Avaliando Estratégias de *Push-out* no Descarte Seletivo de Pacotes de Tráfego TCP de Curta e Longa Duração *

Kleber V. Cardoso(1) Saulo V. de Vasconcellos(1)

José F. de Rezende(1) Nelson L. S. da Fonseca(2)

(1)Grupo de Teleinformática e Automação
COPPE/Programa de Engenharia Elétrica
Universidade Federal do Rio de Janeiro
{kleber,vaz,rezende}@gta.ufrj.br

(2)Instituto de Computação
Universidade de Campinas
nfonseca@ic.unicamp.br

Resumo

Este artigo faz uma avaliação de desempenho de mecanismos de descarte seletivo de pacotes que utilizam técnicas de *push-out* para a diferenciação de serviços na Internet. Essa avaliação foi feita utilizando-se um modelo de geração de tráfego TCP composto de múltiplos fluxos (ou clientes). Nesse modelo, a taxa de chegada de clientes e a quantidade de informação a ser transferida por cada cliente determinam uma carga específica na rede. O objetivo desse estudo é avaliar o grau de diferenciação e a eficiência dos mecanismos de descarte seletivo no aproveitamento da banda passante disponível em diferentes condições de carga. Observamos que as políticas de remoção (*push-out*), quando associadas a mecanismos de gerenciamento de *buffer* do tipo RED não acrescentam vantagens. Foi observado ainda que mecanismos de descarte seletivo do tipo *loss conserving* apresentam menor diferenciação entre classes de serviço.

Palavras-chave: Internet, Qualidade de Serviço (QoS), Avaliação de Desempenho.

Abstract

This paper makes a performance analysis of selective packet discard mechanisms that use push-out techniques to service differentiation in the Internet. Such evaluation makes use of a TCP traffic generation model composed of multiple flows (clients). The arrival rate and amount of information transferred by each client determine the network load. The objective of this study is to evaluate the degree of differentiation and the efficiency of these mechanisms in the available bandwidth use under different load conditions. We observed that push-out policies when associated to RED-like buffer management mechanisms do not add advantages. Addicionaly, it was observed that the selective discard mechanisms of the type loss conserving show lower differentiation between service classes.

Keywords: *Internet, Quality of Service (QoS), Performance Evaluation.*

*Este trabalho foi realizado com recursos da FUJB, CNPq, CAPES, COFECUB e FAPERJ.

1 Introdução

O modelo de serviços diferenciados na Internet (DiffServ) [1] tem recebido grande atenção por parte da comunidade científica. Um dos mecanismos de controle de tráfego, nesse modelo, é o descarte seletivo usado para discriminar pacotes pertencentes a diferentes classes de serviço. O mecanismo de descarte seletivo descarta pacotes em situações de congestionamento de acordo com o nível de prioridade dos mesmos. A perda de pacotes devido a descarte pode ser tanto devido a políticas de gerenciamento de *buffer* quanto a políticas de escalonamento de pacotes. O mecanismo de descarte seletivo diz respeito única e exclusivamente à administração do espaço de armazenamento, ou seja, o mecanismo de descarte seletivo determina qual pacote deve ser descartado para liberar espaço para o pacote recém-chegado a fila do *buffer*. O descarte seletivo discrimina pacotes sem desordená-los, podendo ser usado no tratamento de pacotes do serviço Assegurado [2], definido no modelo DiffServ. Além disso, pela característica de não desordenamento, essas disciplinas podem ser aplicadas na diferenciação de pacotes pertencentes a um mesmo fluxo.

As políticas de descarte seletivo têm uma influência significativa no desempenho obtido por clientes de diferentes classes de serviço. Quando esses clientes são fluxos TCP, ou seja, fluxos sensíveis ao congestionamento, as distribuições de perdas (taxa de perda e espaçamento entre as perdas) imposta por tais políticas determinam a alocação da banda passante dos enlaces de gargalo. Um mecanismo que tem despertado grande interesse no caso de múltiplas classes de usuários é o RED-n (RED com múltiplos níveis de prioridade) ou RED Multiclasse [3]. Na verdade, o RED-n é um mecanismo de gerenciamento de *buffer* que sinaliza quando um descarte deve ser realizado, porém não determina qual pacote deve ser descartado. Por questões de facilidade de implementação, o pacote que chega é o candidato a descarte. Uma outra possibilidade de se definir políticas de descarte é permitir que pacotes já enfileirados sejam removidos ao invés de se descartar pacotes no momento de sua chegada. Tal política é chamada de *push-out*. Até recentemente, as políticas de remoção (*push-out*) eram consideradas de difícil implementação por implicarem numa reorganização da fila. No entanto, com as novas tecnologias usadas no projeto de comutadores de alta velocidade, baseadas em ASICs, constata-se que uma grande parte das políticas de remoção pode ser implementada sem nenhum impacto significativo no desempenho [4, 5].

Várias políticas de remoção e de gerenciamento de *buffer* foram propostas e analisadas na literatura. Na maioria dos trabalhos, o processo de chegada de pacotes segue uma distribuição exponencial ou Markoviana [6, 7, 8, 9]. Fonseca *et al.* [10] utilizam um processo de dependência de longa duração. Kurose *et al.* [11] estudam o comportamento das perdas num tráfego de tempo-real. Nenhum desses trabalhos leva em conta as características do tráfego comumente encontrado na Internet. Este artigo tem como objetivo comparar políticas de descarte em termos da diferenciação fornecida, usando fluxos TCP de curta e longa duração pertencentes a classes de serviço distintas. Comparamos a eficiência de *push-out* com a eficiência de políticas que não descartam pacotes enfileirados. Investiga-se, também, o uso integrado de *push-out* com políticas RED-n. Para a geração deste tráfego foi utilizado um modelo que permite variar a carga na rede. As características do tráfego na Internet são capturadas através da escolha apropriada dos processos de chegada de clientes (fluxos TCP), e do volume de informação a ser transferido por cada cliente desse modelo.

Este artigo apresenta-se da seguinte maneira. A seção 2 discute e classifica os mecanismos de descarte seletivo de pacotes, de acordo com as políticas de remoção e de gerenciamento de *buffer*. A seção 3 descreve o modelo utilizado na avaliação de desempenho, apresenta os resultados de simulação obtidos, e realiza uma análise detalhada destes resultados. Finalmente, na seção 4, são apresentadas as conclusões e as perspectivas deste trabalho.

