corresponde à classe menos prioritária (classe 1). A segunda recebe tráfego gerado pelo nó s_2 e corresponde à classe mais prioritária (classe 2).

4.2.1 Validação do Escalonador WTP sem a Presença de Perdas

Com o objetivo de validar o algoritmo de escalonamento empregado pelo WTP, simulações foram realizadas empregando-se a topologia da Figura 4.1. Nessas simulações, procurou-se verificar se o WTP aproxima satisfatoriamente do modelo proporcional de atraso descrito na Seção 2.3.1. Nesta etapa de validação, não foi considerada a diferenciação de perda, sendo a avaliação do WTP baseada em um modelo sem perda, com *buffers* grandes. Na Seção 4.2.3, este mesmo escalonador será validado em um ambiente com perdas. Assim sendo, nas duas filas lógicas que compartilham o enlace de gargalo (correspondentes às classes 1 e 2), foi empregado o gerenciador de *buffer DropTail*, sendo cada fila configurada para um tamanho de *buffer* (B) igual a 4000 pacotes.

Ambas as filas lógicas são servidas pelo escalonador de pacotes proporcional WTP, onde os Parâmetros de Diferenciação do Escalonador (*Scheduler Differentiation Parameters* - SDP) são configurados com $sdp_1 = 1$ e $sdp_2 = 2$, para a classe 1 e 2 respectivamente, sendo $sdp_1/sdp_2 = SDP$. Assim, a prioridade no escalonamento de pacotes da classe 2 aumenta duas vezes mais rápido com o tempo de espera em fila desses pacotes, resultando em atrasos duas vezes menores. Tudo isso, em relação aos pacotes da classe 1.

Tráfego Monitorado

Fontes de tráfego CBR e Pareto foram empregadas nessa etapa de validação, configuradas como descrito a seguir. As fontes Pareto quando estão no estado de atividade (*burst*), geram um tráfego a uma taxa de pico de 200 kbps e são configuradas com um parâmetro de formato $\alpha = 1.9$. Os períodos de atividade são determinados por uma variável aleatória com distribuição Pareto de média 600 ms. Os períodos de silêncio (*idle*) também seguem uma distribuição Pareto de média 400 ms. Estes valores resultam em uma taxa média para o tráfego Pareto de 120 kbps. As fontes CBR utilizadas possuem taxas de 200 kbps. Em todos os fluxos, CBR ou Pareto, os pacotes utilizados possuem tamanho de 1000 octetos.

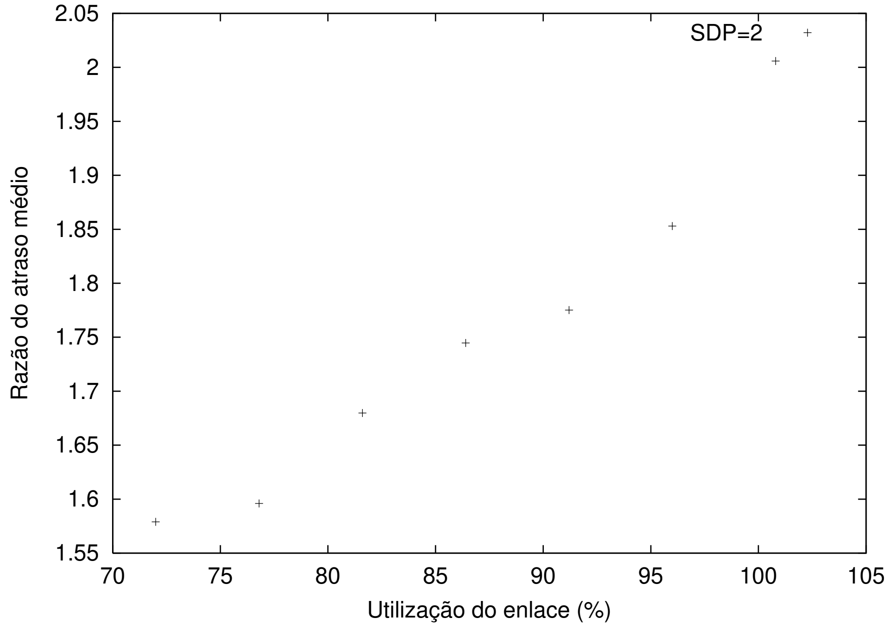
Com o objetivo de avaliar o efeito da carga agregada no desempenho do escalonador WTP, foi variada a utilização (u) do enlace de gargalo de aproximadamente 70% (carga moderada) a 100% (alta carga). Tal variação na carga foi realizada aumentando-se o número de fontes nos nós s_1 e s_2. Se a utilização do enlace é menor do que 70% os atrasos nas filas são pequenos e não afetam a qualidade de nenhum tráfego existente, seja mais ou menos prioritário. Assim sendo, não é necessária a realização de uma diferenciação de serviço a fim de priorizar o serviço a um determinado tráfego.

Resultados

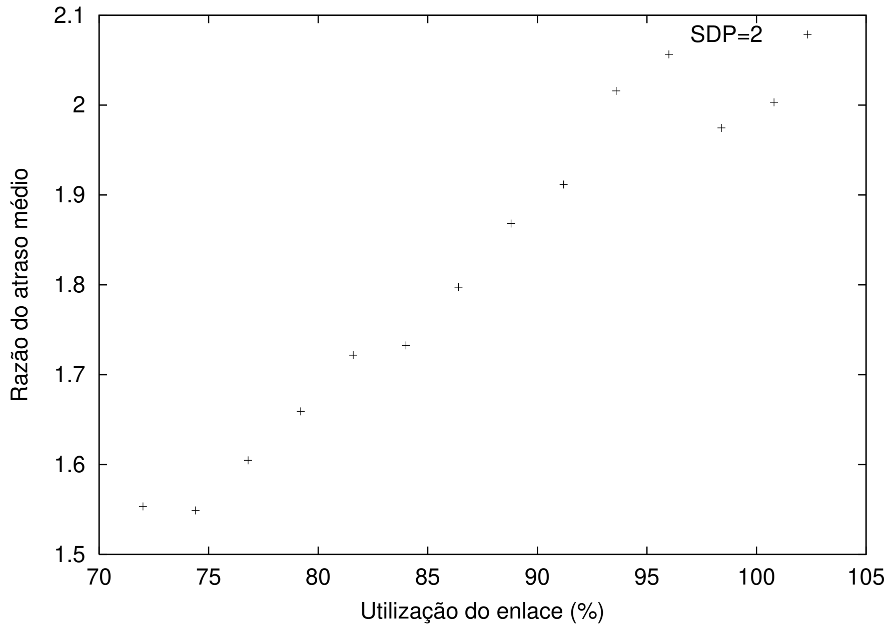
A Figura 4.2 exibe a razão entre os atrasos médios ocorridos nas filas das classes 1 e 2, em condições de carga moderada e alta. Os atrasos médios de cada classe são calculados ao término da simulação e constituem atrasos médios de longa duração. Cada ponto nos gráficos apresentados, resultaram da média entre 10 amostras de simulação com diferentes sementes (*seeds*), sendo o tempo de simulação de cada amostra de 1000 unidades de tempo. Em todos os gráficos, os valores de SDPs escolhidos são representados na forma $sdp_2/sdp_1 = SDP$, onde $SDP = 2$.

Pode-se notar que em ambos os gráficos, com o aumento da carga agregada no enlace de gargalo, o WTP tende a se aproximar mais da diferenciação proporcional de atraso médio estabelecida ($sdp_2/sdp_1 = SDP = 2$). Porém, para cargas moderadas, o escalonador não mantém a diferenciação proporcional de atraso desejada. A ocorrência desse fato está relacionada à factibilidade da diferenciação proporcional de atraso. Assim, este escalonador tende para uma diferenciação proporcional de atraso somente sob condições de carga suficientemente alta.

Os resultados obtidos nessa etapa de simulação foram equivalentes aos apresentados por Dovrolis et.al. em [37]. O escalonador WTP apresentou comportamento equivalente ao analisado na citação, tanto para carga moderada quanto para alta carga. Desta forma, considerou-se a implementação desse escalonador validada e pronta para desempenhar sua função como componente do esquema de encaminhamento proporcional a ser aplicado ao tráfego TCP.



(a) Fontes CBR



(b) Fontes Pareto

Figura 4.2: Razão do atraso médio entre as classes 1 e 2.

4.2.2 Validação do Descartador PLR(∞) em Alta Carga

Com a finalidade de validar o mecanismo de descarte empregado pelo PLR(∞), simulações foram realizadas empregando-se a topologia da Figura 4.1. Nessas simulações, procurou-se verificar se o PLR(∞) aproxima satisfatoriamente do modelo proporcional de taxa de perda descrito na Seção 2.3.1. Os resultados obtidos representam o desempenho do PLR(∞) em alta carga. Na Seção 4.2.3 este mesmo descartador será validado em um ambiente com carga moderada e alta.

O modelo simulado nessa etapa de validação corresponde ao esquema de encaminhamento proporcional da Figura 2.4. Nesse esquema, o escalonador WTP serve as duas filas lógicas que compartilham o enlace de gargalo da Figura 4.1 (correspondentes às classes 1 e 2). Os Parâmetros de Diferenciação do Escalonador (*Scheduler Differentiation Parameters* - SDP) são configurados para $sdp_1 = 1$ e $sdp_2 = 2$, para a classe 1 e 2 respectivamente.

Nas filas, a política de compartilhamento completo *DropTail-CS* é aplicada como política de organização de *buffer* (B). Essa política estabelece que pacotes de ambas as filas compartilham o espaço total de B. O tamanho de B foi selecionado de forma que a taxa de perda gerada ficasse entre 0.5% e 2.5%.

