Avaliação de um Ambiente de Diferenciação de Serviço Proporcional em Tráfego TCP

Aline Carneiro Viana   José F. de Rezende
Leonardo G. de R. Guedes   Otto Carlos M. B. Duarte

Grupo de Teleinformática e Automação – GTA
COPPE/Programa de Engenharia Elétrica
Universidade Federal do Rio de Janeiro
`http://www.gta.ufrj.br/`

Escola de Engenharia Elétrica – EEE
Universidade Federal de Goiás
`http://www.ufg.br/`

Resumo

O modelo de diferenciação proporcional constitui uma forma previsível e controlável de se fornecer diferenciação de serviços relativa. Este modelo garante que, para determinadas métricas, as classes superiores serão melhores do que classes inferiores, ou que no mínimo não serão piores. O modelo proporcional foi analisado para tráfego UDP. Este artigo avalia o desempenho do modelo proporcional para fontes de tráfego TCP, que é fortemente influenciado por perdas e atraso, devido ao mecanismo usado para controle de congestionamento. Foi implementado e validado um ambiente de encaminhamento de pacotes que segue o modelo de diferenciação de serviço proporcional. Os resultados das simulações comprovam o funcionamento apropriado do modelo proporcional para tráfego TCP, no que se refere às perdas e atrasos para conexões de longa duração.

Palavras-chave: Qualidade de Serviço (QoS), Internet, Tráfego TCP.

Abstract

The model of proportional differentiation is a predictable and controllable way of providing relative differentiation service. This model assures, to certain metrics, that higher classes will be better than lower classes, or at least won't be worse. The proporcional model was analyzed for the UDP traffic. This paper evaluates the proportional model performance for TCP traffic, which is strongly influenciated by losses and delay, due to the mechanism used for congestion avoidance. A packet forwarding environment that follows the proportional service differentiation model was implemented and evaluated. The simulation results confirm the appropriated functioning of the proportional model for TCP traffic, in terms of losses and delay for long term connections.

Keywords: *Quality of Service (QoS), Internet, TCP Traffic.*

Este trabalho foi realizado com recursos da UFG, UFRJ, CNPq, CAPES, COFECUB e FAPERJ.

1 Introdução

As atuais aplicações, por trafegarem uma variedade de mídias tais como voz, áudio, vídeo e dados, vêm exigindo diversos requisitos de qualidade de serviço. Para atender esta demanda é necessário trocar o atual paradigma de mesmo-serviço-para-todos (*best-effort*), por um modelo no qual os usuários, as aplicações ou os pacotes individuais, são tratados diferenciadamente, baseado em seus requisitos de qualidade de serviço. Dentro desta perspectiva, uma das principais tendências é a implantação de uma infra-estrutura de rede que suporte a diferenciação de serviços e que garantam a provisão da Qualidade de Serviço (*Quality of Service* - QoS).

Com o objetivo de permitir uma diferenciação de serviços de forma escalável e gerenciável, o modelo de Serviço Diferenciado (*Differentiated Service* - DiffServ) [1, 2] foi proposto pelo IETF (*Internet Engineering Task Force*). Duas abordagens distintas derivaram das pesquisas nesta área: o Serviço Diferenciado Absoluto [3, 4] e o Serviço Diferenciado Relativo [5].

Na primeira abordagem, o tráfego de um usuário, após passar pelo controle de admissão, 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 serviços de Encaminhamento Expresso (*Expedited Forwarding* - EF) [4] e de Encaminhamento Assegurado (*Assured Forwarding* - AF) [3] são exemplos desta abordagem. Por fornecer garantias absolutas de desempenho estabelecidas nos contratos de serviço, esta abordagem necessita de provisionamento de recursos e de controle de admissão, o que introduz uma considerável complexidade na arquitetura de QoS empregada.

Por outro lado, na abordagem de serviço diferenciado relativo [5], a rede assegura apenas 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. Neste contexto, aplicações e usuários não obtêm uma garantia de nível de desempenho absoluta, tal como um valor máximo de atraso fim-a-fim ou de largura de banda, o que torna desnecessário os mecanismos de provisionamento de recursos e controle de admissão. Tal característica reflete diretamente na arquitetura de QoS utilizada, tornando-a significativamente mais simples e fácil de empregar e gerenciar. Por outro lado, o crescimento da popularidade das aplicações adaptativas demonstra que usuários podem tolerar variações na qualidade de serviço oferecida. O modelo de diferenciação relativa se adequa a essa realidade, por permitir que tais aplicações se adaptem às classes oferecidas e selecionem as que melhor atendam suas necessidades.

