

IOPQ: Uma fila para Implementação de Serviços não-Eletivos na Internet

Ramide Dantas¹, Carlos Kamienski^{1,2}, Dênio Mariz^{1,2}, Djamel Sadok²

¹Centro Federal de Educação Tecnológica da Paraíba (CEFET-PB)
Rua 1º de Maio, 720 – 58015-180 – João Pessoa, PB

²Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – Cidade Universitária – 50.732-970 – Recife – PE
rasd@coinfo.cefetpb.edu.br, {cak,dmts,jamel}@cin.ufpe.br

Abstract. *Despite the recent trend for deploying QoS-based services, the Internet still offers only the best effort service. Therefore, non-elevated services have been proposed, offering low delay to some applications that can support a higher packet loss rate. There are currently some approaches for providing non-elevated services, yet they generally rely on complex mechanisms. In this paper we propose a simple mechanism based on a FIFO queue that allows setting different priorities for incoming and outgoing packets, called IOPQ (In/Out Priority Queue). IOPQ was evaluated through comparative simulations with DropTail and RED queues, showing some significant results.*

Resumo. *Apesar da tendência de introduzir serviços baseados em QoS, a Internet continua a oferecer apenas o serviço de melhor esforço. Por este motivo, serviços não-eletivos têm sido propostos, oferecendo baixo atraso para algumas aplicações, que suportam uma taxa de perda de pacotes mais alta. Algumas propostas de serviços não-eletivos têm sido apresentadas, mas em geral são baseadas em mecanismos complexos. Neste artigo propõe-se um mecanismo simples baseado em uma fila FIFO, capaz de atribuir prioridades diferentes para entrada e saída da fila, chamado de IOPQ (In/Out Priority Queue). IOPQ foi avaliada através de simulações comparativas com filas DropTail e RED, apresentando resultados significativos.*

1. Introdução

A Internet pública continua a ser uma rede de melhor esforço, a despeito dos esforços efetuados nos últimos anos para oferecer serviços avançados baseados em QoS. Devido a diversos motivos, técnicos e comerciais, QoS ainda não é uma realidade na Internet. Do ponto de vista técnico, existe uma grande discussão sobre disponibilizar mecanismos de gerenciamento dos recursos da rede versus aumentar a capacidade da rede (QoS X superprovisionamento [9]). Além disso, serviços baseados em QoS exigem que todos os roteadores no caminho entre fonte e destino (fim a fim) o implementem corretamente para que os resultados estejam de acordo com as garantias oferecidas aos usuários. Na Internet2, por

exemplo, o serviço QPS [14] (Qbone Premium Service) foi descontinuado, devido a motivos como falta de demanda, falta de suporte pelos fabricantes, necessidade de atualização integral da rede [13].

Do ponto de vista comercial, o principal problema é o modelo de cobrança de serviços diferenciados. A concessão de garantias de QoS a alguns usuários tende a comprometer o desempenho percebido pelos demais usuários (do serviço de melhor esforço). Isso leva à necessidade de implementar um modelo de cobrança com vários níveis, que envolve medição de tráfego, em vez de utilizar o modelo de tarifa plana (*flat rate*) mais simples de gerenciar. A grande dificuldade deste modelo é remunerar os múltiplos provedores no caminho fim a fim, onde as interconexões entre eles podem ser efetivadas através de relacionamentos de trânsito ou de parceria (*peering*). O problema é que *peering* gera uma situação onde somente metade do caminho é remunerado em uma comunicação fim a fim [8].

Por este motivo, pesquisas recentes em diferenciação de serviços têm sido direcionadas para criar mecanismos destinados a prover serviços não-eletivos [7], [4], [6], cujo objetivo é oferecer baixo atraso para aplicações em tempo real (ex: multimídia interativa), às custas de permitir um aumento na sua taxa de perda de pacotes. Serviços não-eletivos oferecem garantias fracas de QoS, onde as diferentes classes de serviços recebem um tratamento “diferente e igual” simultaneamente. Entretanto, eles mantêm a simplicidade do modelo atual de melhor esforço da Internet, porque não necessitam de condicionamento de tráfego (medição, marcação, descarte, etc.) e poucas alterações na infra-estrutura são necessárias. Eles podem ser implantados de maneira incremental, e o modelo de tarifa plana pode ser utilizado. O princípio básico de serviços não-eletivos é oferecer garantias de atraso a algumas aplicações, enquanto o tráfego de outras aplicações mais sensíveis à vazão não é prejudicado.

Algumas propostas têm sido apresentadas para implementar serviços não-eletivos, como Alternative Best effort (ABE) [7], Best effort Differentiated Services (BEDS) [4] e Equivalent Differentiated Services (EDS) [6]. No entanto, esses modelos são, em geral, baseados em mecanismos relativamente complexos de gerenciamento de filas e escalonamento de pacotes. O Scavenger Service [12], definido para a Internet2, tem uma implementação simples, mas seu modelo de serviço é diferente, do tipo “pior que o melhor esforço”.

Neste artigo propõe-se um mecanismo simples baseado em uma fila FIFO tradicional, capaz de atribuir prioridades diferentes de entrada e saída da fila aos pacotes, chamado de IOPQ (In/Out Priority Queue). Ela faz uso de prioridades atribuídas ao tráfego para definir critérios de gerenciamento da fila e escalonamento dos pacotes no meio físico. Por meio dessas prioridades são oferecidos os serviços de “atraso reduzido” e “descarte reduzido”. Neste artigo, foi utilizado um critério de classificação que atribui esses serviços ao tráfego UDP e TCP, respectivamente.