2 Descarte seletivo de pacotes

Em redes ATM, as políticas de descarte de pacotes foram concebidas para contornar o problema de baixo desempenho observado devido a fragmentação dos pacotes em células, onde a perda de apenas uma das células de um pacote exige a retransmissão de todas elas. Assim, várias políticas de descarte [12, 13, 14, 15, 16] foram projetadas para aliviar o congestionamento e aumentar a utilização da rede. Uma vez que uma célula de um pacote for perdida (ou corrompida), todas as células seguintes do mesmo pacote devem ser selecionadas para descarte. Os objetivos são maximizar a utilização e melhorar o desempenho das aplicações, tentando atingir uma "melhor" taxa de perda de mensagens através do agrupamento das células perdidas.

Em redes IP, as políticas de descarte funcionam de forma diferente. O problema de fragmentação não é uma questão importante e os objetivos são outros. As políticas de descarte de pacotes para redes IP têm se concentrado na redução do comprimento das filas, de forma a diminuir o atraso de pacotes com requisitos de tempo-real, evitar problemas de sincronização global do TCP [17] e evitar que um ou mais fluxos monopolizem o espaço de *buffer* (*lock-out*) [18]. Mais recentemente, as políticas de descarte têm sido usadas para fornecer algum tipo de garantia através da discriminação dos pacotes.

Descarte seletivo de pacotes engloba dois tipos de políticas: as políticas de gerenciamento de *buffers*, que determinam quando algum pacote deve ser descartado, e as políticas de remoção, que determinam qual pacote deve ser descartado, incluindo não apenas aquele que chega, mas todos os pacotes em espera por transmissão.

2.1 Políticas de gerenciamento de *buffers*

No Serviço Assegurado, o gerenciamento de *buffers* é normalmente realizado pela adoção de algoritmos RED: um para cada nível de preferência de descarte. Cada algoritmo RED é configurado de maneira mais agressiva que aquele que trata dos pacotes do nível de preferência exatamente acima, visando o descarte prioritário de pacotes com um maior valor de preferência de descarte. O objetivo é reduzir os efeitos do congestionamento antes da necessidade de descarte de pacotes de menor preferência. Este mecanismo é conhecido como RED-n (RED com múltiplos níveis de prioridade de descarte). Em [3], os autores apresentam quatro categorias gerais para políticas RED quando múltiplos níveis de preferência de descarte, ou múltiplas cores, são utilizados na marcação de pacotes. Essas categorias surgem da maneira pela qual podemos calcular o tamanho médio da fila e estabelecer os limiares de descarte do algoritmo RED. Na categoria Múltiplas Médias/Múltiplos Limiares, um tamanho médio da fila é calculado para cada nível de preferência, onde o número de pacotes na fila de um determinado nível é igual à soma dos pacotes desse nível e, se existirem, dos níveis inferiores. Além disso, para cada nível de preferência de descarte temos diferentes limiares de descarte associados. Por exemplo, a fila RIO (RED com *IN* e *OUT*) [19] pertence a essa categoria. O tamanho médio da fila para pacotes *IN* (alta prioridade) é calculado utilizando apenas o número de pacotes *IN*, enquanto que o tamanho médio da fila para pacotes *OUT* (baixa prioridade) é calculado utilizando o número de pacotes *IN+OUT*. Diferentes limiares de descarte são definidos para cada um destes níveis. Com essa categoria, podemos ser tão agressivos quanto desejarmos no descarte de pacotes com maiores valores de precedência de descarte. Por questões de simplicidade de implementação, o pacote a ser descartado é sempre aquele que chega.

Várias outras políticas foram propostas na literatura [20, 9, 21, 22, 23, 24, 25]. Choudhury e Hahne [24, 25] propõem uma política na qual o limiar para o número de pacotes enfileirados de uma certa classe é dinâmico e depende do tamanho do *buffer* ainda disponível. Em [9], os autores identificam e avaliam políticas de gerenciamento de *buffer* ditas protetoras. Tais políticas garantem um alto desempenho aos pacotes de alta prioridade, independentemente da intensidade e do padrão de chegada do tráfego de menor prioridade. Para isso, tais políticas devem respeitar uma invariante que estabelece que o número de *buffers* disponíveis para pacotes de mais alta prioridade deve ser pelo menos igual aquele de um sistema de referência que trata somente pacotes de mais alta prioridade. Os autores propõem ainda algumas políticas que visam melhorar o serviço fornecido aos pacotes de menor prioridade, limitando o serviço aos pacotes de maior prioridade ao mínimo exigido para respeitar essa invariante e permanecer protetoras. Nesse caso, somente os pacotes de menor prioridade que podem provocar uma violação dessa invariante são descartados.

2.2 Políticas de remoção

Ao considerarmos os mecanismos de remoção de pacotes da fila, diferentes políticas podem ser aplicadas: *First-In-First-Drop* (FIFD), *Last-In-First-Drop* (LIFD), *Modified-FIFD* (M-FIFD) e *Random* (RAND). Tipicamente, essas políticas são utilizadas quando há mais de uma classe de tráfego compartilhando um *buffer*. No entanto, elas poderiam ser aplicadas quando apenas uma classe é utilizada. Como o enfoque deste trabalho é o descarte seletivo de pacotes, consideraremos apenas o primeiro caso. Sendo assim, as políticas de remoção determinam qual pacote será descartado quando um pacote de mais alta prioridade chega e o gerenciamento de *buffer* determina que um pacote deve ser descartado. Na política FIFD, o pacote de menor prioridade mais próximo à cabeça da fila será descartado. Na política LIFD, o pacote de menor prioridade mais próximo do final da fila será descartado. A escolha de uma dessas políticas tem implicações na distribuição das perdas e, consequentemente, no desempenho alcançado. O descarte mais próximo à cabeça da fila faz com que o receptor detecte uma perda mais rapidamente, além de aumentar a probabilidade que um pacote com menor prioridade seja encontrado na fila ao chegar um pacote de maior prioridade [9]. Desta forma, a política FIFD fornece uma menor taxa de perda para pacotes de maior prioridade. Outras políticas, tais como M-FIFD, permitem que um pacote de menor prioridade retire um pacote de mesma classe já enfileirado. Como explicado anteriormente, esse tipo de política aumenta ainda mais a probabilidade que um pacote com menor prioridade seja encontrado na fila na chegada de um próximo pacote de mais alta prioridade. A política RAND escolhe aleatoriamente um pacote de menor prioridade a ser descartado. Não existe nenhuma vantagem aparente na utilização dessa política. Foram realizadas simulações com as políticas FIFD, M-FIFD, LIFD e RAND, a fim de comparar o seu desempenho em relação à banda passante, taxa de perda e atraso na fila. Os resultados obtidos mostraram que não há diferença significativa entre as políticas.