Várias formas de se prover diferenciação de serviço relativa para usuários e operadores de rede foram propostas e avaliadas [6, 7, 8, 9, 10]. 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 [6, 7, 8] foram analisados mecanismos que realizam diferenciação de serviço relativa 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. O mecanismo de controle de congestionamento do protocolo TCP sofre influência dos atrasos nas filas e das perdas de pacotes ocorridas. Esta característica do protocolo TCP torna difícil prever o seu comportamento diante de um modelo de diferenciação de serviço relativa. Desta forma, o objetivo principal deste artigo é avaliar o desempenho do modelo de diferenciação de serviço relativa, quando submetido a um tráfego TCP. Foi implementado no simulador ns-2 [11], um esquema de encaminhamento de pacotes que fornece uma diferenciação de serviço relativa. Os resultados obtidos mostraram que o modelo realiza uma diferenciação relativa adequada.

Este artigo está estruturado da seguinte forma. A Seção 2 descreve o modelo proporcional como uma forma de se prover diferenciação de serviço relativa controlável e previsível. Apresenta também o esquema de encaminhamento que segue o modelo de diferenciação proporcional, onde são descritos os mecanismos que o compõe. Na Seção 3 são descritos os experimentos realizados, a topologia empregada e a análise dos resultados obtidos. Finalmente, a Seção 4 conclui o artigo.

2 Modelo de Diferenciação de Serviço Proporcional

A diferenciação de serviço relativa 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. Neste contexto, 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. Assim, tornam-se desnecessários os mecanismos de provisionamento de recurso e controle de admissão, o que introduz simplicidade em termos de implementação, desenvolvimento e gerenciabilidade da rede.

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, em termos de métricas de desempenho para encaminhamento de pacotes em cada nó, do que a classe $(i-1)$, para $1 < i \leq N$.

Várias formas de prover diferenciação de serviço relativa foram apresentadas e avaliadas em [7, 9, 10], porém para que uma arquitetura de serviço diferenciado relativo seja efetiva para usuários e operadores de rede, ela deve ser **previsível** - a diferenciação deve ser consistente (classes superiores são melhores) e independente das variações de carga nas classes - e **controlável** - os operadores de rede devem ser capazes de ajustar o espaçamento de qualidade entre as classes. Objetivando atender a esses requisitos, o Modelo de Diferenciação de Serviço Proporcional foi proposto e avaliado por Dovrolis et.al. em [7].

Este modelo estabelece que certas métricas de desempenho para o encaminhamento de pacotes em cada nó, devem ser proporcionais a certos parâmetros de diferenciação de qualidade que o operador de rede escolhe. Através destes parâmetros de diferenciação, o operador de rede pode controlar o espaçamento entre as classes, tornando o modelo controlável.

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 de atraso de fila proporcional, como mostra a Equação 1.

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

onde $\overline{d_i}$ é o atraso médio sofrido pelos pacotes na fila da classe i, e 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.

$$\overline{l_i} = \frac{ldp_i}{ldp_j} * \overline{l_j} \qquad (i,j = 1...N) \tag{2}$$

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 fórmulas matemáticas das Equações 1 e 2, pode-se constatar que as métricas de desempenho (atraso médio da fila e taxa média de perda de pacotes, respectivamente) 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, empregados no ajuste do espaçamento entre as mesmas.

2.1 Esquema de Encaminhamento Proporcional

O esquema de encaminhamento de pacotes da Figura 1 foi proposto em [8] 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 devidamente classificados, são marcados e encaminhados para as filas correspondentes. O esquema de encaminhamento proporcional é composto basicamente de três partes:

- O escalonador de pacotes que, de acordo com as restrições descritas na Equação 1 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 *Mean Delay Proportional* (MDP) [12], *Backlog-Proportional Rate* (BPR) e *Waiting-Time Priority* (WTP) [7], são exemplos de escalonadores que realizam diferenciação de atraso proporcional.

- 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, porém mais sofisticado, é o RED [13].

- 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 mecanismo de descarte que procurará manter as restrições descritas na Equação 2 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)) [8], são exemplos de descartadores que realizam diferenciação de taxa de perda proporcional.