A fila IOPQ foi avaliada através de simulações comparativas com filas DropTail e RED, apresentando resultados animadores. Ela foi capaz de reduzir significativamente o atraso de tráfego UDP, apresentando uma taxa de perda de pacotes limitada em valores aceitáveis, enquanto que a vazão de aplicações TCP não foi alterada.

Na sequência deste artigo, a seção 2 apresenta trabalhos relacionados na área de serviços não-eletivos. A descrição da fila IOPQ está na seção 3. Na seção 4 são apresentadas as configurações e cenários de simulação utilizados para obter os resultados, que são mostrados e comentados na seção 5. Finalmente, a seção 6 delinea algumas conclusões e apresenta indicações de trabalhos futuros.

2. Trabalhos Relacionados

Em [7] é apresentado o serviço "melhor esforço alternativo" (*Alternative Best Effort - ABE*), uma melhoria do "melhor esforço" tradicional, que se baseia na idéia de prover baixa atraso às custas de uma menor vazão. O modelo ABE introduz duas classes de serviço, "green" e "blue". A classe "blue" corresponde ao serviço melhor esforço e a classe "green" é uma classe que experimenta baixo atraso, com limites estabelecidos. Duas condições devem acontecer para que "green não interfira em blue": transparência local e transparência de vazão para blue. Transparência local para blue assume que o atraso experimentado por blue no ABE não seja maior que aquele que experimentaria no modelo melhor esforço. Além disso, o pacote blue não deve ser descartado no ABE se não devesse ser descartado no modelo melhor esforço. Transparência de vazão assume que um fluxo blue alcança pelo menos a mesma vazão que alcançaria no modelo melhor esforço. Embora o ABE proveja transparência local para o blue, ele relaxa na transparência de vazão, ou seja, não há garantia que o blue alcança a mesma vazão que alcançaria no modelo melhor esforço. O serviço ABE consegue prover baixo atraso sem o conceito de reserva ou sinalização e preserva o modelo de tarifa plana para tarifação.

O modelo *Best effort Differentiated Services (BEDS)* [4] define dois tipos de serviços: "drop-conservative", que busca privilegiar uma baixa taxa de descarte (cujo alvo é o TCP) e "delay-conservative", que busca privilegiar a redução do atraso (com UDP como alvo). Dessa forma, o serviço "drop-conservative" tem menor perda de pacotes, mas maior atraso em relação ao serviço "delay-conservative". No BEDS, pacotes TCP e UDP são enfileirados em duas filas RED separadas, com parâmetros distintos. Com base no esquema de escalonamento empregado, três versões do BEDS são definidos: a) com taxa forçada de atraso; b) com taxa forçada de perda de pacotes e; c) com taxa forçada de atraso e perda. A deficiência do BEDS com taxa forçada de atraso é que o tráfego UDP obtém menor atraso e alta vazão, comparados ao serviço melhor esforço. Ou seja, esse tipo de diferenciação é injusto e não obedece à idéia geral por trás dos serviços não-eletivos. Para lidar com esse problema, um parâmetro α para diferenciação de perda foi definido nas versões "com taxa forçada de atraso" e "com taxa forçada de atraso e perda" do BEDS. Entretanto, a escolha do parâmetro α para obter-se pelo menos a mesma vazão para os fluxos TCP no modelo melhor esforço é um importante problema que não foi discutido em [4]. Além disso, nenhum resultado analítico ou baseado em simulações foi apresentado para ilustrar a influência dos mecanismos propostos na vazão de fluxos TCP e UDP.

O modelo *Equivalent Differentiated Services (EDS)*, [6] é semelhante ao ABE e ao BEDS, no sentido de que praticam o mesmo tipo de balanceamento entre atraso e perda. O modelo EDS definiu um conjunto de classes com diferenciação assimétrica de atraso e perda. Parâmetros para diferenciação de perda e atraso são ajustados de forma que se uma classe α

experimenta um menor atraso do que a classe *b*, então a classe *b* experimenta uma menor probabilidade de perda que a classe *a*. Contudo, o ajuste do parâmetro de diferenciação não é feito pelo mecanismo EDS, mas por um protocolo de uma camada superior. Outra opção é oferecer um conjunto de classes com coeficientes fixos de diferenciação para atraso e probabilidade de perda, e deixar as fontes de tráfego decidirem quais classes satisfazem os requisitos dos seus serviços.

O serviço *Scavenger* é um mecanismo de rede proposto pelo projeto QBone, da rede Internet2 [12]. O *Scavenger* é um mecanismo que permite que usuários e aplicações tirem proveito da capacidade ociosa da rede, de maneira que isso não afete substancialmente o desempenho das classes tradicionais de serviço melhor esforço. De forma simples, o *Scavenger* define um serviço “pior que o melhor esforço”, criando uma rede virtual paralela com recursos altamente escassos, mas que pode expandir sua capacidade de forma elástica até atingir a capacidade da classe normal de melhor esforço, sempre que a rede apresente ciclos de ociosidade. A expansão ocorre com uma alta granularidade de tempo, permitindo que toda a capacidade não usada pela classe padrão seja aproveitada pela classe *Scavenger*. Para usar o serviço *Scavenger*, usuários e aplicações marcam voluntariamente algum tráfego para tratamento *Scavenger* ajustando o campo *differentiated services code point* (DSCP) para o valor binário 001000 no cabeçalho IP. Roteadores colocam este tráfego em uma fila especial com uma capacidade alocada muito pequena usando uma disciplina de fila como *weighted round-robin* (WRR), *modified deficit round-robin* (MDRR), *weighted fair queuing* (WFQ), ou similar.