3 Simulação

Neste trabalho, três diferentes mecanismos de descarte seletivo são avaliados e comparados: PO (*complete sharing push-out*), PRIO (*push-out* RIO) e RIO. O primeiro mecanismo (PO) utiliza uma política *loss conserving*. Essa política utiliza o máximo possível dos *buffers* disponíveis, evitando descartar quaisquer pacotes. O segundo mecanismo (PRIO) utiliza o algoritmo RIO, como descrito anteriormente, para sinalizar a necessidade de descarte de um pacote. Esse último mecanismo (PRIO) é do tipo *non-loss conserving*. Os mecanismos com *push-out* são avaliados com diferentes políticas de remoção (FIFD, LIFD e M-FIFD).

Para compararmos esses mecanismos e suas políticas, utilizaremos métricas de desempenho que capturam o ponto de vista tanto do provedor do serviço como do usuário. Do ponto de vista do provedor, a primeira métrica é a eficiência no aproveitamento da banda passante, ou seja, a utilização da banda passante. Dada uma certa carga na rede, uma baixa utilização do enlace pode ser inaceitável. Além disso, se o provedor do serviço utiliza uma tarifação baseada na banda passante utilizada pelo cliente, ao invés de uma tarifação fixa (*flat rate*), seria desejável otimizar essa métrica de desempenho, o que aumentaria a sua arrecadação. Uma outra métrica importante é a taxa de perda de pacotes. A perda de pacotes representa banda passante e *buffers* desperdiçados nos enlaces anteriores. Do ponto de vista do usuário, a métrica mais importante é a taxa efetiva (*goodput*) média, ou seja, a banda passante realmente usada do tráfego gerado. A segunda métrica diz respeito ao grau de diferenciação fornecido pelos mecanismos avaliados. Ela corresponde à razão entre as taxas obtidas por uma classe de serviço e pela classe exatamente inferior, o que representa o espaçamento entre essas classes. O usuário que utiliza uma classe superior, e paga mais por isso, deseja obter um serviço de melhor qualidade do que aquele que paga menos. No entanto, isso não significa que devemos ignorar o desempenho oferecido aos pacotes de menor prioridade, uma vez que os pacotes de mais alta prioridade estão protegidos.

As simulações foram realizadas em duas etapas. Isso foi feito com o intuito de isolar os efeitos no desempenho introduzidos pelo gerador de tráfego, dos efeitos envolvendo os mecanismos de descarte seletivo utilizados. Numa primeira fase, estudamos a influência de parâmetros associados ao modelo de tráfego TCP, utilizando o mecanismo de gerenciamento de *buffer DropTail* (DT). Em seguida, avaliamos as diferentes políticas de gerenciamento de *buffer* e de remoção descritos anteriormente.

3.1 Topologia e modelo de tráfego

Neste estudo de simulação foi utilizada a topologia da figura 1. O enlace de gargalo corresponde ao enlace entre os nós $r1$ e $r2$. Este enlace, de capacidade C, é compartilhado por duas classes de tráfego. A taxa de chegada de clientes da classe i ($i \in 1, 2$) é dado por λ_i, ou seja, a cada $1/\lambda_i$ chega um novo microfluxo da classe i por um enlace de capacidade M de acesso ao nó s_i. Logo após o nó de agregação s_i, o tráfego originado dos vários clientes de uma mesma classe representa uma carga ρ_i dada por $\rho_i = \lambda_i * L_i/C$, onde L_i é o volume médio de informação transferido por cada cliente (fluxo TCP). Esta carga atravessa o enlace de gargalo em direção ao nó d_i. A condição de estabilidade do sistema é dada por $\rho = \rho_1 + \rho_2 < 1$. Para aumentar ou diminuir a carga basta modificar a taxa de chegada de novos clientes. A quantidade de clientes simultâneos pode ser alterada, modificando-se o tamanho médio da informação transferida por cada cliente (L).

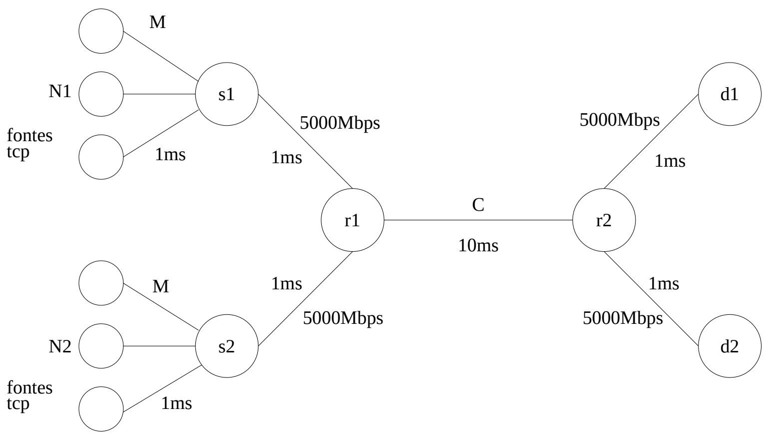


Figura 1: Topologia.