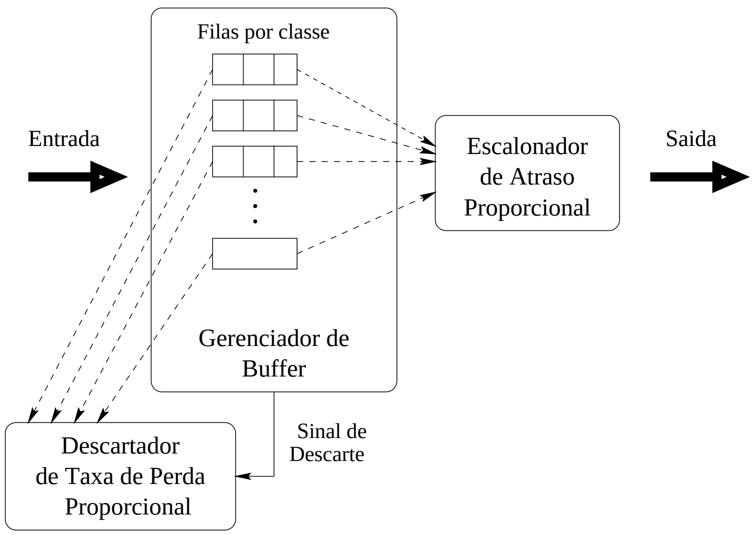


Figura 1: Esquema de Encaminhamento Proporcional.

A fim de avaliar o comportamento do modelo de diferenciação de atraso e de taxa de perda proporcional sob tráfego TCP, o esquema de encaminhamento de pacotes da Figura 1 foi implementado no simulador ns-2 [11], onde foram empregados os algoritmos WTP [7] e PLR(∞) [8] como, escalonador de atraso proporcional e descartador de taxa de perda proporcional, respectivamente. Além destes, também foi implementado o algoritmo de gerenciamento de *buffer* *DropTail-CS*.

2.1.1 Escalonador WTP

No escalonador WTP, 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 determinada como indica a Equação 3.

$$p_i(t) = w_i(t) * sdp_i \tag{3}$$

onde, $w_i(t)$ é o tempo de espera do pacote no tempo t e sdp_i é o Parâmetro 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 em [14]. Posteriormente, Dovrolis et.al. o avaliou como um escalonador de atraso proporcional em um ambiente sem perdas (*buffers* grandes) e em um ambiente com perdas (*buffers* finitos) [7, 8]. As análises realizadas mostram que o escalonador WTP:

- se aproxima bastante do modelo de diferenciação proporcional da Equação 1 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 mais 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 [7].

Outro escalonador proporcional proposto em [12], o MDP (*Mean Delay Proportional*), apresenta uma melhor convergência para o modelo de diferenciação proporcional da Equação 1 do que o escalonador WTP. Este comportamento é observado em curtas escalas de tempo e em condições de baixas e altas cargas de tráfego. Apesar disso, o WTP, devido à simplicidade envolvida em sua implementação, foi escolhido como o escalonador de atraso proporcional a ser utilizado nas simulações apresentadas nesse artigo. As fontes utilizadas nessas simulações, são de longa duração (fontes FTP), ficando como perspectiva para trabalhos futuros o emprego de fontes de curta duração (fontes HTTP).

2.1.2 Descartador PLR(∞)

Este descartador foi originalmente proposto e avaliado por Dovrolis et.al. em [8]. 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 como indica a Equação 4.

$$classe = min \frac{D_i}{A_i * ldp_i} \qquad (l_i = \frac{D_i}{A_i}) \tag{4}$$

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

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* [1], 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 em [8], 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 em [8] mostram que fatores como LDP e DDP especificados, a ocupação média das filas 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 $M \ll \infty$ [8]. Apesar disso, o PLR(∞) foi escolhido como o descartador proporcional a ser utilizado nas simulações apresentadas neste artigo, devido à simplicidade envolvida em sua implementação. Além disso, as conexões de longa duração empregadas nas simulações, quando obtêm o estado de equilíbrio [15, 16], apresentam menos variações na distribuição da carga.

3 Diferenciação Proporcional em Tráfego TCP

O objetivo deste artigo consiste em avaliar o desempenho de um modelo de diferenciação proporcional em tráfego TCP. Para tanto, o esquema de encaminhamento de pacotes descrito na Seção 2.1 foi implementado no simulador ns-2 [11]. Este esquema emprega o WTP como escalonador de atraso proporcional e o PLR(∞) como descartador de taxa de perda proporcional, estando suas implementações de acordo com o proposto por Dovrolis et.al. em [7, 8]. Além destes, foi implementado também o gerenciador de *buffer DropTail-CS*. Sua implementação consistiu em alterações no algoritmo original do *DropTail*, para que passasse a empregar a política de organização de *buffer* de compartilhamento completo. Para tanto, o tamanho máximo das filas é configurado para o espaço total disponível no *buffer*, sendo verificado a cada chegada de pacote, se a somatória dos espaços ocupados por elas não ultrapassa o tamanho total configurado para o *buffer*. Caso isso ocorra, uma sinalização de necessidade de descarte é emitida para o descartador

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

PLR(∞).