3. A Fila IOPQ

Seguindo a idéia de serviços não-eletivos como ABE e BEDS, o modelo proposto visa oferecer às aplicações multimídia baseadas em UDP uma redução no atraso total sofrido pelos pacotes — quando comparado ao serviço de melhor esforço padrão — sem prejudicar a vazão obtida pelo tráfego TCP. O tráfego UDP, em contrapartida, estaria sujeito a uma maior taxa de descarte de pacotes. Aos fluxos TCP, cuja vazão é mais sensível à perda de pacotes, seria oferecida uma menor probabilidade de descarte, ao custo de um maior tempo de permanência na fila.

Para atingir este propósito, foram elaborados dois tipos (ou classes) de serviço: “atraso reduzido”, destinado ao tráfego UDP; e “descarte reduzido”, para o tráfego TCP. Ambos são, por definição, serviços de melhor esforço, não oferecendo garantias rígidas quanto ao atraso ou taxa de descarte sofridos pelos pacotes. Diferentemente de abordagens como IntServ [3] e DiffServ [2], não é feita alocação prévia de recursos, tampouco controle de admissão, marcação, medição ou policiamento do tráfego. Em consequência disto, o comportamento do tráfego pode fazer com que os serviços propostos degenerem para o serviço de melhor esforço padrão.

Para implementar o modelo proposto foi desenvolvida a fila IOPQ (*In/Out Priority Queue*, Fila de Prioridades de Entrada e Saída). Ela faz uso de prioridades atribuídas ao tráfego para definir critérios de gerenciamento da fila e de escalonamento dos pacotes no meio físico. Por meio dessas prioridades são oferecidos os serviços de “atraso reduzido” e “descarte reduzido”. Neste artigo, foi utilizado um critério de classificação que atribui esses

serviços ao tráfego UDP e TCP, respectivamente. É importante observar que qualquer critério de classificação pode ser usado para enquadrar os pacotes em determinado serviço.

Ao chegar na fila IOPQ, cada pacote recebe uma prioridade de entrada e outra de saída, que pode ser alta ou baixa. Estas prioridades determinam se o pacote entrará e permanecerá na fila e quando ele será encaminhado. Pode-se distinguir quatro tipos de serviços resultantes da combinação destas prioridades, conforme mostra a Tabela 1.

Para promover o serviço de “descarte reduzido” para o tráfego TCP, a fila IOPQ foi configurada de forma que aos pacotes TCP fosse atribuída alta prioridade de entrada e baixa prioridade de saída. Aos pacotes UDP são atribuídas prioridades baixa e alta para a entrada e saída respectivamente, constituindo o serviço de “atraso reduzido”.

Tabela 1. Serviços oferecidos pela fila IOPQ

Tipo de prioridade		Serviço oferecido
Entrada	Saída	
Baixa	Baixa	Atraso e taxa de descarte de pacotes elevados.
Baixa	Alta	Atraso reduzido, maior taxa de descarte.
Alta	Baixa	Descarte reduzido, maior atraso.
Alta	Alta	Atraso e descarte reduzidos.

Pacotes com alta prioridade de entrada têm maior facilidade para serem inseridos na fila. Se necessário, pacotes com baixa prioridade serão descartados para ceder espaço na fila aos de maior prioridade. A prioridade alta também previne que um pacote seja descartado uma vez que ele entrou na fila. A seguir está descrito o procedimento de entrada de um pacote p na fila:

- a. O pacote p é classificado, recebendo prioridades de entrada e saída;
- b. Estando a fila com espaço disponível, p entrará no final da fila, independente de sua prioridade de entrada;
- c. No caso da fila estar cheia, a prioridade de entrada de p é verificada:
 - c.1. Se a prioridade de p for baixa, p é descartado;
 - c.2. Se for alta, procede-se como a seguir:
 - c.2.1. Seja q o primeiro pacote da fila (menos recente) dentre os pacotes com baixa prioridade de entrada. Caso q exista, é atribuída a q alta prioridade de entrada.
 - c.2.2. Seja q' o último pacote da fila (mais recente) dentre os pacotes com baixa prioridade de entrada. Caso q' exista, q' é descartado e p entra no fim da fila; caso contrário, p é descartado.

Havendo a necessidade de retirar da fila um pacote com baixa prioridade de entrada (conseqüência do passo c.2), antes é atribuída prioridade alta ao pacote com baixa prioridade

que está mais próximo de ser encaminhado. Na prática, este mecanismo de “promoção” evita que um pacote UDP que está na iminência de ser encaminhado seja descartado. Como a elevação de prioridade é feita antes da entrada do novo pacote na fila, este pacote será descartado caso exista apenas um pacote com baixa prioridade na fila. Com isto, eventualmente pacotes TCP serão descartados — apesar de possuírem preferência para entrar na fila —, o que é desejável pois força as fontes TCP a diminuïrem suas taxas de transmissão periodicamente.

Nas primeiras versões da fila IOPQ não existia o mecanismo de “promoção” de pacotes UDP. Estando a fila cheia na chegada de um pacote TCP, invariavelmente era removido o último pacote UDP da fila, caso houvesse, de forma a ceder espaço para a inserção do pacote TCP recém chegado no final da fila. Durante os testes preliminares desta implementação foi percebida uma taxa de descarte para o tráfego UDP muito acima do tolerado por qualquer aplicação. Sob determinadas condições, ocorria a estagnação do tráfego UDP, com total predomínio do tráfego TCP. O mecanismo de “promoção” foi introduzido visando solucionar estes problemas, conforme foi explicado no parágrafo anterior.