Neste modelo, a taxa de chegada dos clientes, assim como o tamanho de uma transferência, são especificados de acordo com as características do tráfego desejado. Por exemplo, num tráfego *web*, o primeiro parâmetro define a dinâmica agregada dos usuários no pedido de páginas. De acordo com [26], o intervalo entre os pedidos de páginas de um tráfego *web* agregado segue uma distribuição exponencial com uma média de 0.871 segundos. O segundo parâmetro determina a "duração" de uma conexão TCP. No tráfego *web*, temos uma maioria de conexões de curta duração, cada uma transferindo um ou mais objetos de uma página [1]. Também de acordo com [26], esse parâmetro segue uma distribuição log-normal com média de 4827 bytes. Outros estudos do tráfego *web* mostram que o tamanho das transferências segue uma distribuição pareto com decrescimento lento [27]. Conexões TCP de longa duração são, na sua maioria, utilizadas por aplicações FTP, mas também estão presentes em tráfego *web*, podendo atingir valores acima de 5 MB. Nas simulações executadas, verificamos a influência desse parâmetro, utilizando diferentes tamanhos de transferência, de acordo com o tipo de tráfego a ser modelado. A implementação do TCP utilizado é Reno, por ser a versão mais utilizada na Internet.

Na topologia utilizada, a capacidade do enlace de acesso (M) determina a taxa de pico (ou taxa máxima) na qual rajadas emitidas pelas fontes chegam aos nós de agregação s_i. Quando a capacidade deste enlace de acesso é maior do que a capacidade do enlace de gargalo, a taxa de pico pode ser considerada como infinita. Caso contrário, ela limita a banda passante conseguida por uma classe de serviço. Os nós s_i são ligados ao nó de gargalo $r1$ por um enlace de banda passante suficientemente grande para não ocasionar perdas. Os nós $r1$ e $r2$ implementam os seguintes mecanismos de gerenciamento de filas: PO, PRIO, RIO e DT. O tamanho dos *buffers* nesses nós é igual ao produto banda passante-latência, de modo que os clientes possam atingir suas vazões máximas.

[1] HTTP 1.0 utiliza uma conexão TCP por objeto. o HTTP v1.1 reutiliza conexões TCP.

3.2 Estudo da carga na rede

Inicialmente, foram feitas simulações para realizar uma calibragem do ρ em relação a carga medida na rede. No modelo utilizado, o ρ corresponde a uma aproximação da carga imposta na rede. No entanto, ao se utilizar um protocolo que realize controle de congestionamento, como no caso do TCP, essa relação poderia não ser direta. Isso ocorreria porque a taxa de serviço passa a não ser dada diretamente pela razão $1/L/C$, uma vez que a fonte regula a taxa de transmissão de acordo com a ocorrência de perdas. Dessa forma, foram realizadas simulações para identificar a relação efetiva entre carga média da rede e ρ. A metodologia utilizada baseou-se em [28]. Nessas simulações foram utilizadas fontes de tráfego com transferências de tamanho fixo igual a 256KBytes (tráfego FTP), e também, seguindo uma distribuição log-normal com média igual a 4827 bytes (tráfego *web*). Esse último modela o tamanho de transferências realizadas por um tráfego HTTP [26]. A figura 2 apresenta os resultados obtidos.

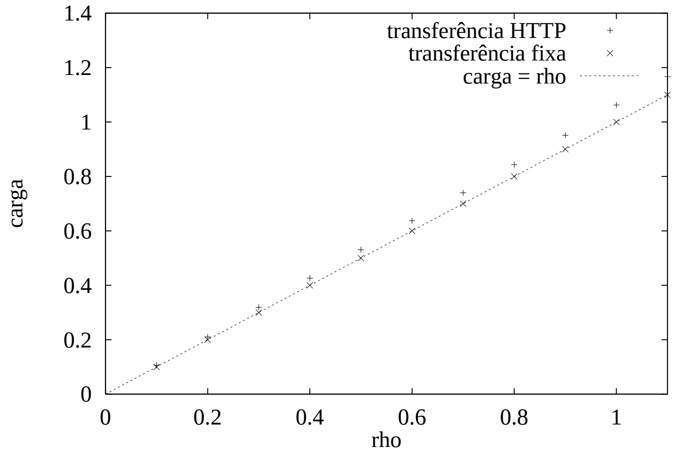
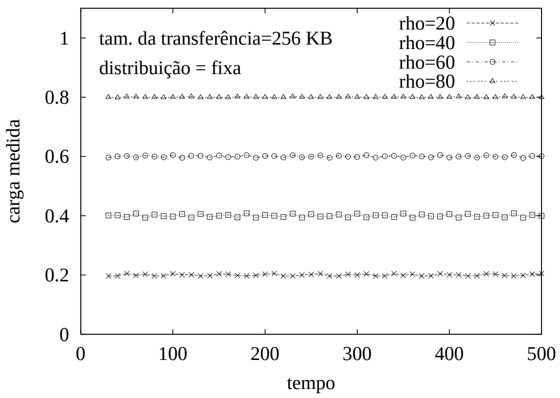


Figura 2: Relação entre carga e ρ.

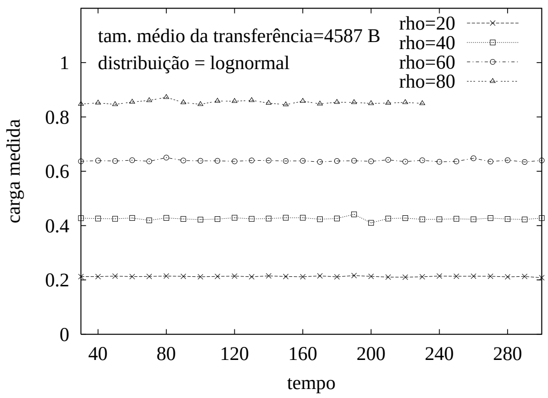
A partir do conjunto de pontos encontrados, foram obtidas as funções que melhor representam os dados, utilizando-se o método dos mínimos quadrados (m.m.q.) não linear. No caso das transferências de tamanho fixo, a relação é $carga = 1,00001 * \rho - 4,50\text{E-}6$ e no caso de tráfego HTTP, $carga = 1,06581 * \rho - 2,07\text{E-}3$. A figura apresenta também a reta $carga = \rho$. As figuras 3(a) e 3(b) mostram a variação da carga na rede calculada a intervalos constantes iguais a 10 segundos. Em escalas de tempo menores, a carga média apresenta uma maior variação. Essas curvas demonstram que a carga na rede se aproxima bastante da carga imposta por ρ, o que demonstra a acurácia do nosso modelo. Desta forma, utilizaremos doravante a relação $carga = \rho$.