O descartador PLR(∞) provê a diferenciação de taxa de perda e, quando sinalizado pelo algoritmo *DropTail-CS*, é responsável por selecionar a classe que terá um pacote descartado. Os Parâmetros de Diferenciação de Taxa de Perda (*Loss rate Differentiation Parameters* - LDP) foram configurados para $ldp_1 = 2$ e 4 e $ldp_2 = 1$, para a classe 1 e 2 respectivamente. Assim, a classe 1 terá 2 e 4 vezes mais perdas do que a classe 2. Por efeito de simplicidade, nos gráficos adotou-se a representação $SDP = sdp_2/sdp_1$ para o SDP empregado em cada simulação. Para todos os casos analisados, classes superiores terão desempenho melhor em ambas as métricas, de atraso e taxa de perda. Não foram levadas em consideração outras combinações destas métricas, como por exemplo, casos em que uma classe possui atrasos mais altos e taxas de perda menores.

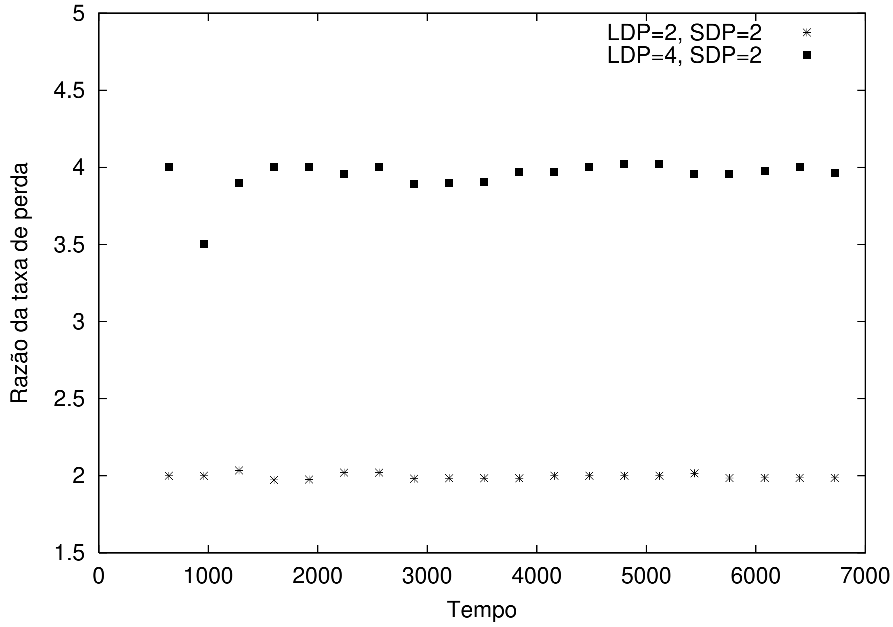
Tráfego Monitorado

Fontes de tráfego CBR e Pareto foram empregadas nessa etapa de validação, configuradas como descrito a seguir. As fontes Pareto quando estão no estado de atividade (*burst*), geram um tráfego a uma taxa de pico de 4 Mbps e são configuradas com um parâmetro de formato $\alpha = 1.5$. Os períodos de atividade são determinados por uma variável aleatória com distribuição Pareto de média 600 ms. Os períodos de silêncio (*idle*) também seguem uma distribuição Pareto de média 400 ms. Estes valores resultam em uma taxa média para o tráfego Pareto de 2.4 Mbps. As fontes CBR utilizadas possuem taxas de 2.5 Mbps. Com essas taxas configuradas, a carga agregada no enlace de gargalo, permanece em $u = 96\%$ para fontes Pareto e $u = 100\%$ para fontes CBR. A carga oferecida foi mantida alta para que fossem geradas perdas suficientes à realização da diferenciação de taxa de perda

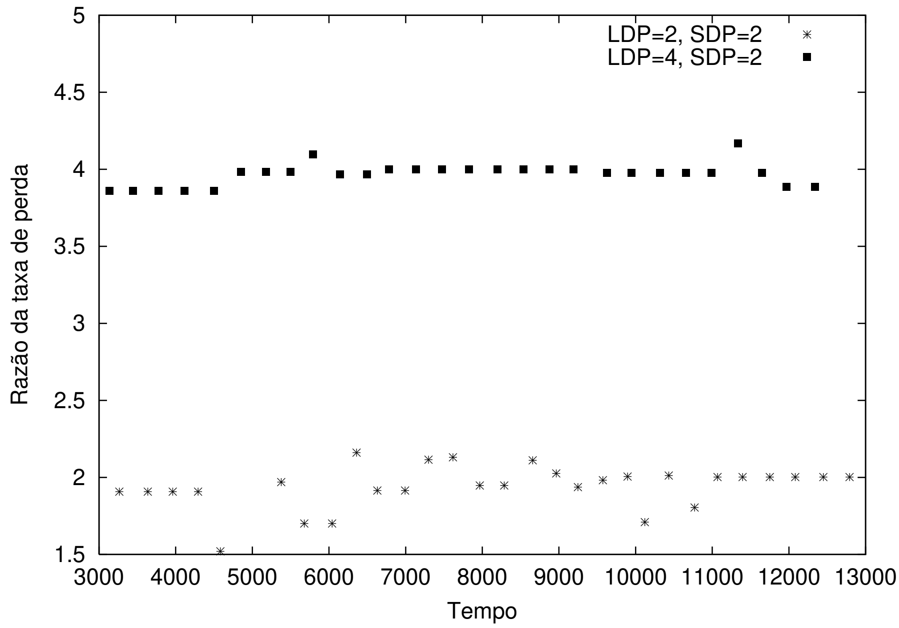
serem realizadas. Em todos os fluxos, CBR ou Pareto, todos os pacotes utilizados possuem o mesmo tamanho de 1000 octetos.

Resultados

A Figura 4.3 exibe a razão entre as taxas de perda ocorridas nas filas das classes 1 e 2, para simulações com fontes CBR e Pareto. Em ambos os casos, a taxa de chegada média em cada classe se mantém alta e constante, sendo a distribuição de carga estacionária. As taxas de perda são medidas em intervalos de tempo de curta duração. Cada ponto representado nos gráficos corresponde à razão entre as taxas de perda de ambas as classes, medida a cada T chegadas de pacotes. Estas medidas foram realizadas após um período de *warm-up* sendo T configurado pra 100.000 pacotes. Em todos os gráficos, os valores de SDPs e LDPs escolhidos são representados como $sdp_2/sdp_1 = SDP$, onde $SDP = 2$, e $ldp_1/ldp_2 = LDP$, onde $LDP = 2$ e 4.



(a) Fontes CBR



(b) Fontes Pareto

Figura 4.3: Razão da taxa de perda entre a classe 1 e 2.

Pode-se notar que em ambos os gráficos e na maior parte dos intervalos de tempo, o descartador PLR(∞) se aproxima da diferenciação proporcional especificada com restrições de $ldp_1/ldp_2 = 2$ e 4. Esses resultados foram obtidos sob uma distribuição de carga estacionária. Como analisado por Dovrolis et.al. em [38], esse descartador é afetado negativamente por mudanças na distribuição da carga, onde as taxas de perda se afastam das restrições especificadas. Porém, esse fato não afetará seu correto desempenho nas simulações principais deste trabalho, pois as conexões TCP de longa duração empregadas (fontes FTP), apresentam baixas variações na distribuição da carga quando obtêm o estado de equilíbrio [41, 44].

Vale observar a existência de alguns picos de magnitude significante no gráfico da Figura 4.3(b) correspondente a fontes Pareto, onde as razões da taxa de perda de curta duração não são iguais às razões especificadas. Existem dois motivos para este desvio. O primeiro motivo está relacionado à operação do descartador, pois em intervalos de curta duração (como é o caso das medidas periódicas exibidas para $T = 100.000$), as taxas de chegada podem se desviar significativamente de suas médias de longa duração devido a tráfego em rajada. Um descartador que mantém uma história mais longa do que T chegadas de pacotes, tal como o PLR(∞), descarta pacotes baseado nas taxas de perda medidas através de uma história mais longa do que os últimos T pacotes. Conseqüentemente, as taxas de perda de curta duração, as quais são medidas a cada T chegadas, podem se desviar das restrições de diferenciação proporcional.

O segundo motivo dos desvios observados na Figura 4.3(b), está relacionado à factibilidade do modelo proporcional. O descartador PLR não pode descartar um pacote de uma classe que está vazia (*idle*), sendo que tal descarte pode ser necessário para obter as restrições proporcionais de taxa de perda especificadas. Se a maior parte dos descartes no intervalo de T chegadas, ocorre durante uma rajada na qual a classe i está vazia, não será possível equalizar as taxas de perda normalizadas (seção 2.4.1) da classe i com a de outras classes. Outro aspecto relacionado à factibilidade é que se a taxa de perda em um intervalo de T chegadas é muito pequena, não será possível descartar suficientes pacotes de cada classe a fim de se ajustar proporcionalmente as taxas de perda. Em ambos esses casos, obtém-se uma diminuição dos desvios das restrições de diferenciação proporcional de taxa de perda, quando aumenta-se a escala de tempo de monitoramento de T.

Os resultados obtidos nessa etapa de simulação foram equivalentes aos apresentados por Dovrolis et.al. em [38]. O descartador PLR(∞) apresentou comportamento equivalente ao analisado na citação, para uma distribuição de carga estacionária. Desta forma, também considerou-se a implementação desse descartador validada e pronta para desempenhar sua função como componente do esquema de encaminhamento proporcional a ser empregado ao tráfego TCP, juntamente com o escalonador WTP e o gerenciador de *buffer DropTail-CS*.