Posteriormente à implementação do esquema de encaminhamento, simulações foram realizadas a fim de validar o correto funcionamento dos algoritmos empregados. Esta validação consistiu na repetição de algumas simulações realizadas por Dovrolis et.al. em [7, 8] e na comparação dos resultados obtidos com os apresentados nestas citações. Primeiramente, o escalonador WTP foi validado isoladamente, em um ambiente sem perdas, aplicando-se fontes de tráfego CBR e Pareto (aplicado $\alpha = 1.9$). Em seguida, as mesmas fontes de tráfego foram empregadas na validação do descartador PLR(∞). Por fim, os dois mecanismos foram validados em conjunto aplicando-se unicamente fontes de tráfego Pareto (aplicado $\alpha = 1.5$). Em todas as simulações, os resultados obtidos com tráfego Pareto mostraram-se equivalentes aos originalmente apresentados nas citações.

Tendo-se o esquema sido validado e a constatação do correto funcionamento de suas partes, iniciaram-se as simulações com tráfego TCP. Mais especificamente, as simulações realizadas objetivaram 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 [15, 17, 18, 16] 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. Também é constatado que seu comportamento é fortemente influenciado pelas perdas de pacotes e atrasos ocorridos nas filas em cada nó. Assim, pretende-se avaliar 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. Vale ressaltar que as análises originalmente realizadas em [7, 8] não levam em consideração o tráfego TCP.

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.

3.1 Topologia de Simulação

Todos os casos analisados neste artigo seguem o modelo de topologia descrito na Figura 2. 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.

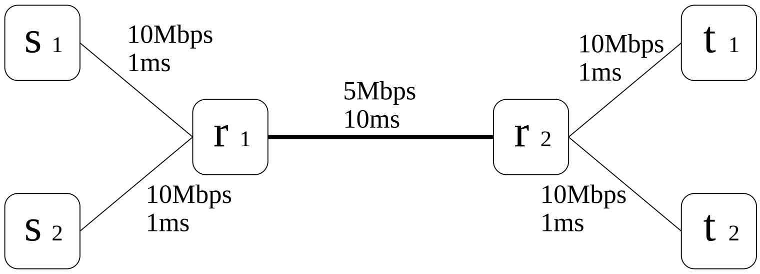


Figura 2: Topologia de simulação.

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 [11], 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 taxas de perda do que a classe 2. Por efeito de simplicidade, nos gráficos adotou-se a representação de que $sdp_2/sdp_1 = SDP$ e $ldp_1/ldp_2 = LDP$ para os SDPs e LDPs empregados 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.

Nas filas, a política de compartilhamento completo é aplicada como política de organização de *buffer* (B), a qual estabelece que pacotes de ambas as filas compartilham o espaço total de B, estando o tamanho dessas, limitadas ao tamanho do *buffer*. O gerenciador de *buffer* utilizado é o *DropTail-CS*.

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 $BW * \tau$ [2], configurar o tamanho de B para pelo menos este valor, permitirá que, antes que pacotes sejam perdidos, a janela de congestionamento do emissor cresça

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

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 [19].

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.

Todos os resultados apresentados neste artigo 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.

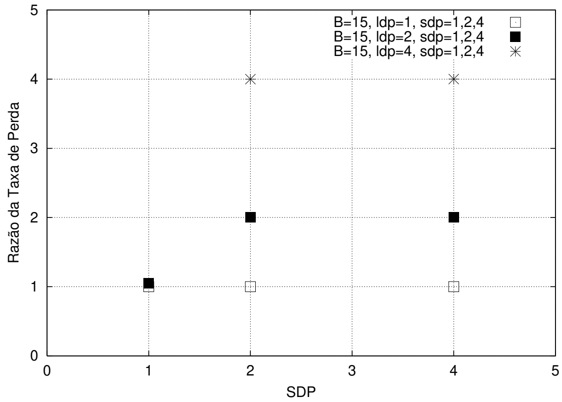
3.2 Avaliação da Taxa de Perda

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.