Pacotes com alta prioridade de saída possuem preferência no escalonamento, de forma que nenhum pacote com baixa prioridade de saída seja escalonado, caso haja na fila pacotes com prioridade alta. Em resumo, pacotes UDP sempre serão encaminhados antes de pacotes TCP. O escalonamento de um pacote é efetuado como descrito abaixo:

- a. Seja p o primeiro da fila dentre os pacotes com alta prioridade de saída:
 - a.1. Existindo p , ele é retirado da fila e encaminhado;
 - a.2. Caso contrário, é encaminhado o primeiro pacote da fila com baixa prioridade de saída, caso exista (isto é, quando a fila não estiver vazia).

IOPQ preserva a seqüência dos pacotes com mesma prioridade de saída, evitando reordenação no destino. Quando o tráfego contiver apenas pacotes com exatamente as mesmas prioridades, tanto de entrada quanto de saída, a fila IOPQ se comporta de forma análoga a FIFO/DropTail, tornando o serviço oferecido idêntico ao de melhor esforço.

A fila IOPQ foi desenvolvida tendo em mente o mecanismo de controle de congestionamento do protocolo TCP. Ela busca oferecer ao tráfego TCP uma menor taxa de descarte de pacotes, de modo que a probabilidade de uma janela inteira de segmentos ser perdida é menor e conseqüentemente o transmissor TCP tende a cair significativamente menos para o algoritmo de *slow start* (que causa diminuição de desempenho). Em contrapartida, o *Round Trip Time* (RTT) dos pacotes tende a ser mais elevado, já que o atraso imposto pela fila é maior. Conseqüentemente, a velocidade com a que a janela de transmissão cresce estando o transmissor em *congestion avoidance* tende a diminuir. Isto evita um crescimento acelerado da taxa de transmissão, evitando que o tráfego TCP se apodere da banda disponível¹.

¹ Detalhes sobre o mecanismo de controle de congestionamento do protocolo TCP podem ser encontrados em [1].

Não há necessidade de configuração adicional da fila IOPQ. O único parâmetro a ser definido é sua capacidade, que é limitada pelo equipamento em que será utilizada, da mesma forma que no serviço de melhor esforço atual. Em consequência, o impacto na operação da rede é mínimo.

4. Configurações de Simulação

O modelo de serviço não-eletivo implementado pela fila IOPQ foi avaliado através de simulação, utilizando o Network Simulator 2 (*ns-2* [11] versão 2.26). Para este estudo inicial, foi adotada uma topologia simples, apresentada na Figura 1. As simulações envolveram três tipos diferentes de filas: IOPQ, DropTail e RED [5]². O desempenho de cada fila foi avaliado sob condições de tráfego variadas e os resultados comparados em seguida.

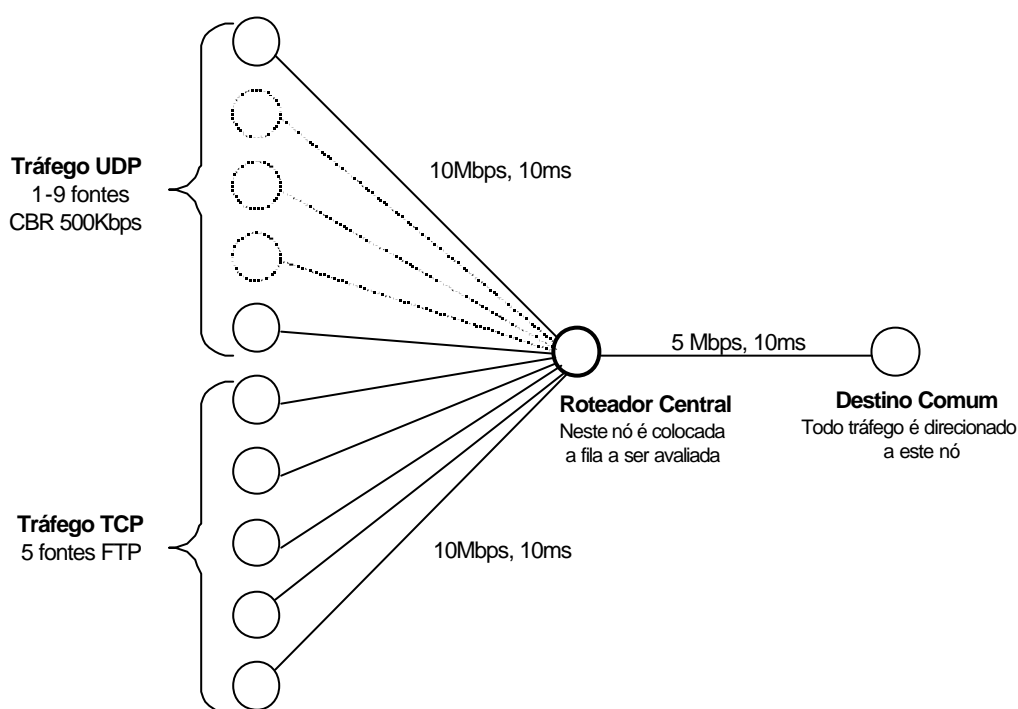


Figura 1- Topologia de simulação

A rede utilizada é formada por um número variável de nós, que hospedam as fontes de tráfego, interligados a um nó central, por sua vez conectado ao nó de destino. Os enlaces entre as fontes e o nó central têm a capacidade de 10 Mbps, enquanto que o enlace entre o nó central e o nó de destino é de 5 Mbps, representando o gargalo da rede. Foi adotado em todos os enlaces um atraso de propagação de 10 ms. A fila a ser avaliada foi configurada na saída do nó central para o nó de destino do tráfego.