4.2.3 Validação Completa do Esquema de Encaminhamento Proporcional

Nessa etapa, o esquema de encaminhamento proporcional pôde ser completamente validado sobre condições diversas. A seguir serão descritas cada fase desta validação. O escalonador WTP foi isoladamente validado na Seção 4.2.1, em um modelo sem perdas. Nessa seção os resultados da validação do esquema são estendidos a fim de avaliar a diferenciação de atraso que o WTP pode obter em um cenário mais realístico, com um tamanho de *buffer* limitado. Assim, simulações foram realizadas empregando-se a topologia da Figura 4.1, onde procurou-se verificar se em condições de perdas, o WTP aproxima satisfatoriamente do modelo proporcional de atraso descrito na Seção 2.3.1.

Da mesma forma que o WTP, o PLR(∞) foi validado na Seção 4.2.2 com uma distribuição de carga estacionária porém alta. Assim, nessa seção o desempenho desse descartador será avaliado também em carga moderada, onde será verificado se nessas condições, ele apresenta uma aproximação do modelo proporcional de taxa de perda descrito na Seção 2.3.1.

O modelo simulado nessa etapa de validação corresponde ao esquema de encaminhamento proporcional da Figura 2.4. Nesse esquema, o escalonador WTP serve as duas filas lógicas que compartilham o enlace de gargalo da Figura 4.1. O descartador de pacotes PLR(∞) realiza, quando necessário, o descarte de pacotes das filas, procurando-se manter uma diferenciação proporcional de taxa de perda. Os

Parâmetros de Diferenciação do Escalonador (*Scheduler Differentiation Parameters* - SDP) são configurados para $sdp_1 = 1$ e $sdp_2 = 2$, enquanto que os Parâmetros de Diferenciação de Taxa de Perda (*Loss rate Differentiation Parameters* - LDP) são configurados para $ldp_1 = 2$ e $ldp_2 = 1$. Por efeito de simplicidade, nos gráficos adotou-se a representação $sdp_2/sdp_1 = SDP$ e $ldp_1/ldp_2 = LDP$. Para os casos analisados, classes superiores têm desempenho melhor em ambas as métricas, de atraso e taxa de perda.

Nas filas, a política de compartilhamento completo é aplicada como política de organização de *buffer* (B), sendo o algoritmo *DropTail-CS* utilizado como gerenciador. O tamanho de B foi selecionado de forma que a taxa de perda gerada ficasse entre 0.5% e 2.5%.

Tráfego Monitorado

Nessa etapa, apenas fontes de tráfego Pareto foram empregadas, as quais foram configuradas como descrito a seguir. As fontes Pareto quando estão no estado de atividade (*burst*), geram um tráfego a uma taxa de pico de 213 kbps e são configuradas com um parâmetro de formato $\alpha = 1.5$. Os períodos de atividade são determinados por uma variável aleatória com distribuição Pareto de média 600 ms. Os períodos de silêncio (*idle*) também seguem uma distribuição Pareto de média 400 ms. Estes valores resultam em uma taxa média para o tráfego Pareto de 128 kbps. Em todos os fluxos Pareto, os pacotes utilizados possuem tamanho de 1000 octetos.

Para que o efeito da carga agregada no desempenho do escalonador WTP e do descartador PLR(∞) pudesse ser avaliado, foi variada a utilização do enlace de gargalo de aproximadamente $u = 75\%$ (carga moderada) a $u = 100\%$ (alta carga). Tal variação na carga foi realizada aumentando-se o número de fontes nos nós s_1 e s_2. Para que a taxa de perda gerada se mantivesse entre 0.5% e 2.5%, o tamanho de B também teve que ser alterado a medida que o número de fontes era aumentado. Os valores empregados para o tamanho de B e o número de fontes necessários para gerar as cargas plotadas no gráfico da Figura 4.4(a), podem ser vizualizados na Tabela 4.1.

Tabela 4.1: Valores empregados para B, quantidade de fontes e carga no enlace.

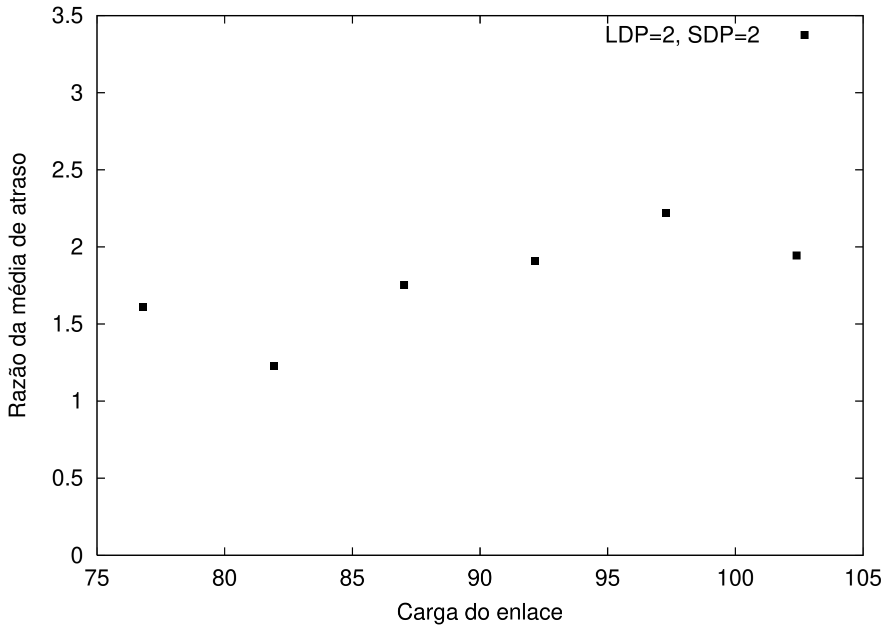
B	10	23	100	200	850	1000
Quant. de fontes	30	32	34	36	38	40
Carga	76.8%	81.92%	87%	92.16%	97.28%	102.4%

Resultados

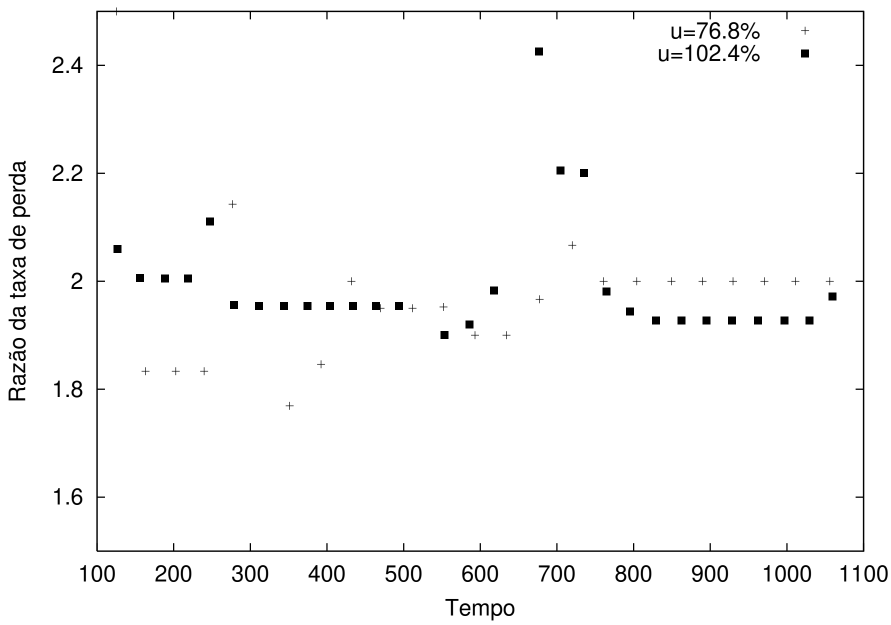
O primeiro gráfico da Figura 4.4 exibe a razão entre os atrasos médios ocorridos nas filas das classes 1 e 2, em presença de perdas e em condições de carga moderada e alta. Os atrasos médios de cada classe são calculados ao término da simulação (após um período de *warm-up*) e constituem atrasos médios de longa duração. Cada ponto nos gráficos apresentados, correspondem à média entre 10 amostras de simulação com diferentes sementes (*seeds*), sendo o tempo de simulação de cada amostra de 1100 unidades de tempo.

Pode-se notar que com o aumento da carga agregada no enlace de gargalo, o WTP tende a se aproximar da diferenciação proporcional de atraso médio estabelecida ($sdp_2/sdp_1 = SDP = 2$), isso mesmo em presença de perdas. Porém, como também observado nas análises realizadas na Seção 4.2.1, em cargas moderadas, o escalonador não mantém a diferenciação de atraso proporcional desejada. Desta forma, como discutido anteriormente, a ocorrência desse fato está relacionada à factibilidade da diferenciação proporcional de atraso, estando o desempenho do WTP de acordo com essa diferenciação, somente sobre condições de carga suficientemente alta.

Por outro lado, a razão entre as taxas de perda ocorridas nas filas das classes 1 e 2 é exibida no gráfico da Figura 4.4(b), para condições de carga moderada (76,8%) e alta (102.4%). Em ambos os casos, a taxa média de chegada em cada classe se mantém constante, sendo a distribuição de carga estacionária. As taxas de perda são medidas em intervalos de curta duração. Cada ponto representado nos gráficos corresponde à razão entre as taxas de perda de ambas as classes, medida a cada T chegadas de pacotes. Estas medidas foram realizadas após um período de *warm-up*, sendo T configurado pra 10.000 pacotes. No gráfico, a percentagem de utilização do enlace é indicado pela variável u.



(a) Razão do atraso médio entre as classes 1 e 2.



(b) Razão da taxa de perda entre as classes 1 e 2, para LDP e $SDP = 2$.