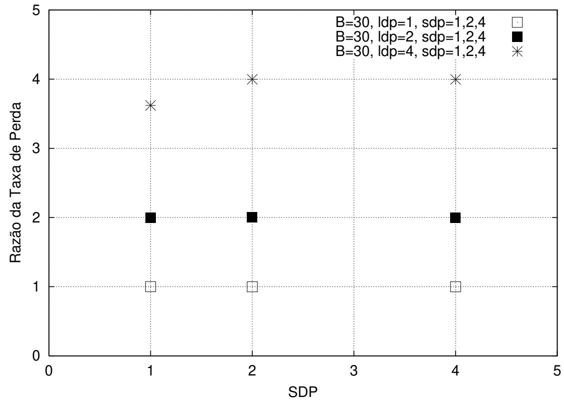
A fim de avaliar as condições de factibilidade e o desempenho do modelo proporcional implementado em relação à taxa de perda, simulações com 1 e 16 fontes de tráfego e diferentes tamanhos de B foram geradas. A Figura 3(a) exibe a razão da taxa de perda entre as duas classes, quando aplicado um B de tamanho 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 ($ldp_1 = 1$, 2 e 4) quando $sdp_2 = 2$ e 4. Para $sdp_2 = 1$, as restrições de proporcionalidade correspondentes a $ldp_1 = 2$ 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_1) 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 de atraso proporcional é 1 ($sdp_2 = 1$), 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 ficou vazia durante todo o período de simulação.

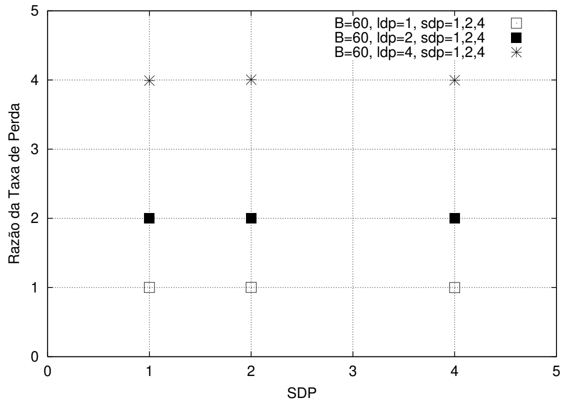
Aumentando-se o tamanho de B para 30, pode-se observar uma melhora no atendimento às restrições de $ldp_1 = 2$ e 4, para $sdp_2 = 1$, Figura 3(b). Esta melhora deve-se ao



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



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

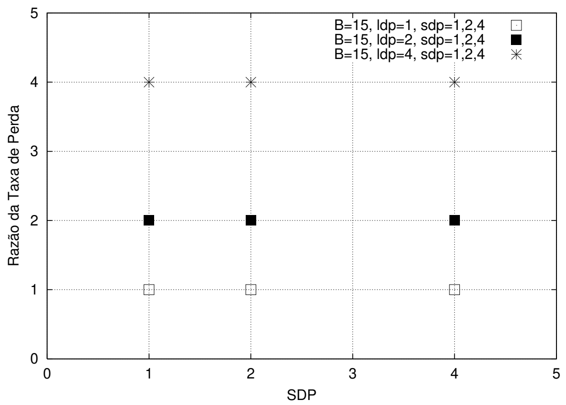


(c) $B = 60, \frac{sdp_2}{sdp_1} = SDP, \frac{ldp_1}{ldp_2} = LDP$

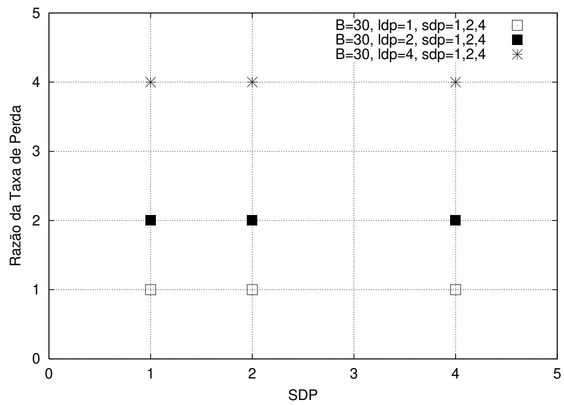
Figura 3: Razão da taxa de perda entre as classes 1 e 2 com uma conexão TCP por classe.

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 garante a proporção estabelecida. Porém, pode ser observado também, que estas condições não foram suficientes para garantir a restrição de $ldp_1 = 4$, quando $sdp_2 = 1$. Por outro lado, para um $B = 60$, Figura 3(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.