As filas foram configuradas com tamanho máximo de 60 pacotes. Para IOPQ e DropTail foi necessária nenhuma configuração adicional. Os parâmetros utilizados para RED

² Mecanismos mais complexos de gerenciamento de filas e escalonamento de pacotes utilizados para prover serviços diferenciados (como WFQ - Weighted Fair Queuing) não foram incluídos, uma vez que o escopo desta avaliação são os serviços não-eletivos.

foram: limiar mínimo (min_th) = 20 pacotes; limiar máximo (max_th) = 50 pacotes; probabilidade máxima de descarte antecipado (max_p) = 0,1; peso do tamanho instantâneo no tamanho médio da fila (w) = 0,002. O motivo de incluir RED na comparação é a sua característica de descartar antecipadamente pacotes TCP e UDP de forma indiscriminada. Este comportamento produz resultados relevantes para ambos os protocolos, como mostrado na seção 5, embora aumente consideravelmente a taxa de perda de pacotes.

Para gerar o tráfego TCP foram usadas 5 fontes FTP (de longa duração). Foram executados experimentos variando o número de fontes UDP de 1 a 9. Cada fonte UDP gerava tráfego CBR (taxa constante de bits) com taxa de 500Kbps (para simular transmissões de vídeo). A participação do agregado de tráfego UDP com relação à capacidade do enlace de gargalo variou, entre os experimentos, de 10 a 90 por cento. O tamanho do pacote UDP foi configurado para 500 bytes.

O tempo total de simulação de cada experimento foi de 60 segundos, tempo considerado suficiente para se observar os resultados desejados. Para cada experimento foram realizadas 30 replicações, que representa um compromisso aceitável entre o custo computacional e a confiança estatística nos resultados. Todas as fontes de tráfego (TCP e UDP) foram iniciadas de forma aleatória dentro do primeiro segundo de simulação e em instantes diferentes para cada replicação realizada.

Para todos os resultados, foi calculado um intervalo de confiança assintótico ao nível de 99,9 %, que é representado através de barras horizontais nos resultados da seção 5. Na maior parte dos casos, no entanto, as barras não são visíveis, porque os resultados não apresentaram variações muito significativas (ou seja, os intervalos de confiança apresentaram tamanho reduzido).

As seguintes métricas foram coletadas durante as simulações: a vazão obtida por cada tipo de tráfego; o atraso total e a variação do atraso (*jitter*) experimentados pelo tráfego UDP; a taxa de descarte de pacotes; e a ocupação média da fila. As medições foram realizadas considerando apenas um sentido do tráfego, da fonte para o destino. Os resultados estão apresentados na seção seguinte.

5. Resultados

Esta seção apresenta os resultados de simulação obtidos no estudo comparativo realizado de acordo com os parâmetros, fatores e níveis apresentados na seção anterior. O objetivo é demonstrar que, tendo uma implementação simplificada — se comparada a outros modelos de serviços não-eletivos como ABE e BEDS —, a fila IOPQ, nos cenários simulados, foi capaz de oferecer baixo atraso para o tráfego UDP com uma taxa de perda tolerável, enquanto o impacto sobre o desempenho dos fluxos TCP com relação à vazão foi pequeno.

5.1. Vazão

A Figura 2 apresenta os resultados correspondentes à vazão média obtida por cada tipo de tráfego para os três tipos de fila avaliados. O que se percebe é uma equivalência na vazão proporcionada pelas filas IOPQ, DropTail e RED para ambos os tipos de tráfego até o ponto em que a participação do tráfego UDP é de 60 %. Deste ponto em diante, o tráfego UDP

passa a ser privilegiado pela fila IOPQ, ainda que de forma sutil. RED, ao contrário, privilegia o tráfego TCP. Tal proporção de tráfego UDP não ocorre na Internet, ficando normalmente limitada a 5 por cento do total do tráfego [15]. Considerando este fato, os resultados demonstram que, ao menos para os cenários simulados, a fila IOPQ foi capaz de executar sua função sem prejudicar a vazão do tráfego TCP, se aproximando ao que seria obtido com fila DropTail (ou RED), que é um dos objetivos dos modelos não-eletivos.

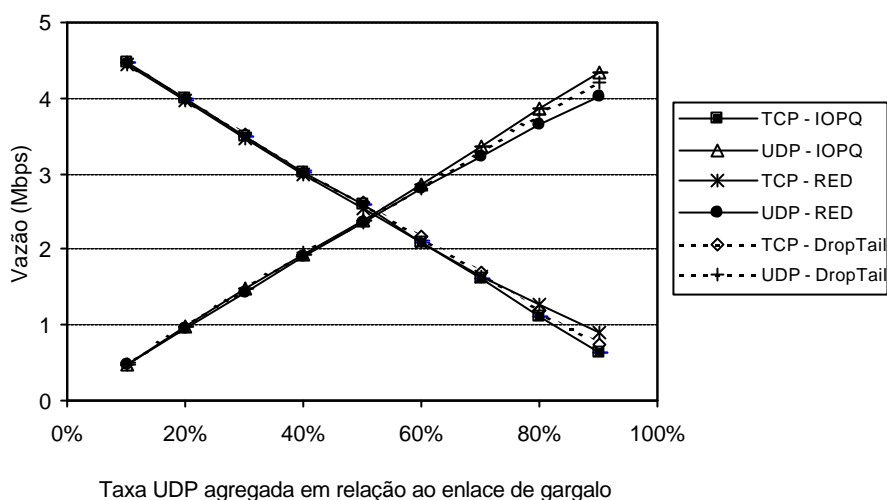


Figura 2- Vazão média das fontes TCP e UDP

5.2. Atraso e variação do atraso

O atraso imposto pela fila IOPQ aos pacotes UDP mostrou-se sempre inferior ao obtido com uso do serviço de melhor esforço padrão, o que pode ser observado na Figura 3. Enquanto o atraso total médio com a fila IOPQ ficou em torno de 24 ms, com a fila DropTail obteve-se um atraso de cerca de 100 ms. Nestes valores está embutido o atraso de propagação dos enlaces, que no cenário utilizado soma 20 ms. O tempo médio de permanência na fila ficou acima de 80 ms com a fila DropTail. Com a fila IOPQ, este tempo foi de cerca de 4 ms.