Figura 4.4: Razão da taxa de perda entre as classes 1 e 2.

Pode-se notar que para ambas as condições de carga e na maior parte dos intervalos de tempo, o descartador PLR(∞) se aproxima da diferenciação proporcional especificada com restrições de $ldp_1/ldp_2 = 2$. Esses resultados foram tirados sob uma distribuição de carga estacionária. Vale observar a existência de alguns picos de magnitude significante, onde as razões da taxa de perda de curta duração não são iguais às razões especificadas. Os motivos para este desvio são os mesmos já discutidos na Seção 4.2.2 e estão relacionados à operação do descartador e à factibilidade do modelo proporcional.

Os resultados obtidos nessa etapa de simulação foram equivalentes aos apresentados por Dovrolis et.al. em [38]. Os mecanismos de escalonamento e de descarte empregados pelo escalonador WTP e pelo descartador PLR(∞), respectivamente, apresentaram comportamento satisfatório, com uma margem de erro de no máximo 10%, e equivalente aos analisados na citação, mesmo em presença de perdas e condições variadas de carga. Desta forma, considerou-se completamente validado todas as partes constituintes do esquema de encaminhamento proporcional apresentado na Seção 2.4.1. Assim, o esquema apresenta-se em condições ideais à realização das simulações que se seguirão, e que constituem foco principal das pesquisas realizadas nesse trabalho.

4.3 Diferenciação Proporcional em Tráfego TCP

Nos últimos anos, vem se observando na Internet um grande crescimento em popularidade de aplicações adaptativas. Como exemplo desse tipo de aplicação, pode-se citar os produtos relacionados à telefonia IP. Porém, por causa da existência de enlaces mal provisionados e do modelo de "mesmo-serviço-para-todos" oferecido hoje na Internet, estas aplicações recebem níveis inaceitáveis de serviço. Uma vez que esse modelo seja trocado por esquemas de diferenciação relativa de serviço, os usuários e aplicações contariam com múltiplas classes de serviço como um nível de adaptação adicional, para controlar a QoS que elas recebem.

Um fator relevante e que vem impulsionando várias pesquisas na Internet é o relacionado ao protocolo TCP. Tal atenção deve-se ao fato de ser este protocolo responsável pela geração de 85% a 95% dos *bytes* que trafegam hoje pela rede mundial. Ciente das novas demandas de tráfego e aplicações na Internet, é que pretende-se avaliar no presente trabalho, o comportamento de um esquema de diferenciação relativa de serviço em tráfego TCP.

Tendo sido o esquema de encaminhamento proporcional implementado e corretamente validado (seção 4.2), essa seção apresenta as simulações realizadas com tráfego TCP. Mais especificamente, as simulações apresentadas objetivam avaliar se, no esquema implementado, as proporções estabelecidas para atraso médio nas filas e taxa de perda de pacotes entre as classes de serviço são garantidas, quando aplicado fontes de tráfego TCP. Várias pesquisas [41, 42, 43, 44] vêm sendo realizadas sobre o protocolo TCP, onde é avaliado seu comportamento no ajuste da janela de transmissão a fim de se controlar o congestionamento na rede. Nessas pesquisas, foi constatado que o comportamento do TCP é fortemente influenciado pelas perdas de pacotes e atrasos ocorridos nas filas em cada nó (capítulo 3). Assim, essa seção avalia este tráfego quando submetido a um modelo de diferenciação proporcional, onde as métricas de atraso nas filas e taxa de perdas de pacotes são tratadas.

No contexto do modelo de diferenciação proporcional implementado, primeiramente será avaliado seu desempenho em relação à métrica taxa de perda. Em seguida, a métrica de atraso médio ocorrido nas filas será levada em consideração. Vale ressaltar que nas duas avaliações são empregados, simultaneamente, mecanismos de escalonamento e de descarte que realizam diferenciação proporcional. Além disso, em todos os casos analisados, classes superiores possuem desempenho melhor em ambas as métricas de desempenho, ficando a cargo de simulações futuras outras combinações destas métricas.

4.3.1 Topologia de Simulação

Todos os casos analisados nesta seção seguem o modelo de topologia descrito na Figura 4.1. Todos os nós são interligados por enlaces de 10Mbps e 1ms de atraso de propagação, com exceção do enlace de gargalo entre os nós r_1 e r_2 com capacidade de 5Mbps e atraso de propagação de 10ms, o que resulta em um atraso de propagação ida e volta (τ) de 24ms. Os nós s_1 e s_2 possuem fontes responsáveis pela geração de tráfego de duas classes de serviço, as quais devem ser tratadas diferenciadamente no enlace de gargalo. Os pacotes gerados por esses nós são enviados para os nós de destino t_1 e t_2, respectivamente.

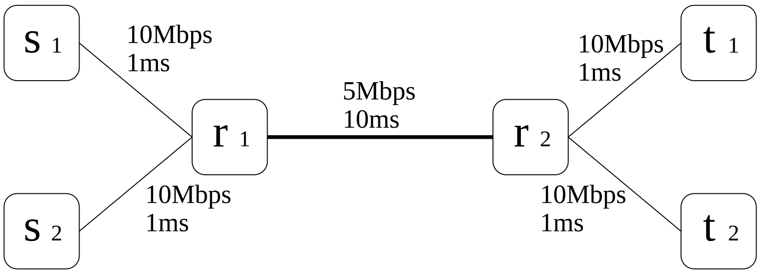


Figura 4.5: Topologia de simulação.

As fontes geram tráfego TCP e utilizam a implementação do TCP-Reno que adiciona o algoritmo de recuperação rápida (*fast recovery*) aos mecanismos de controle de congestionamento originais do TCP. O tráfego TCP é gerado por aplicações FTP ativas durante toda a simulação. Todos os pacotes tem o tamanho de 1000 bytes. Nas configurações com tráfego TCP realizadas no simulador *NS-2*, o controle de fluxo não influencia no controle de congestionamento realizado pelo TCP, não fornecendo assim, limites ao crescimento de sua janela.

O enlace de gargalo é compartilhado por duas filas lógicas. A primeira fila recebe tráfego gerado pelo nó s_1 e corresponde à classe menos prioritária (classe 1). A segunda recebe tráfego gerado pelo nó s_2 e corresponde à classe mais prioritária (classe 2). Ambas as filas são servidas pelo escalonador de pacotes proporcional WTP, onde os Parâmetros de Diferenciação do Escalonador (*Scheduler Differentiation Parameters* - SDP) são configurados para $sdp_1 = 1$ e $sdp_2 = 1$, 2 e 4, para a classe 1 e 2 respectivamente. Assim, a prioridade dos pacotes da classe 2 de serem escalonados, aumentará 1, 2 e 4 vezes mais no tempo de espera na fila, do que a da classe 1. O descartador de pacotes PLR(∞) realiza o descarte de pacotes das filas, estando os Parâmetros de Diferenciação de Taxa de Perda (*Loss rate Differentiation*

Parameters - LDP) configurados para $ldp_1 = 1$, 2 e 4 e $ldp_2 = 1$, para a classe 1 e 2, respectivamente. Assim, a classe 1 terá 1, 2 e 4 vezes mais perdas do que a classe 2. Por efeito de simplicidade, nos gráficos adotou-se a representação $sdp_2/sdp_1 = SDP$ e $ldp_1/ldp_2 = LDP$ empregados em cada simulação.

Nas filas, a política de compartilhamento completo é aplicada como política de organização de *buffer*, sendo o *DropTail-CS* utilizado como gerenciador de *buffer*.

Objetivando a utilização máxima da capacidade do enlace disponível tão logo a fase inicial de *slow start* do TCP termine, o *buffer* B foi configurado para valores igual e acima de $BW * \tau$, onde BW é a largura de banda do enlace de gargalo e τ é o maior atraso de propagação ida e volta do meio físico, na topologia empregada. Sendo a memória da rede igual a $BW * \tau$ [1], configurar o tamanho de B para pelo menos este valor, permitirá que, antes que pacotes sejam perdidos, a janela de congestionamento do emissor cresça até $2 * BW * \tau$, o que corresponde a duas vezes a vazão da rede. Assim, a forma de onda da janela de congestionamento oscilará entre uma e duas vezes o produto da largura de banda pelo atraso de propagação. Pesquisas relacionadas a essas análises podem ser encontradas em [70].

Na topologia empregada, $BW = 5Mbps$ e $\tau = 24ms$. Assim, para as simulações realizadas, foram atribuídos à B tamanhos de 15 ($BW * \tau$), 30 ($2 * BW * \tau$), 60 ($3 * BW * \tau$) pacotes e, em alguns casos, 120 ($4 * BW * \tau$). Apesar de tais valores permitirem o crescimento da janela de congestionamento para, respectivamente, 2, 3, 4 e 5 vezes a vazão disponível na rede, a maior motivação para o emprego destes valores foi o interesse em se avaliar a influência de *buffers* maiores no desempenho do modelo de diferenciação.

Todos os resultados apresentados nesta seção originaram de simulações realizadas em 1000 unidades de tempo. Porém, o monitoramento das filas foi inicializado após 10 unidades de tempo de simulação (*warm-up period*), objetivando evitar o período transiente.

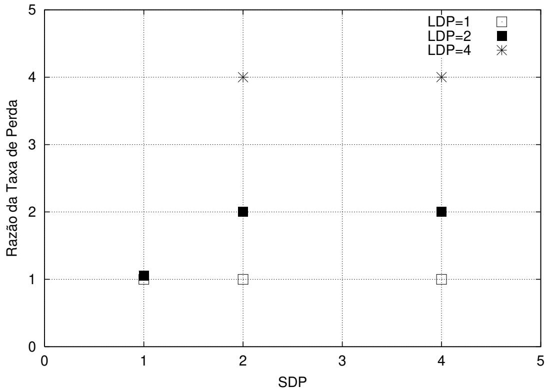
[1] Corresponde à quantidade de dados ainda não reconhecidos que estarão em trânsito a fim de manter o canal cheio, assumindo nenhuma perda de dados.