3.3 Validade dos resultados obtidos

Devido a característica aleatória do nosso modelo de simulação, principalmente do modelo de tráfego adotado, tivemos a preocupação em observar que todos os resultados obtidos tivessem sempre um intervalo de confiança associado. Dois métodos foram empregados para esse fim. Num primeiro método, as simulações são executadas uma única vez e perduram até que as métricas de desempenho de interesse atinjam valores médios com intervalos



(a) tamanho fixo



(b) http

Figura 3: Variação da carga medida na rede no tempo.

de confiança 95% dentro de uma faixa de mais ou menos 10% deste valor médio. Para diminuir a correlação entre as observações dessa única amostra, utilizamos uma técnica conhecida como média por janela (*batch means*) [29]. No segundo método, utilizamos a técnica de replicações independentes, onde várias simulações são executadas até que os intervalos de confiança desejados sejam atingidos. Nesse método, cada simulação é executada até que um número N de clientes de ambas as classes de serviço tenham completado suas transferências. Para eliminar os resultados do estado transiente, apenas os clientes cujas transferências se iniciam após 50 segundos de simulação são considerados. Por exemplo, quando a métrica de desempenho medida é a banda passante obtida por uma classe de serviço, calculamos o valor médio de todas as transferências realizadas por clientes dessa classe. A mesma metodologia foi empregada para as demais métricas de desempenho.

3.4 Influência do tamanho da transferência

Nas primeiras simulações, verificamos a influência do tamanho de uma transferência no desempenho obtido, em termos de banda passante, pelas duas classes de serviço em diferentes condições de carga. As curvas da figura 4 mostram o desempenho agregado obtido por ambas as classes de serviço em função do tamanho médio de uma transferência (em bytes) para diferentes valores de carga total (ρ), onde $\rho_1 = \rho_2 = \rho/2$. Nos resultados mostrados abaixo, a banda obtida é normalizada com relação a banda passante do enlace de gargalo. Com o intuito de isolar os efeitos do descarte seletivo de pacotes, utilizamos nessas simulações a disciplina *DropTail* (DT), a qual não exerce nenhuma diferenciação entre as classes. Para permitir uma variação nos tamanhos de transferência (L), esse parâmetro segue uma distribuição exponencial de média L.

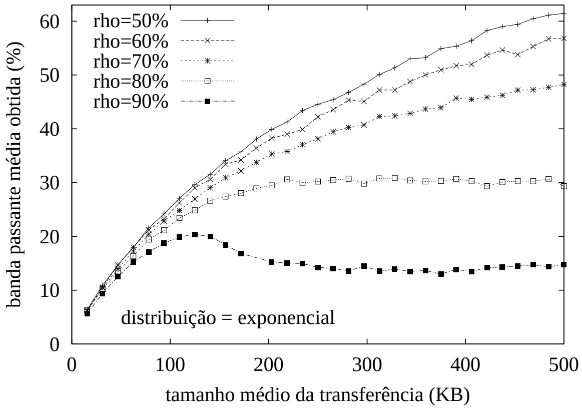
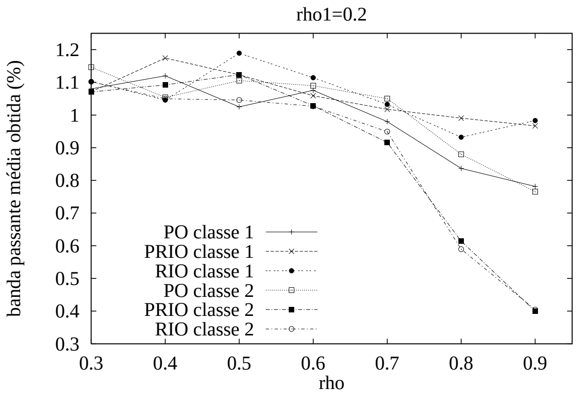


Figura 4: Banda passante obtida em função do tamanho médio de transferência.

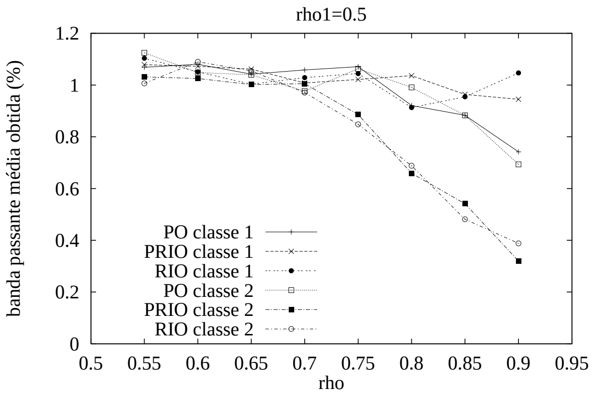
As curvas mostram que a banda passante obtida decresce com a diminuição do tamanho médio de uma transferência. Isso acontece porque conexões TCP de curta duração operam praticamente no período transiente do mecanismo de controle de congestionamento (fase *slow-start*), não conseguindo atingir o seu estado estacionário (fase de *congestion avoidance*). Devido a presença de rajadas de perdas nessa fase, o protocolo não atinge a fase de crescimento linear da janela de congestionamento antes de algumas temporizações, o que contribui para um baixo desempenho médio do protocolo.

3.5 Conexões de curta duração

Em seguida, avaliamos a influência dos mecanismos de descarte seletivo em transferências de curta duração, mantendo a carga da classe 1 fixa e aumentando a carga total até 90%. Através dessas simulações, pudemos identificar a sensibilidade dos mecanismos com relação ao aumento da carga. Um resultado desejável seria que o desempenho da classe mais prioritária (classe 1) permanecesse constante com o aumento da carga total. Ou seja, que o mecanismo de descarte seletivo fornecesse um isolamento entre as classes. As figuras 5(a) e 5(b) apresentam os gráficos da banda passante média obtida pelos clientes das classes 1 e 2 com o aumento da carga, para ρ_1 (carga da classe 1) igual a 20% e 50%, respectivamente. Novamente, a banda obtida por cada classe é normalizada com relação a banda passante do enlace de gargalo.