No cenário com RED, o atraso total obtido ficou em torno de 35 ms. Este fato não implica que RED pode ser um mecanismo utilizado para implementar serviços não eletivos, pois o RED realiza os descartes antecipados de pacotes de maneira indiscriminada. Desta forma, para este cenário o atraso foi menor, mas em outras situações não seria possível prever o seu comportamento. O baixo atraso do tráfego UDP com RED ocorreu devido ao tamanho da fila no roteador ter sido mantido próximo a 12 pacotes, enquanto que IOPQ e DropTail atingiram valores ao redor de 50 pacotes (Figura 6). Como a ocupação média da fila é menor, os pacotes passam menos tempo esperando o escalonamento e como consequência o atraso também é menor.

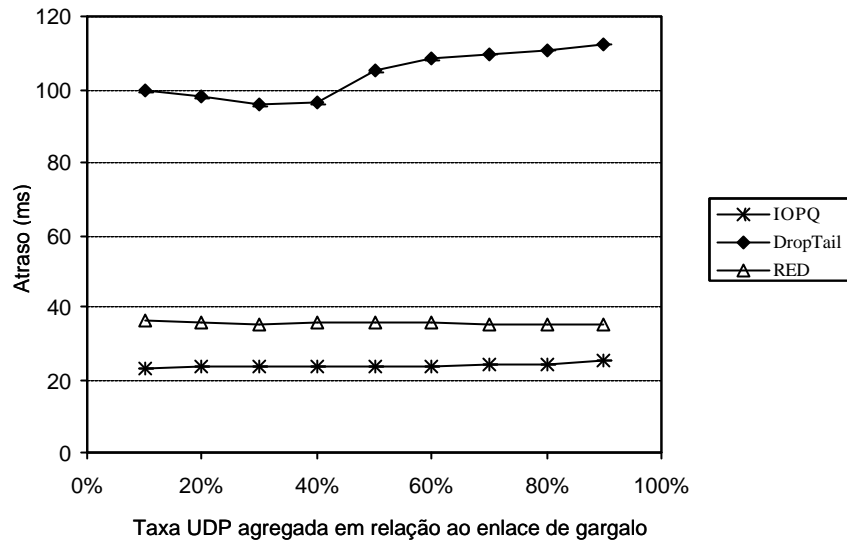


Figura 3 – Atraso médio das fontes UDP

Também a variação do atraso mostrou-se menor quando utilizada a fila IOPQ, não ultrapassando 2 ms, conforme mostra a Figura 4. Vale ressaltar que para IOPQ observa-se uma tendência de crescimento da variação do atraso, o que não ocorre com DropTail e RED. Um estudo mais detalhado será necessário para determinar se esse crescimento se comprova e, em caso positivo, se é suficientemente grande para introduzir algum tipo de problema potencial. No entanto, para o cenário avaliado as três filas produziram baixos valores de variação de atraso (inferior a 3 ms), de modo que este parâmetro não apresenta maior relevância neste estudo.