4.3.2 Avaliação da Taxa de Perda

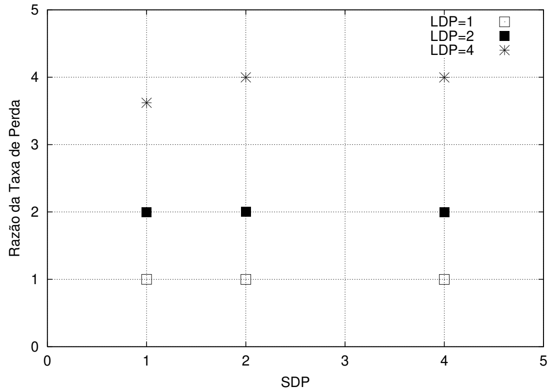
Durante todo o tempo de simulação e após um período inicial de *warm-up*, as filas das duas classes de serviço foram monitoradas, sendo gerados arquivos com estatísticas da quantidade de pacotes que chegaram, foram descartados e saíram de cada fila. Baseado nessas estatísticas, pôde-se obter a taxa de perda ocorrida em cada classe e posteriormente, calcular a razão da taxa de perda da classe 1 sobre a classe 2 (l_1/l_2). Em todos os gráficos apresentados, os valores de sdps e ldps empregados, são representados como $sdp_2/sdp_1 = SDP$ e $ldp_1/ldp_2 = LDP$, sendo sdp_1 e ldp_2 iguais a 1.

A fim de avaliar as condições de factibilidade e o desempenho do modelo proporcional implementado em relação à taxa de perda, simulações com 1 e 16 fontes de tráfego e diferentes tamanhos de B foram geradas. A Figura 4.6(a) exibe a razão da taxa de perda entre as duas classes, quando aplicado um B de tamanho 15 (em pacotes) e apenas uma fonte de tráfego TCP, em cada classe.

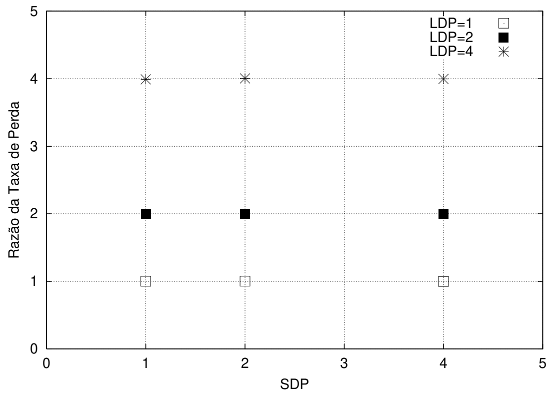
Pode-se observar que o descartador PLR(∞) atende a todas as restrições de diferenciação proporcional especificadas ($LDP = 1$, 2 e 4) quando $SDP = 2$ e 4. Para $SDP = 1$, as restrições de proporcionalidade correspondentes a $LDP = 2$ e 4 não são atendidas. A razão para este desvio está relacionada à factibilidade do modelo proporcional, pois o descartador PLR não pode descartar um pacote de uma fila que está vazia, podendo ser este descarte necessário para obter as restrições de taxa de perda proporcionais especificadas. Para manter restrições de proporcionalidade de taxa de perda (LDP) de 2 e 4, os pacotes da classe 1 (menos prioritária) devem ser descartados 2 e 4 vezes mais do que os da classe 2 (mais prioritária), respectivamente. Porém, quando a restrição proporcional de atraso é 1 ($SDP = 1$), as filas são servidas igualitariamente pelo escalonador WTP, passando os pacotes da classe 1 a sofrerem atrasos na fila na mesma proporção que os da classe 2. Assim, em conseqüência desta igualdade no escalonamento e da existência apenas de uma fonte de tráfego TCP em cada classe, não existem pacotes suficientes na classe 1 para serem descartados a fim de garantir a restrição de taxa de perda de 2 e 4 especificadas. A ocorrência deste fato pôde ser verificada pela quantidade de vezes que a classe 1



(a) $B = 15, \frac{sdp_2}{sdp_1} = SDP, \frac{ldp_1}{ldp_2} = LDP$



(b) $B = 30, \frac{sdp_2}{sdp_1} = SDP, \frac{ldp_1}{ldp_2} = LDP$



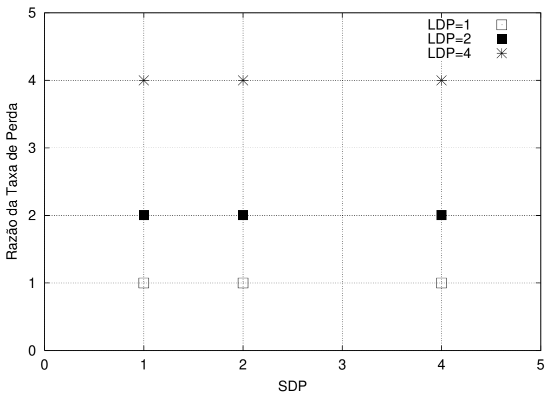
(c) $B = 60, \frac{sdp_2}{sdp_1} = SDP, \frac{ldp_1}{ldp_2} = LDP$

Figura 4.6: Razão da taxa de perda entre as classes 1 e 2 com uma conexão TCP por classe.

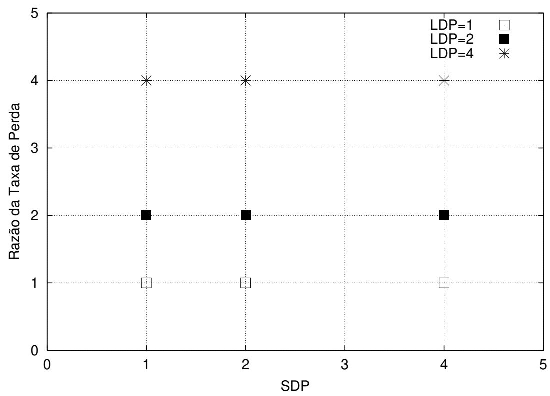
ficou vazia durante todo o período de simulação.

Aumentando-se o tamanho de B para 30, pode-se observar uma melhora no atendimento às restrições de $LDP = 2$ e 4, para $SDP = 1$, Figura 4.6(b). Esta melhora deve-se ao fato de que aumentando-se o tamanho do *buffer*, mais pacotes podem ser armazenados, diminuindo assim, as chances de não haver pacotes da classe 1 para serem descartados, o que permitiu melhoras na obtenção da proporção estabelecida. Porém, pode ser observado também, que estas condições não foram suficientes para garantir a restrição de $LDP = 4$, quando $SDP = 1$. Por outro lado, para um $B = 60$, Figura 4.6(c), todas as restrições de proporcionalidade de taxa de perda são garantidas, permitindo o suficiente armazenamento de pacotes da classe 1 necessários à manutenção dessas restrições. Ao aumentar o tamanho do *buffer*, observou-se reduções significativas na quantidade de vezes que a classe 1 ficou vazia, onde pôde-se comprovar a presença de mais pacotes na fila desta classe.

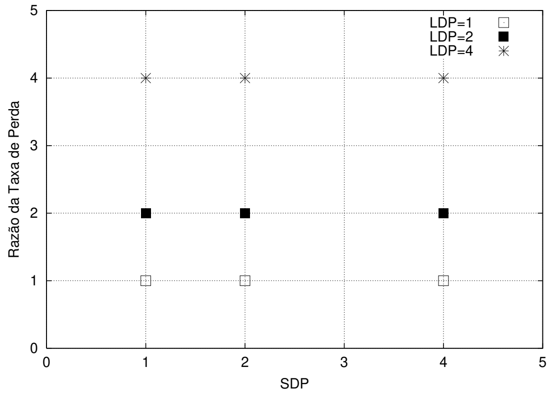
Utilizando um maior número de fontes de tráfego TCP em cada classe, constatou-se nas simulações realizadas que o problema de factibilidade do modelo também pôde ser contornado. A Figura 4.7 exibe os resultados obtidos utilizando-se 16 fontes em cada nó. Com um maior número de fontes, mesmo para $B = 15$, são gerados pacotes suficientes para evitar que a classe menos prioritária fique vazia, permitindo assim a garantia das restrições estabelecidas.



(a) $B = 15$



(b) $B = 30$



(c) $B = 60$

Figura 4.7: Razão da taxa de perda entre as classes 1 e 2 com 16 conexões TCPs por classe.

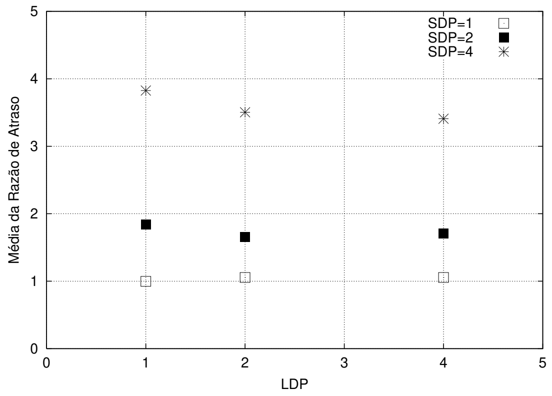
4.3.3 Avaliação do Atraso nas Filas

Da mesma forma que na avaliação anterior, as filas das duas classes de serviço, durante todo o tempo de simulação e após um período inicial de *warm-up*, foram monitoradas, sendo gerados arquivos com estatísticas sobre a média de atraso, a variância e a quantidade de pacotes que saíram de cada fila. Baseado nessas estatísticas, pôde-se calcular a razão da média de atraso da classe 1 sobre a da classe 2 ($\overline{d_1}/\overline{d_2}$). Em todos os gráficos apresentados, os valores de sdps e ldps empregados, são representados como $sdp_2/sdp_1 = SDP$ e $ldp_1/ldp_2 = LDP$, sendo sdp_1 e ldp_2 iguais a 1.