(a) $\rho_1 = 20\%$



(b) $\rho_1 = 50\%$

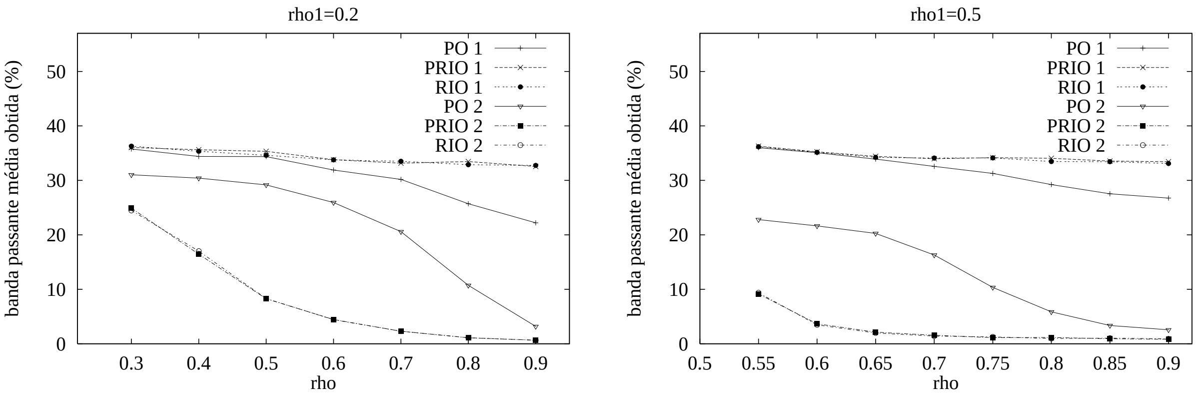
Figura 5: Banda passante das classes 1 e 2 em função da carga na rede.

Como pode ser observado nas figuras 5(a) e 5(b), o desempenho permanece quase constante com o aumento da carga para as políticas PRIO e RIO. Uma outra constatação é que o mecanismo de remoção (*push-out*) da disciplina PRIO não oferece nenhum ganho no desempenho, tanto para a classe 1 como para a classe 2. O mecanismo mais sensível ao aumento da carga é o PO. Além disso, essa política, apesar de oferecer um melhor desempenho para a classe 2, não diferencia significativamente as classes. Podemos ainda observar uma queda brusca na banda passante da classe 2 para as políticas PRIO e RIO, a medida que a carga total aumenta. Isso faz com que a banda média obtida por tais clientes seja muito baixa, o que poderia ser explicado pelo fato das conexões serem de curta duração. Para permitir uma comparação mais aprofundada entre as políticas estudadas, realizamos simulações com conexões de longa duração, que são mostradas a seguir.

3.6 Conexões de longa duração

Neste cenário, utilizamos conexões cujo tamanho segue uma distribuição exponencial com média de 512KBytes. As cargas impostas por cada classe são distribuídas da mesma forma que nas simulações anteriores. As figuras 6(a) e 6(b) mostram a banda passante média obtida pelos clientes das classes 1 e 2 com o aumento da carga, para ρ_1 (carga da classe 1) igual a 20% e 50%, respectivamente.

Para analisar os resultados em termos de banda passante média obtida foi necessário avaliar métricas de desempenho que têm um forte impacto no desempenho do protocolo TCP. Essas métricas são: taxa de perda, atraso e número médio de temporizações. Quanto maiores forem essas métricas menor é o desempenho do protocolo TCP. Os gráficos das figuras 7(a) e 7(b), 8(a) e 8(b), e 9(a) e 9(b) mostram essas métricas, para ambas as classes, em função do aumento da carga na rede.



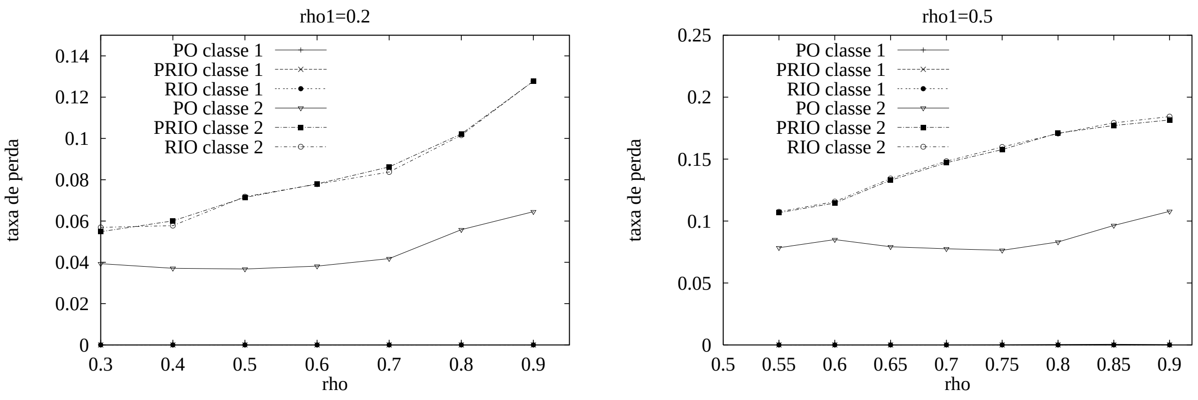
(a) $\rho_1 = 20\%$

(b) $\rho_1 = 50\%$

Figura 6: Banda passante da classes 1 e 2 em função da carga na rede.