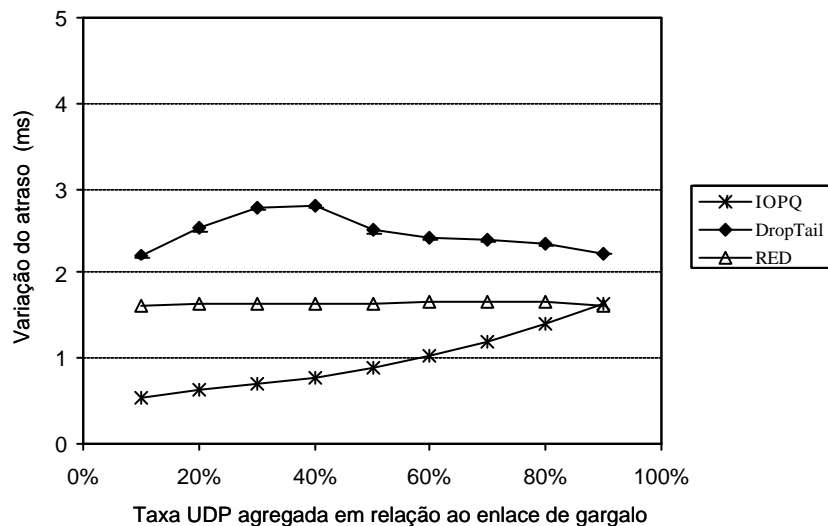


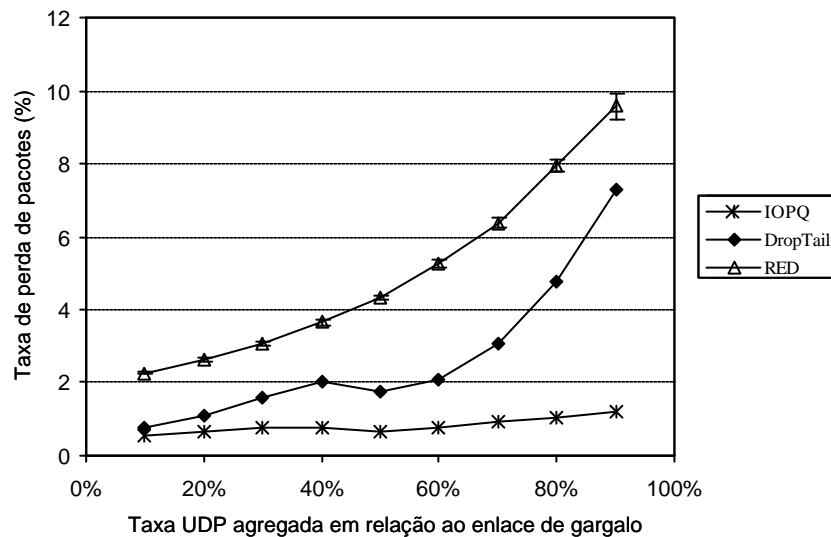
Figura 4 – Variação do atraso média das fontes UDP

5.3. Taxa de Descarte de Pacotes

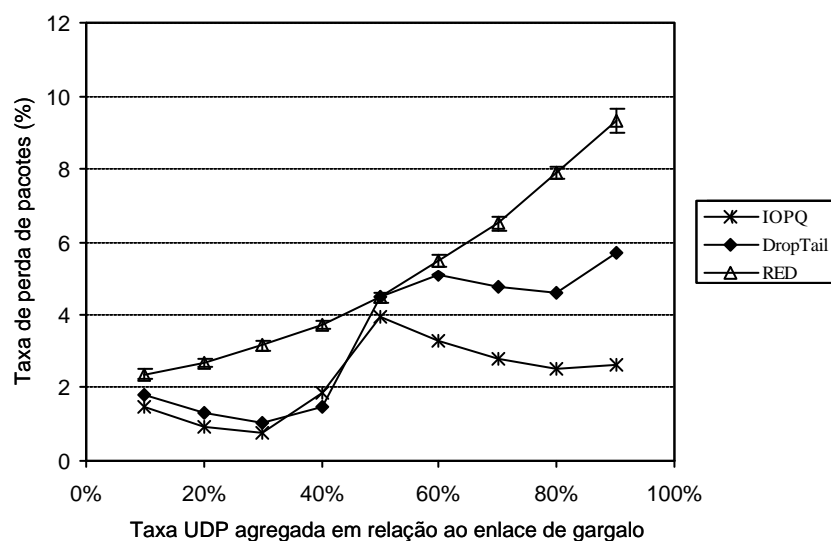
Um dos resultados mais relevantes deste trabalho diz respeito à taxa de descarte de pacotes, uma vez que serviços não-eletivos em geral fornecem baixo atraso, assumindo uma postura

mais flexível com a taxa de perda de pacotes (desde que esteja dentro de limites aceitáveis para as aplicações).

A Figura 5a mostra que o tráfego TCP obteve uma taxa de descarte de pacotes de cerca de 1 % quando utilizada a fila IOPQ, ficando abaixo da perda oferecida pela DropTail nas condições de tráfego simuladas. Também o tráfego UDP (Figura 5b) foi beneficiado com o uso da fila IOPQ, ficando a taxa de perda muito próxima ou mesmo inferior à perda proporcionada pela fila DropTail. Este resultado é significativamente superior ao esperado para a fila IOPQ, que poderia suportar taxas de perda de pacotes levemente maiores que a fila DropTail. Isso se deve à característica específica do cenário de simulação utilizada (topologia, mix de tráfego, etc.). Em simulações adicionais realizadas com parâmetros diferentes, a taxa de perda obtida por IOPQ assumiu valores ligeiramente superiores a DropTail, conforme o esperado.



(a)



(b)

Figura 5- Taxa de perda de pacotes média das fontes; a) TCP; b) UDP

Um ponto importante a ser observado é a taxa de perda de pacotes para TCP e UDP produzida com a utilização de RED, que foi significativamente mais alta do que DropTail e IOPQ³. Esse resultados vem a corroborar com a conclusão da seção 5.2, de que RED não é um mecanismo adequado para implementar serviços não-eletivos do tipo “baixo atraso” e “baixa perda”, como IOPQ. A taxa de perda de pacotes, para qualquer tipo de tráfego, pode tornar-se incontrolável, conforme mostram os resultados da Figura 5. É importante também observar que os resultados de RED representam um limite superior para a perda de pacotes, tanto para TCP quanto para UDP.

5.4. Ocupação da Fila

A Figura 6 mostra a taxa média de ocupação das filas DropTail, IOPQ e RED. IOPQ manteve o tamanho médio da fila similar ao DropTail. Quando a participação de tráfego UDP é mais baixa (até 50%), o tamanho da fila observado para o IOPQ ficou levemente superior ao DropTail. A partir de 50%, essa tendência se inverte. É interessante notar que apesar dessa similaridade de valores, o tráfego UDP obteve atraso significativamente menor com IOPQ do que com DropTail (Figura 3, conforme mencionado na seção 5.2).

RED conseguiu manter a fila com baixa ocupação, devido ao descarte antecipado de pacotes. Isto ocasionou um atraso baixo para o tráfego UDP (conforme explicado na seção 5.2). Esta característica observada de maneira isolada poderia representar um resultado razoavelmente bom. No entanto, deve-se compará-lo com o resultado de perda de pacotes, já discutido na seção 5.3.