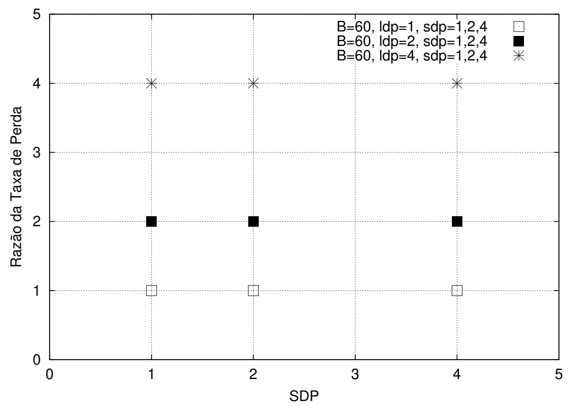
Aplicando-se um maior número de fontes de tráfego TCP nas classes, constatou-se nas simulações realizadas, que o problema de factibilidade do modelo também pôde ser contornado. A Figura 4 exibe os resultados obtidos utilizando-se 16 fontes em cada nó. Com um maior número de fontes, mesmo para $B = 15$, 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) $B = 60$

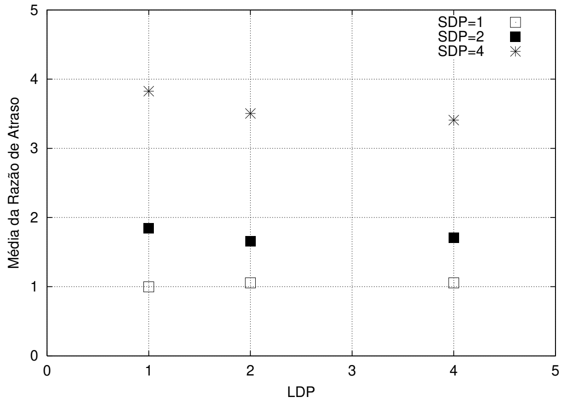
Figura 4: Razão da taxa de perda entre as classes 1 e 2 com 16 conexões TCPs por classe.

3.3 Avaliação do Atraso nas Filas

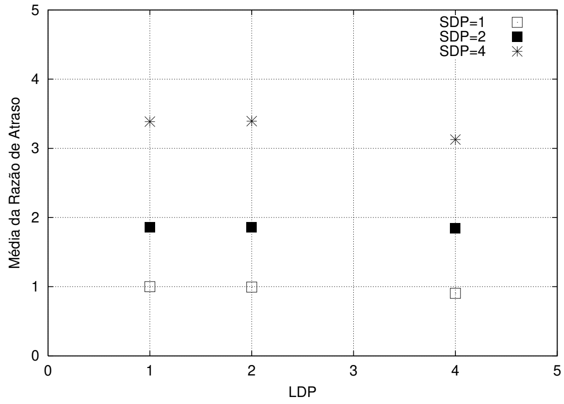
Da mesma forma que na avaliação anterior, as filas das duas classes de serviço, durante todo o tempo de simulação e após um período inicial de *warm-up*, foram monitoradas, sendo gerados arquivos com estatísticas sobre a média de atraso, a variância e a quantidade de pacotes que saíram de cada fila. 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 5(a) exibe a razão da taxa de perda entre as classes 1 e 2, quando aplicado um B de tamanho 15 (em pacotes) e apenas uma fonte de tráfego TCP, em cada classe.

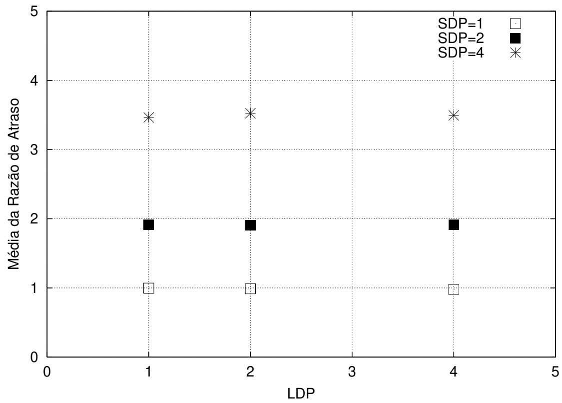
Pode-se observar que para $B = 15$, obtém-se um desvio das restrições de diferenciação de atraso proporcional especificadas em $sdp_2 = 2$ e 4, quando aumenta-se as restrições