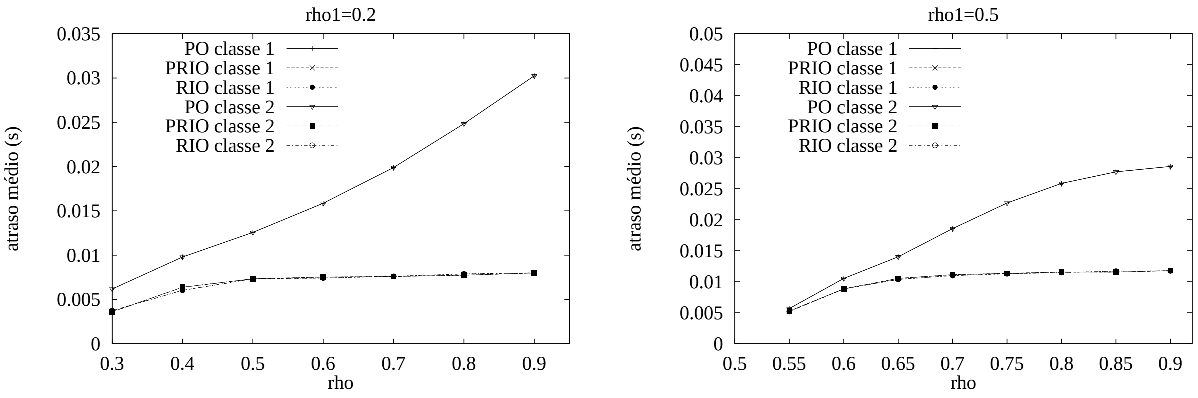
Como pode ser observado, as políticas de descarte baseadas em RED (RIO e PRIO) oferecem uma melhor banda passante para a classe mais prioritária. Esse fato se deve principalmente ao aumento significativo da taxa de perda da classe menos prioritária (figuras 7(a) e 7(b)), característico das políticas de descarte prévio (*early discard*) com múltiplas classes. No entanto, ocorre uma pequena queda na banda passante da classe 1 devido a um pequeno aumento do retardo, uma vez que não ocorrem perdas e nem temporizações (figuras 9(a) e 9(b)). A partir das figuras 6(a) e 6(b) é possível também observar que a política PO não diferencia significativamente as classes, pois a diferença entre as taxas de perda das classes permanece menor que as apresentadas pelo RIO e pelo PRIO, conforme observado nas figuras 7(a) e 7(b). Isso ocorre porque essa política somente efetua descartes com o transbordo do *buffer* do enlace de gargalo. A diferenciação de classes do PO não é observada nas figuras 5(a) e 5(b) porque o tráfego de conexões apenas de curta duração não permite que o TCP atinja seu estado estacionário. A maior queda na banda de ambas as classes para a política PO se deve ao aumento significativo do atraso na fila de acordo com o gráfico das figuras 8(a) e 8(b). Por outro lado, as políticas PRIO e RIO mantém um baixo atraso nas filas, característico de políticas de descarte prévio.

Além disso, observamos que RIO e PRIO não apresentam diferenças significativas de desempenho. Um dos motivos para esse fato é que ambos são baseados em um mecanismo de descarte prévio que sobrepõem a função principal das políticas de remoção: liberar espaço de um pacote mais antigo para dar lugar a um mais recente. A diferença é que mecanismos com descarte prévio (como RIO) já realizam essa tarefa sem acrescentar atraso às filas. Outro motivo é que a característica das políticas de remoção de poder alterar a distribuição das perdas é imperceptível quando combinada com um mecanismo de descarte prévio, porque a ocupação média do *buffer* é mantida baixa e o descarte dos pacotes OUT é agressivo. Dessa forma, a influência dos pacotes OUT, em pequena quantidade, se torna insignificante.



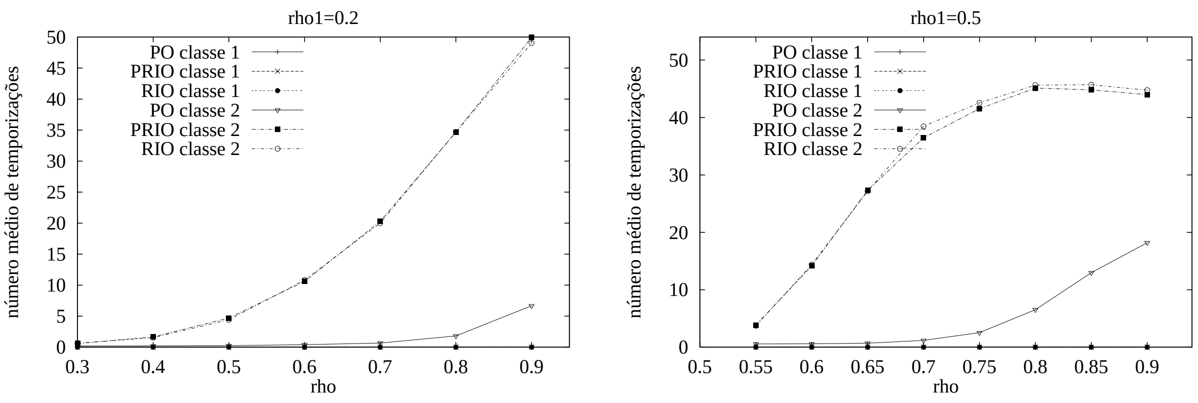
(a) $\rho_1 = 20\%$ (b) $\rho_1 = 50\%$

Figura 7: Taxa de perda das classes 1 e 2 em função da carga na rede.



(a) $\rho_1 = 20\%$ (b) $\rho_1 = 50\%$

Figura 8: Atraso na fila em função da carga na rede.



(a) $\rho_1 = 20\%$ (b) $\rho_1 = 50\%$

Figura 9: Número médio de temporizações das classes 1 e 2 em função da carga na rede.

4 Conclusões e perspectivas

Neste artigo discutimos sobre diversos mecanismos de descarte seletivo voltados para a diferenciação de serviços na Internet. Um maior enfoque foi dado aos mecanismos de descarte seletivo que utilizam técnicas de *push-out* para o descarte de pacotes menos prioritários. Esses mecanismos foram avaliados utilizando-se fontes de tráfego TCP de curta e longa duração. Para isso foi concebido um modelo de geração de tráfego TCP, através do qual pudemos gerar tráfegos que se assemelham ao encontrado em agregação de tráfegos *web* medidos empiricamente. Além de modelar o tráfego normalmente encontrado na Internet, esse modelo é parametrizável, via aproximação, pela carga a ser gerada na rede. A validade desta aproximação foi avaliada, garantindo a acurácia do modelo.