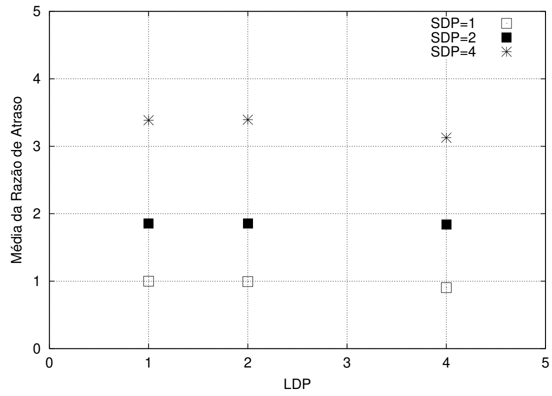
Simulações com 1 e 16 fontes de tráfego e um número variado de tamanho de B foram executadas, com o objetivo de avaliar as condições de factibilidade e o desempenho do modelo proporcional implementado em relação à métrica de atraso. A Figura 4.8(a) exibe a razão entre o atraso médio das filas das classes 1 e 2, quando aplicado um B de tamanho 15 (em pacotes) e apenas uma fonte de tráfego TCP, em cada classe.

Pode-se observar que para $B = 15$, obtém-se um desvio das restrições de diferenciação proporcional de atraso especificadas em $SDP = 2$ e 4. Quando aumenta-se as restrições de taxa de perda ($LDP = 2$ e 4), este desvio tende a aumentar. Este comportamento deve-se à factibilidade do modelo implementado, pois pesquisas realizadas em [37] demonstram que o escalonador WTP é factível apenas em altas cargas de tráfego. Este escalonador não pode servir um pacote de uma fila que está vazia, podendo ser este escalonamento necessário à obtenção das restrições proporcionais de atraso especificadas. Além disso, se as duas filas permanecerem vazias durante a maior parte do tempo da simulação, não será possível escalonar suficientes pacotes de cada classe a fim de se ajustar proporcionalmente os atrasos médios das filas.

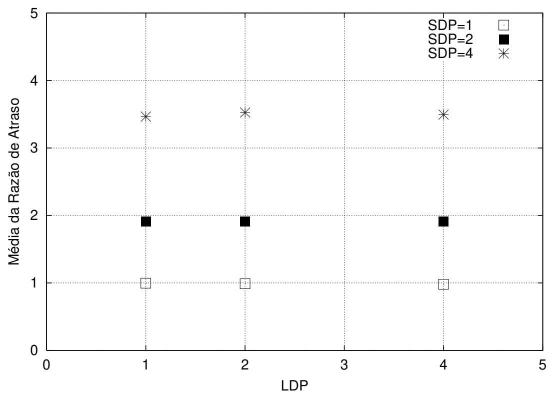
A fim de detectar ocorrências de infactibilidade no modelo, procurou-se verificar a quantidade de vezes que as classes ficam vazias durante o tempo de simulação. Pôde-se constatar várias ocorrências desse fato e com números bastante elevados,



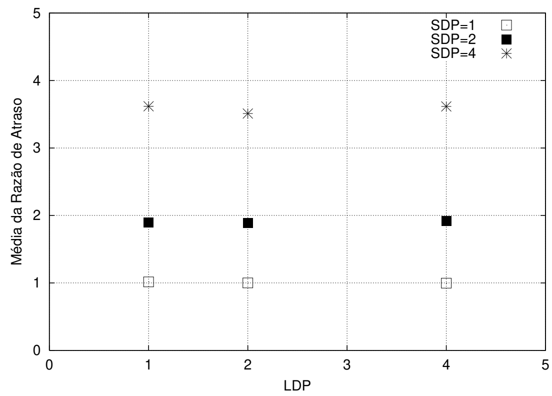
(a) $B = 15$



(b) $B = 30$



(c) $B = 60$



(d) $B = 120$

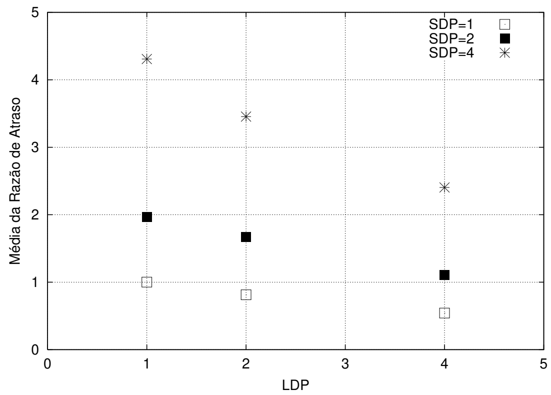
Figura 4.8: Atraso Médio entre as classes 1 e 2 com uma conexão TCP por classe.

onde a fila da classe menos prioritária (classe 1) apresentou sempre estar mais na condição de vazia do que a classe mais prioritária (classe 2).

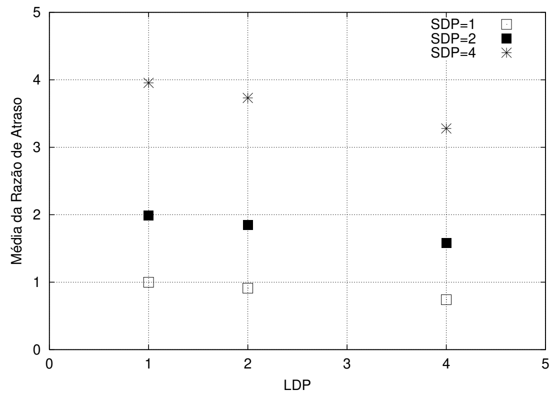
Aumentando-se o tamanho do *buffer*, Figura 4.8, os resultados apresentaram significativas melhoras para a restrição de atraso igual a 2 (SDP). Esta melhora deve-se ao fato de que com um maior tamanho de *buffer*, mais pacotes podem ser armazenados, diminuindo assim, as chances de não haver suficientes pacotes da classe 1 a serem servidos, ou de ambas as classes estarem vazias. Em adição a este fato, vale lembrar que para *buffers* maiores, maior será o crescimento inicial da janela de congestionamento das duas conexões TCP utilizadas, ficando o canal mais cheio e, por conseguinte, as filas também.

Para a restrição de atraso igual a 4 ($SDP = 4$), observou-se variações na obtenção da proporcionalidade estabelecida (Figura 4.8). O aumento do tamanho do *buffer* para 120 pacotes reduziu o número de vezes que as classes ficaram vazias. Essa redução permitiu alcançar, com uma diferença da ordem de 10%, a restrição de diferenciação proporcional de atraso estabelecida ($SDP = 4$). Verificou-se que a quantidade de vezes que as classes ficam vazias durante o tempo de simulação, ainda apresentam um número bastante elevado, onde a fila da classe menos prioritária (classe 1) apresentou sempre estar mais na condição de vazia do que a classe mais prioritária (classe 2). Pôde-se verificar também que com o aumento do tamanho dos *buffers*, maior é a aproximação da restrição estabelecida.

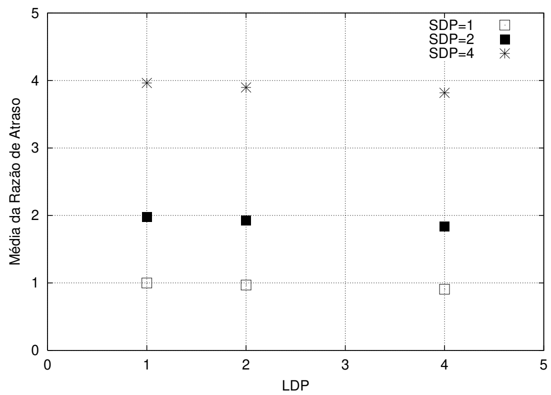
Aplicando-se um maior número de fontes de tráfego TCP nas classes, também pôde-se verificar, para B=15, a ocorrência de uma grande quantidade de vezes que a classe 1 fica vazia. Assim, verifica-se pelo gráfico da Figura 4.9(a), que aumentando-se a proporção de taxa de perda (ldp_1) - as quais estão sendo garantidas de acordo com o gráfico exibido na Figura 4.7(a) - os resultados apresentam um considerável desvio da restrição proporcional de atraso estabelecida, conseqüência da não existência de suficientes pacotes da classe 1 no *buffer* para serem servidos.



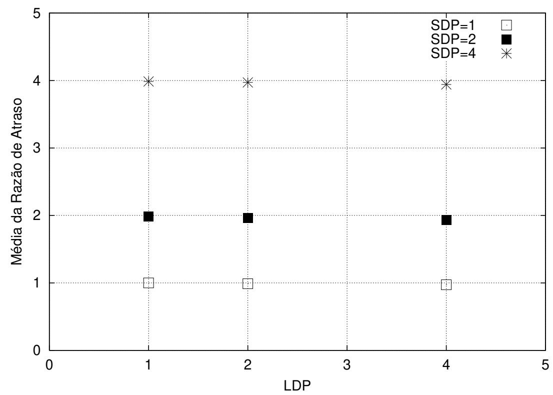
(a) $B = 15$



(b) $B = 30$



(c) $B = 60$



(d) $B = 120$

Figura 4.9: Atraso Médio entre as classes 1 e 2 com 16 conexões TCPs por classe.