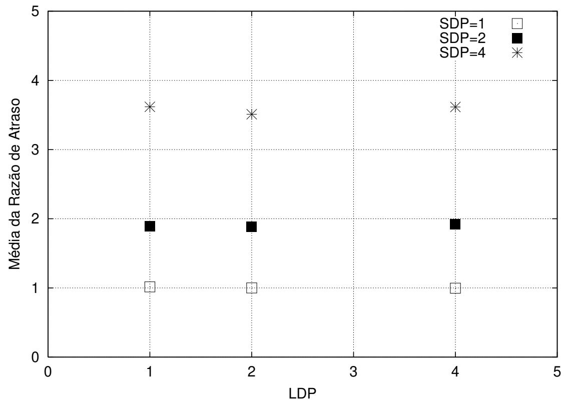
(a) $B = 15$



(b) $B = 30$



(c) $B = 60$



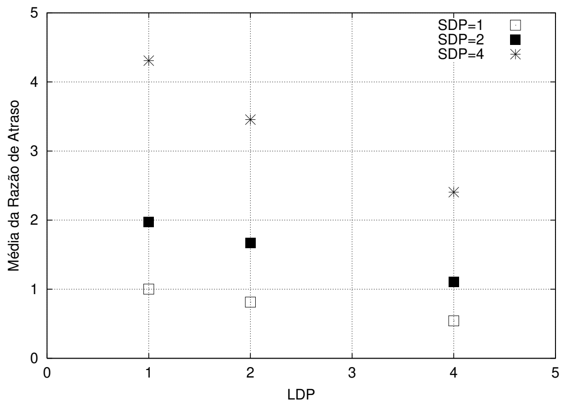
(d) $B = 120$

Figura 5: Atraso Médio entre as classes 1 e 2 com uma conexão TCP por classe.

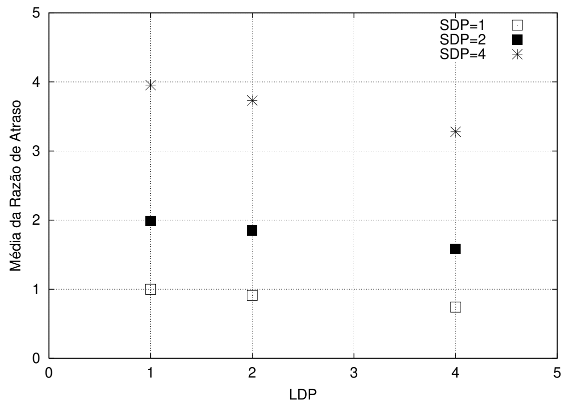
de taxa de perda ($ldp_1 = 2$ e 4). Este comportamento deve-se à factibilidade do modelo implementado, pois pesquisas realizadas em [7] demonstram que o escalonador WTP é factível apenas em altas cargas de tráfego. Assim, este escalonador não pode servir um pacote de uma fila que está vazia, podendo ser este escalonamento necessário à obtenção das restrições proporcionais de atraso especificadas. O que observa-se é que, ao garantir apropriadamente as restrições proporcionais de taxa de perda (Figura 3(a)), não existem pacotes suficientes na fila da classe 1 a serem escalonados, a fim de garantir as restrições proporcionais de atraso estabelecidas. A ocorrência deste fato foi constatada pela observação da quantidade de vezes que a classe 1 fica vazia durante todo o tempo de simulação, o que mostrou um número bastante elevado.

Aumentando-se o tamanho do *buffer*, Figura 5, os resultados apresentaram significativas melhoras para a restrição de atraso igual a 2 (sdp_2). 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 pacotes da classe 1 a serem servidos, a fim de garantir a proporção estabelecida. 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. Pôde-se constatar que a fila da classe 1 permaneceu mais cheia, pela análise da quantidade de vezes que esta fica vazia, a qual apresentou uma redução bastante significativa em relação aos resultados obtidos para $B = 15$. Para a restrição de atraso igual a 4 ($sdp_2 = 4$), observou-se variações na obtenção da proporcionalidade estabelecida.

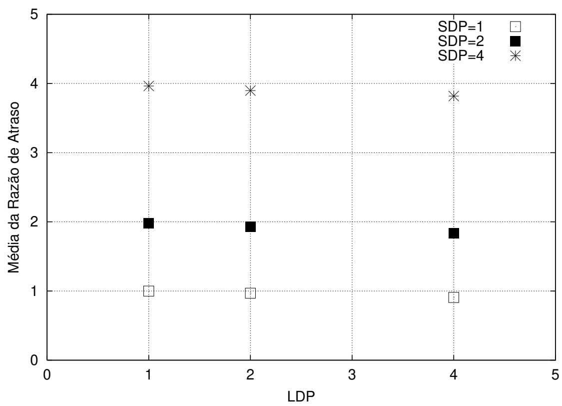
Aplicando-se um maior número de fontes de tráfego TCP nas classes, também pôde-se verificar, para B=15, a ocorrência de uma grande quantidade de vezes que a classe 1 fica vazia. Assim, verifica-se pelo gráfico da Figura 6(a), que aumentando-se a proporção de taxa de perda (ldp_1) - as quais estão sendo garantidas de acordo com o gráfico exibido na Figura 4(a) - os resultados apresentam um considerável desvio da restrição de atraso proporcional estabelecida, conseqüência da não existência de suficientes pacotes da classe 1 no *buffer* para serem servidos.