Em seguida, avaliamos o grau de diferenciação e a eficiência dos mecanismos de descarte seletivo no aproveitamento da banda passante disponível em diferentes condições de carga e tamanhos de transferência. Observamos que as políticas de remoção, quando associadas a mecanismos de gerenciamento de *buffer* de decarte prévio (tipo RED), não acrescentam vantagens, pois suas características são sobrepostas pelas dos mecanismos. Foi observado ainda que mecanismos de descarte seletivo do tipo *loss conserving* apresentam menor diferenciação entre classes de serviço.

Como perspectivas deste trabalho, pretendemos utilizar o modelo implementado para avaliar outros mecanismos de descarte seletivo, propor configurações de mecanismos do tipo RED apropriadas ao tráfego *web*, modificar o modelo de forma a gerar um tráfego que seja uma mistura de conexões TCP de curta e longa duração, mais próximo daquele encontrado na Internet.

Referências

[1] S. Blake, D. L. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services," *RFC 2475*, Dec. 1998.

[2] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group," *RFC 2597*, June 1999.

[3] M. Goyal, A. Durresi, P. Misra, C. Liu, and R. Jain, "Effect of Number of Drop Precedences in Assured Forwarding," in *IEEE/Globecomm'99*, (Rio de Janeiro), Dec. 1999.

[4] H. Chao, H. Cheng, Y.-R. Jenq, and D. Jeong, "Design of a Generalized Priority Queue Manager for ATM Switches," *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 867–880, June 1997.

[5] B. Suter, T. V. Lakshman, D. Stiliadis, and A. K. Choudhury, "Buffer Management Schemes for Supporting TCP in Gigabit Routers with Per-flow Queueing," *IEEE Journal on Selected Areas in Communications*, Sept. 1999.

[6] N. Yin, S. Li, and T. Stern, "Congestion Control for Packet Voice by Selective Discarding," *IEEE Transactions on Communications*, vol. 38, pp. 674–683, May 1990.

[7] J. J. Bae, T. Suda, and R. Simha, "Analysis of Individual Packet Loss in a Finite Buffer Queue with Heterogeneous Markov Modulated Arrival Processes: a Study of

Traffic Burtiness and Priority Packet Discarding," in *Proc. of IEEE INFOCOM'92*, pp. 219–230, June 1992.

[8] N. Fonseca and J. Silvester, "Estimating the Loss Probability in a Multiplexer Loaded with Multi-Priority MMPP Streams," in *Proc. of IEEE International Conference on Communication'93*, pp. 1037–1041, May 1993.

[9] I. Cidon, R. Guérin, and A. Khamisy, "On Protective Buffer Policies," *IEEE/ACM Transactions on Networking*, vol. 2, pp. 240–246, June 1994.

[10] N. Fonseca, M. Ferreira, and G. Mayor, "On the Effectiveness of Loss-Conserving Disciplines under a Long-Range Dependent Process," in *Proc. of IEEE Global Telecommunications Conference'97*, pp. 1102–1107, May 1997.

[11] H. Schulzrinne and J. Kurose, "Distribution of the Loss Period for Some Queues in Continuous and Discrete Time," in *Proc. of IEEE INFOCOM'91*, pp. 1446–1455, June 1991.

[12] S. Floyd and A. Romanov, "Dynamics of TCP Traffic over ATM Networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 4, pp. 633–641, 1995.

[13] R. Goyal, R. Jain, S. Kalyanaraman, and S. Fahmy, "Improving Performance of TCP over ATM-UBR Service," in *Proc. of ICC'97*, pp. 1042–1048, June 1997.

[14] J. Turner, "Maintaining High Throughput during Overload in ATM Switches," in *Proc. of INFOCOM'96*, pp. 287–295, Apr. 1996.

[15] T. Lakshman, A. Neidhardt, and T. Ott, "The Drop from Front Strategy in TCP over ATM," in *Proc. of INFOCOM'96*, pp. 1242–1250, Apr. 1996.

[16] M. Labrador and S. Banerjee, "Packet Dropping Policies for ATM and IP Networks," *IEEE Communications Surveys*, vol. 2, no. 3, 1999. http://www.comsoc.org/pubs/surveys.

[17] S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, Aug. 1993.

[18] B. Braden, D. D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, "Recommendations on queue management and congestion avoidance in the Internet," *Internet RFC 2309*, Apr. 1998.

[19] D. D. Clark and W. Fang, "Explicit Allocation of Best Effort Packet Delivery Service," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 362–373, Aug. 1998.

[20] H. Kroner, G. Hebuterne, P. Boyer, and A. Gravey, "Priority Management in ATM Switching Nodes," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 418–427, Apr. 1991.

[21] A. M. Lin and J. A. Silvester, "Priority Queueing Strategies and Buffer Allocation for Traffic Control at an ATM Integrated Broadband Switching System," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 1524–1536, Dec. 1991.

[22] A. K. Choudhury and E. L. Hahne, "A Simulation Study of Space Priorities in a Shared Memory ATM Switch," *Journal of High Speed Networks*, vol. 3, pp. 491–512, Nov. 1994.

[23] I. Cidon, L. Georgiadis, R. Guérin, and A. Khamisy, "Optimal Buffer Sharing," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1229–1240, Sept. 1995.

[24] A. K. Choudhury and E. L. Hahne, "Dynamic Queue Length Thresholds for Multi-priority Traffic," in *Proc. 15th International Teletraffic Congress*, pp. 561–570, June 1997.

[25] A. K. Choudhury and E. L. Hahne, "Dynamic Queue Length Thresholds for Shared-Memory Packet Switches," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 130–140, Apr. 1998.

[26] M. Molina, P. Castelli, and G. Foddis, "Web Traffic Modeling Exploiting TCP Connections' Temporal Clustering through HTML-REDUCE," *IEEE Network Magazine*, vol. 14, no. 3, pp. 46–55, 2000.

[27] M. E. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic: Evidence and possible causes," in *Proc. of the ACM SIGMETRICS Conference on Measurement & Modeling of Computer Systems*, (Philadelphia), pp. 160–169, 1996.

[28] M. Christiansen, K. Jeffay, D. Ott, and F. D. Smith, "Tunning RED for Web Traffic," in *Proc. ACM/SIGCOMM'00*, (Stockholm), 2000.

[29] R. Jain, *The Art of Computer Systems Performance Analysis*. John Wiley and Sons Inc., 1991.