Com uma quantidade maior de *buffer*, Figura 4.9, observa-se melhoras significativas na obtenção das restrições de atraso estabelecidas no modelo implementado, chegando-se a alcançar condições ideais para $B = 120$, Figura 4.9(d). Nestes casos, pôde-se verificar, de acordo com os parâmetros *ldp* e *sdp* empregados, poucas ou nenhuma ocorrência de fila vazia para ambas as classes de serviço.

Capítulo 5

Conclusão

NESTE trabalho foi apresentado a análise do desempenho de um modelo de diferenciação proporcional de serviço considerando tráfego TCP. Como principal contribuição, este trabalho avalia uma proposta de provisão relativa de Qualidade de Serviço na Internet e leva em consideração o protocolo responsável pela geração de 85% a 95% dos *bytes* que trafegam hoje pela rede mundial. Nas análises realizadas, foram utilizadas as métricas de desempenho de atraso nas filas e taxa de perda, as quais exercem forte influência no comportamento do protocolo TCP. Diversas, simulações foram realizadas objetivando a verificação da garantia das proporções estabelecidas para atraso e perdas entre as classes de serviço.

A primeira seqüência de simulações enfocou a validação do esquema de encaminhamento proporcional implementado no simulador *NS-2*. Tráfego UDP foi gerado, onde empregou-se tanto fontes de tráfego modeladas por fontes do tipo Pareto quanto por fontes CBR. Os algoritmos referentes aos mecanismos de escalonamento e descarte proporcionais do esquema foram validados isoladamente no esquema completo. As etapas de validação consistiram na repetição de algumas simulações realizadas por Dovrolis et al. [37, 38] e na comparação dos resultados obtidos. Em todas as simulações implementadas, os resultados obtidos com tráfego Pareto mostram-se equivalentes aos originalmente apresentados nas citações, aproximando-se adequadamente das proporções desejadas. Em adição a esta validação, algumas simulações foram realizadas para fontes de tráfego CBR que também exibiram uma diferenciação proporcional adequada entre as classes, estando de acordo com as proporções estabelecidas.

É importante ressaltar que atrasos nas filas e perdas só ocorrem em situações de congestionamento. São para estas situações que a diferenciação de serviços deve ser aplicada. Assim, o escalonador WTP tende para uma diferenciação proporcional de atraso somente sobre condições de carga suficientemente alta. Mais especificamente, foram obtidos resultados mais precisos, ou seja, com uma margem de erro inferior a 10%, para uma utilização do enlace acima de 97%, em condições variadas de perdas e tamanho de *buffer*.

Simulações de validação com o descartador proporcional PLR(∞) foram realizadas na presença de uma distribuição de carga estacionária moderada e alta, e medições de curta duração das taxas de perda. Os resultados obtidos apresentam um desempenho apropriado deste descartador, quanto a obtenção das distribuições proporcionais de taxa de perda estabelecidas entre as classes de serviço. Nos gráficos correspondentes a fontes Pareto, observa-se a existência de alguns desvios de magnitude significante das restrições de taxa de perda especificadas. Tais desvios estão relacionados à operação do descartador em medições de curta duração e à factibilidade do modelo proporcional, sendo causados por tráfego em rajada, por falta de pacotes nas classes para serem descartados ou pela baixa ocorrência de perdas. Os resultados são suficientes para considerar o descartador PLR(∞) validado e mantém equivalência com os apresentados por Dovrolis et al. [38]. Com o término destas simulações concluiu-se a validação apropriada do esquema de encaminhamento proporcional implementado.

Em uma segunda seqüência de simulações avaliou-se sob tráfego TCP, a garantia da diferenciação proporcional entre classes de serviço para as métricas de atraso e taxa de perda. Aplicando-se simultaneamente mecanismos de escalonamento e de descarte proporcionais, primeiramente foi avaliado o desempenho do modelo de diferenciação proporcional em relação à métrica de taxa de perda. Em seguida, a métrica de atraso médio ocorrido nas filas foi levada em consideração. Em todas as simulações realizadas ambientes com 1 e 16 fontes de tráfego e um número variado de tamanho de *buffer* foram considerados.

Os resultados obtidos mostram que o modelo se comporta adequadamente, mantendo as proporcionalidades estabelecidas entre as classes. As análises realizadas mostram que a proporcionalidade entre as classes, tanto em relação à métrica de taxa de perda quanto ao atraso, é melhor atendida quanto maior é o tamanho dos *buffers*. Para *buffers* com tamanho acima de 60 pacotes chega-se a resultados precisos das proporcionalidades entre as classes com uma diferença da ordem de 10%. Isso ocorre pois, com o aumento do tamanho dos *buffers*, observa-se um aumento da probabilidade de todas as classes de serviço possuirem pacotes em suas filas. Para tamanho de *buffers* pequenos, ocorrem situações onde a fila de alguma classe de serviço fica vazia, prejudicando assim a garantia da proporcionalidade entre as classes. A proporcionalidade também é melhor atendida quanto maior o número de conexões TCP. No entanto, para a métrica de atraso, observa-se uma determinada degeneração do processo quando o número de *buffers* é pequeno, apresentando uma pior diferenciação proporcional com o aumento do número de conexões. Isto ocorre porque neste caso, a classe mais prioritária praticamente monopoliza o *buffer*. Este trabalho comprova o bom funcionamento do modelo proporcional mesmo para tráfegos do tipo TCP, o qual possui comportamento afetado por perdas e atrasos. Derivou-se das pesquisas realizadas neste trabalho, o artigo "Avaliação de um Ambiente de Diferenciação de Serviço Proporcional em Tráfego TCP", aprovado para publicação e apresentação no XIX Simpósio Brasileiro de Redes de Computadores a ser realizado em Florianópolis em maio deste ano.

Como sugestões para trabalhos futuros, podem ser destacados:

- a simulação com um número mais abrangente de topologias, eventualmente utilizando uma ferramenta para a geração destas topologias;

- a avaliação do modelo de diferenciação proporcional onde serviços bi-direcionais são oferecidos, ou seja, onde as classes superiores possuem um melhor desempenho apenas em uma das métricas, de atraso ou de taxa de perda, ficando a classe inferior com a outra métrica;

- a análise do impacto da diferenciação proporcional de atraso e de taxa de perda na vazão do TCP;

- o emprego de um gerenciador ativo de *buffer* como o RED, que poderia proporcionar melhor utilização do enlace além de menor taxa de perda e atraso nas filas;

- a utilização de outras implementações do TCP, tipo TCP New Reno, ou TCP SACK, que provêem formas mais eficientes de reagir às diversas condições de perdas de pacotes.

Referências Bibliográficas

[1] Shenker, S. Fundamental design issues for the future Internet. *IEEE Journal on Selected Areas in Communications, v. 13*, n. 7 (setembro de 1995), 1176–1188.

[2] Xiao, X. e Ni, L. Internet QoS: the big picture. *IEEE Network, v. 13*, n. 2 (março de 1999), 1–13.

[3] Lefelhocz, C., Lyles, B., Shenker, S. e Zhang, L. Congestion control for best-effort service: Why we need a new paradigm. *IEEE Network, v. 10*, n. 1 (janeiro de 1996), 10–19.

[4] Clark, D. D. Adding service discrimination to the Internet. Relatório técnico, MIT, setembro de 1995.

[5] Kumar, V. P., Lakshman, T. V. e Stiliadis, D. Beyond best effort: Router architectures for the differentiated services of tomorrow's Internet. *IEEE Communications Magazine, v. 36*, n. 5 (maio de 1998), 152–164.

[6] Keshav, S. *An Engineering Approach to Computer Networking.* Addison-Wesley, 1997.

[7] Rosen, E. C., Viswanathan, A. e Callon, R. Multiprotocol label switching architecture. *Internet Draft* (abril de 1999). `draft-ietf-mpls-arch-05`.

[8] Rajagopalan, B. e Ma, Q. An overlay model for constraint-based routing. *Internet Draft* (janeiro de 1999). `draft-rajagopalan-CR-overlay-00`.

[9] Braden, R., Clark, D. D. e Shenker, S. Integrated services in the Internet architecture: an overview. *Internet RFC 1633* (junho de 1994).

[10] Blake, S., Black, D. L., Carlson, M., Davies, E., Wang, Z. e Weiss, W. An architecture for differentiated services. *Internet RFC 2475* (dezembro de 1998).

[11] Soares, L. F. G., Lemos, G. e Colcher, S. *Redes de Computadores - Das LANs, MANs e WANs às Redes ATM.* Editora Campus, 1995.

[12] Black, D. P. *Building Switched Networks.* Addison-Wesley, 1999.

[13] Moy, J. OSPF version 2. In *Internet RFC 2178* (julho de 1997).

[14] Crawley, E., Nair, R., Rajagopalan, B. e Sandick, H. A framework for QoS-based routing in the Internet. *Internet RFC 2386* (agosto de 1998).

[15] Wang, Z. e Crowcroft, J. Quality of service routing for supporting multimedia applications. *IEEE Journal on Selected Areas in Communications, v. 14*, n. 7 (setembro de 1996), 1228–1234.

[16] White, P. P. e Crowcroft, J. The integrated services in the Internet: State of the art. *Proceedings of the IEEE, v. 85*, n. 12 (dezembro de 1997), 1934–1946.

[17] de Rezende, J. F. e Romano, S. P. Arquitetura de integração de serviços para a comutação IP. In *XVI Simpósio Brasileiro de Redes de Computadores, SBRC'98* (Rio de Janeiro, RJ, maio de 1998).