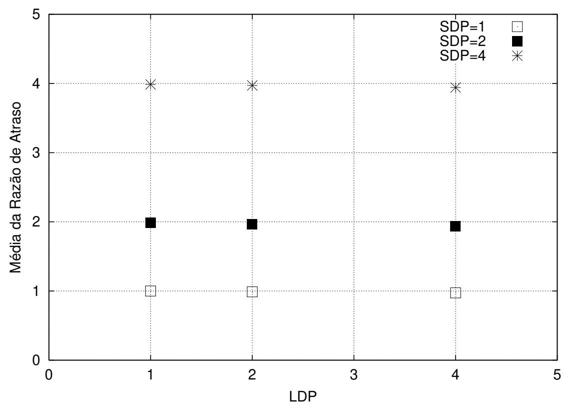
(a) $B = 15$



(b) $B = 30$



(c) $B = 60$



(d) $B = 120$

Figura 6: Atraso Médio entre as classes 1 e 2 com 16 conexões TCPs por classe.

Com uma quantidade maior de *buffer*, Figura 6, observa-se melhoras significativas na obtenção das restrições de atraso estabelecidas no modelo implementado, chegando-se a alcançar condições ideais para $B = 120$, Figura 6(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.

4 Conclusão

O modelo proporcional é uma abordagem promissora para prover uma diferenciação de serviço relativa na Internet. Neste artigo, esse modelo é avaliado para fontes de tráfegos TCP, considerando as métricas de desempenho de atraso e de taxa de perda. 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*. 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. 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, pois a proporcionalidade piora com o aumento do número de conexões. Isto ocorre por que 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 possuem comportamento afetado por perdas e atrasos.

Referências

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

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

[3] J. Heinanen, F. Baker, W. Weiss e J. Wroclawski, “Assured forwarding PHB group”, *Internet RFC 2597*, junho de 1999.

[4] V. Jacobson, K. Nichols e K. Poduri, “An expedited forwarding PHB”, *Internet RFC 2598*, junho de 1999.

[5] C. Dovrolis e D. Stiliadis, “Relative differentiated services in the internet: Issues e mechanisms”, in *Proceedings of ACM SIGMETRICS’99*, maio de 1999.

[6] C. Dovrolis e P. Ramanathan, “A case for relative differentiation services and the proportional differentiation model”, *IEEE Network*, vol. 13, no. 5, setembro de 1999.

[7] C. Dovrolis, D. Stiliadis e P. Ramanathan, “Proportional differentiated services: Delay differentiation and packet scheduling”, in *Proceedings of ACM SIGCOMM’99*, setembro de 1999.

[8] C. Dovrolis e P. Ramanathan, “Proportional differentiated services, part ii: Loss rate differentiation and packet dropping”, in *Proceedings of IWQoS’00*, junho de 2000.

[9] A. Odlyzko, “Paris metro pricing: The minimalist differentiated services solution”, in *Proceedings of the International Workshop on QoS - IWQoS’99*, junho de 1999.

[10] J. C. R. Bennett e H. Zhang, "Hierarchical packet fair queueing algorithms", *IEEE/ACM Transactions on Networking*, vol. 5, no. 5, pp. 675–689, outubro de 1997.

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

[12] T. Nandagopal, N. Venkitaraman, R. Sivakumar e V. Bharghavan, "Delay differentiation and adaptation in core stateless networks", in *Proceedings of IEEE INFOCOM'00*, março de 2000.

[13] S. Floyd e V. Jacobson, "Random early detection gateways for congestion avoidance", *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, pp. 397–413, agosto de 1993.

[14] L. Kleinrock, "A delay dependent queue discipline", *Journal of the ACM*, vol. 14, no. 2, no. 2, pp. 242–261, 1967.

[15] V. Jacobson, "Congestion avoidance and control", in *Proceedings of the SIGCOMM'88*, agosto de 1988.

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

[17] K. Fall e S. Floyd, "Simulations-based comparisons of Tahoe, Reno and SACK TCP", *IEEE Computer Communications Magazine*, vol. 26, no. 3, julho de 1996.

[18] W. R. Stevens, "TCP slow start, congestion avoidance, fast retransmit and fast revovery algorithms", in *Internet RFC 2001*, janeiro de 1997.

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