[18] Zhang, L., Deering, S. E., Estrin, D., Shenker, S. e Zappala, D. RSVP: A new resource reservation protocol. *IEEE Network, v. 7*, n. 5 (setembro de 1993), 8–18.

[19] Braden, R., Zhang, L., Berson, S., Herzog, S. e Jamin, S. Resource reservation protocol (RSVP) – version 1 functional specification. *Internet RFC 2205* (setembro de 1997).

[20] Wroclawski, J. The use of RSVP with IETF integrated services. *Internet RFC 2210* (setembro de 1997).

[21] White, P. P. RSVP and integrated services in the Internet: A tutorial. *IEEE Communications Magazine, v. 35*, n. 5 (maio de 1997), 100–106.

[22] Shenker, S., Partridge, C. e Guerin, R. Specification of guaranteed quality of service. *Internet RFC 2212* (setembro de 1997).

[23] Wroclawski, J. Specification of the controlled-load network element service. *Internet RFC 2211* (setembro de 1997).

[24] Parekh, A. K. e Gallager, R. G. A generalized processor sharing approach to flow control in integrated services networks: The single node case. *IEEE/ACM Transactions on Networking, v. 1*, n. 3 (junho de 1993), 344–357.

[25] Parekh, A. K. e Gallager, R. G. A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. *IEEE/ACM Transactions on Networking, v. 2*, n. 2 (abril de 1994), 137–150.

[26] Berson, S. e Vincent, S. Aggregation of Internet integrated services state. *Internet Draft* (novembro de 1997). `draft-berson-classy-approach-01`.

[27] Mankin, A., Baker, F., Braden, R., Bradner, S., O'Dell, M., Romanow, A., Weinrib, A. e Zhang, L. Resource reservation protocol (RSVP) – version 1 applicability statement. *Internet RFC 2208* (setembro de 1997).

[28] Nichols, K., Blake, S., Baker, F. e Black, D. L. Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers. *Internet RFC 2474* (dezembro de 1998).

[29] Baumgartner, F., Braun, T. e Habergger, P. Differentiated services: A new approach for quality of service in the Internet. In *Proceedings of the 8th IFIP Conference on High Performance Networking – HPN'98* (setembro de 1998).

[30] Postel, J. Internet Protocol. *Internet RFC 791* (setembro de 1981).

[31] Almquist, P. Type of service in the Internet protocol suite. *Internet RFC 1349* (julho de 1992).

[32] Deering, S. e Hinden, R. Internet Protocol, version 6 (IPv6) specification. *Internet RFC 2460* (dezembro de 1998).

[33] Heinanen, J., Baker, F., Weiss, W. e Wroclawski, J. Assured forwarding PHB group. *Internet RFC 2597* (junho de 1999).

[34] Jacobson, V., Nichols, K. e Poduri, K. An expedited forwarding PHB. *Internet RFC 2598* (junho de 1999).

[35] Dovrolis, C. e Stiliadis, D. Relative differentiated services in the internet: Issues e mechanisms. In *Proceedings of ACM SIGMETRICS'99* (maio de 1999).

[36] Dovrolis, C. e Ramanathan, P. A case for relative differentiation services and the proportional differentiation model. *IEEE Network, v. 13*, n. 5 (setembro de 1999).

[37] Dovrolis, C., Stiliadis, D. e Ramanathan, P. Proportional differentiated services: Delay differentiation and packet scheduling. In *Proceedings of ACM SIGCOMM'99* (setembro de 1999).

[38] Dovrolis, C. e Ramanathan, P. Proportional differentiated services, part ii: Loss rate differentiation and packet dropping. In *Proceedings of IWQoS'00* (junho de 2000).

[39] Odlyzko, A. Paris metro pricing: The minimalist differentiated services solution. In *Proceedings of the International Workshop on QoS - IWQoS'99* (junho de 1999).

[40] Bennett, J. C. R. e Zhang, H. Hierarchical packet fair queueing algorithms. *IEEE/ACM Transactions on Networking, v. 5*, n. 5 (outubro de 1997), 675–689.

[41] Jacobson, V. Congestion avoidance and control. In *Proceedings of the SIGCOMM'88* (agosto de 1988).

[42] Fall, K. e Floyd, S. Simulations-based comparisons of Tahoe, Reno and SACK TCP. *IEEE Computer Communications Magazine, v. 26*, n. 3 (julho de 1996).

[43] Stevens, W. R. TCP slow start, congestion avoidance, fast retransmit and fast revovery algorithms. In *Internet RFC 2001* (janeiro de 1997).

[44] Mathis, M., Semke, J. e Mahdavi, J. The macroscopic behavior of the TCP congestion avoidance algorithm. In *Proceedings of ACM SIGCOMM'97* (julho de 1997).

[45] Network Simulator (NS). `http://www-mash.cs.berkeley.edu/ns/`.

[46] Differentiated Services (DiffServ) IETF Working Group. `http://www.ietf.org/html.charters/diffserv-charter.html`.

[47] Braden, R. Requirements for internet hosts – communication layers. *Internet RFC 1122* (outubro de 1989).

[48] Ziviani, A., de Rezende, J. F. e Duarte, O. C. M. B. Avaliando o suporte ao tráfego de voz em um ambiente de diferenciação de serviços. In *Anais do VI Simpósio Brasileiro de Multimídia e Sistemas Hipermídia - SBMIDIA'2000* (Natal, RN, junho de 2000), pp. 341–356.

[49] Ziviani, A., de Rezende, J. F. e Duarte, O. C. M. B. Tráfego de voz em um ambiente de diferenciação de serviços na Internet. In *Anais do XVII Simpósio Brasileiro de Redes de Computadores, SBRC'99* (Salvador, BA, maio de 1999), pp. 354–367.

[50] Ziviani, A., de Rezende, J. F. e Duarte, O. C. M. B. Análise de mecanismos de diferenciação de serviços para tráfego de voz na Internet. In *Anais do XIX Congresso Nacional da Sociedade Brasileira de Computação, XXVI Seminário Integrado de Software e Hardware, SEMISH'99* (Rio de Janeiro, RJ, julho de 1999), vol. 1, pp. 23–37.

[51] Ziviani, A., de Rezende, J. F. e Duarte, O. C. M. B. Towards a differentiated services support for voice traffic. In *Proceedings of the IEEE Global Telecommunications Conference - GLOBECOM'99* (Rio de Janeiro, Brasil, dezembro de 1999), vol. 1, pp. 59–63.

[52] DE REZENDE, J. F. Avaliação do serviço assegurado para a diferenciação de serviços na Internet. In *Anais do XVII Simpósio Brasileiro de Redes de Computadores, SBRC'99* (Salvador, BA, maio de 1999), pp. 339–353.

[53] DE REZENDE, J. F. Assured service evaluation. In *Proceedings of the IEEE Global Telecommunications Conference - GLOBECOM'99* (Rio de Janeiro, Brasil, dezembro de 1999), vol. 1, pp. 100–104.

[54] ALVES, I. B. H., REZENDE, J. F. E MORAES, L. F. M. Evaluating fairness in the aggregated traffic marking. In *Proceedings of the IEEE Global Telecommunications Conference - GLOBECOM'00* (novembro de 2000), vol. 1, pp. 445–449.

[55] DEMERS, A., KESHAV, S. E SHENKER, S. Analysis and simulation of a fair queueing algorithm. *Internetworking: Research and Experience* (1990), 3–26.

[56] STILIADIS, D. E VARMA, A. Latency-rate servers: A general model for analysis of traffic schedulingalgorithms. *IEEE/ACM Transactions on Networking, v. 6* (outubro de 1998).

[57] JAGANNATHAN, P., DURRESI, A. E JAIN, R. Proportional forwarding PHB. *Internet Draft* (fevereiro de 2001). `draft-jagannathan-diffserv-pf-00`.

[58] MORET, Y. E FDIDA, S. A Proportional Queue Control Mechanism to provide differentiated service. In *International Symposium on Computer and Information Systems (ISCIS)* (outubro de 1998).

[59] NANDAGOPAL, T., VENKITARAMAN, N., SIVAKUMAR, R. E BHARGHAVAN, V. Delay differentiation and adaptation in core stateless networks. In *Proceedings of IEEE INFOCOM'00* (março de 2000).

[60] FLOYD, S. E JACOBSON, V. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking, v. 1*, n. 4 (agosto de 1993), 397–413.

[61] KLEINROCK, L. A delay dependent queue discipline. *Journal of the ACM, v. 14*, n. 2 (1967), 242–261.

[62] Postel, J. Transmission Control Protocol. *Internet RFC 793* (setembro de 1981).

[63] Jacobson, V., Braden, R. e Borman, D. TCP extensions for high performance. *Internet RFC 1323* (maio de 1992).

[64] Jain, R. Congestion control in computer network: Issues and trends. *IEEE Network Magazine, v. 4*, n. 3 (maio de 1990), 24–30.

[65] Santos, G. P. T. e Vasconcellos, S. V. Avaliação por simulação dos mecanismos de controle de congestionamento do tcp. Relatório técnico, Programa de Engenharia Elétrica - COPPE/UFRJ, agosto de 2000.

[66] Tanenbaum, A. S. *Computer Network*. Prentice Hall, 1996.

[67] Stevens, W. R. *TCP/IP Ilustrated, Volume 1: The Protocols*. Addison-Wesley, 1994.

[68] Floyd, S. Connections with multiple congested gateways en packet-switched networks. *IEEE Computer Communications Magazine, v. 21*, n. 5 (outubro de 1991).

[69] Fall, K. e Varadhan, K. NS – notes and documentation. Relatório técnico, The VINT Project, janeiro de 1999.

[70] Srinivasan Keshav Thesis, 1991. `http://www.cs.cornell.edu/home/skeshav`.