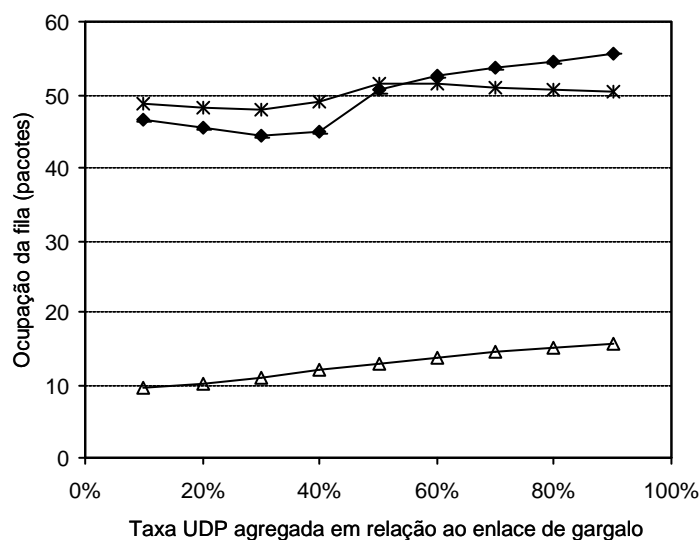


Figura 6- Ocupação média da fila

³ Os resultados foram obtidos com os valores padrão do *ns-2* para os parâmetros do RED (descritos na seção 4), embora a utilização de outros valores pudesse produzir taxas de perdas mais baixas. Duas razões balizaram esta escolha. Em primeiro lugar, vários estudos envolvendo RED utilizam tais valores (várias referências podem ser encontradas na literatura). Em segundo lugar, a configuração dos parâmetros do RED para fazer sintonia fina nos resultados não é trivial, nem existe uma fórmula aplicável a todos os cenários [10].

6. Conclusão

Este trabalho apresentou a proposta da fila IOPQ, que implementa um modelo de serviço não-eletivo capaz de proporcionar baixo atraso para tráfego UDP (possivelmente suportando uma maior taxa de perda de pacotes) e baixa perda de pacotes para tráfego TCP. O objetivo do segundo aspecto é manter a vazão das aplicações baseadas em TCP em níveis equivalentes aos obtidos pelo serviço de melhor esforço, padrão do protocolo IP. IOPQ foi avaliada através de simulações comparativas com filas DropTail e RED, apresentando resultados significativos.

Os resultados de simulação apresentados neste artigo demonstraram a efetividade da fila IOPQ em implementar o serviço proposto. O tráfego UDP com IOPQ obteve atraso significativamente inferior ao registrado com fila DropTail. Apesar de beneficiar UDP com baixo atraso (o que conseqüentemente aumenta o atraso de outros tipos de tráfego), IOPQ não afetou o desempenho do tráfego TCP, o que pôde ser comprovado nos resultados referentes à vazão. Ao mesmo tempo, a taxa de perda de pacotes produzida por IOPQ foi mantida em níveis considerados aceitáveis (pelo menos, inferiores aos produzidos com DropTail para o cenário avaliado). Com relação à avaliação de RED, a sua utilização não se mostrou adequada para a implementação de serviços não-eletivos, pois a taxa de descarte e atraso gerada não é controlável.

Como trabalhos futuros, pretende-se analisar a inclusão do mecanismo proposto para IOPQ em uma fila que implementa RED, com a expectativa de obter os benefícios proporcionados por RED, ao mesmo tempo em que conserva as características de serviços não-eletivos de IOPQ. Pretende-se também avaliar a adequação de IOPQ como um mecanismo de implementação de ABE. Além disso, é importante avaliar IOPQ em uma topologia de rede maior e mais real, uma vez que o discernimento básico acerca do seu funcionamento foi obtido com os estudos apresentados neste artigo.

Referências

- [1] Allman, M., Paxson, V. & Stevens W., "TCP Congestion Control", RFC 2581, Abril de 1999.
- [2] Black, D. et al., An Architecture for Differentiated Services, RFC 2475, Dezembro de 1998.
- [3] Braden, R., Clark, D. & Shenker, S., Integrated Services in the Internet Architecture: an Overview, RFC 1633, Junho de 1994.
- [4] Firoiu, V. & Zhang, X., "Best Effort Differentiated Services: trade-off service differentiation for elastic applications," in Proceedings of IEEE ICT'01, Bucharest, Romania, Junho de 2001.
- [5] Floyd, S., and Jacobson, V., "Random Early Detection gateways for Congestion Avoidance ", IEEE/ACM Transactions on Networking, Agosto de 1993.
- [6] Gaidioz, B. & Primet, P., "EDS: a new scalable service differentiation architecture for Internet," IEEE Symposium on Computers and Communications (ISCC), Julho de 2002.

- [7] Hurley, P., Kara, M. Boudec, J. Y. Le & Thiran, P., “ABE: providing a low-delay service within best effort,” IEEE Network, Maio de 2001.
- [8] Huston, G., “Interconnection, Peering, and Settlements”, INET’99, Junho de 1999.
- [9] Kamienski, C.A., “An Architecture for Providing End-to-End QoS-based Advanced Services in the Internet”, Tese de Doutorado, Universidade Federal de Pernambuco, Fevereiro de 2003.
- [10] May, M., Bolot, J., Diot, C. & Lyles, B, “Reasons not to Deploy RED”, IEEE/IFIP International Workshop on Quality of Service - IWQoS’99, Junho de 1999.
- [11] Network Simulator (versão 2.26), <http://www.isi.edu/nsnam/ns/>.
- [12] QBone Scavenger Service (QBSS). <http://qbone.internet2.edu/qbss>, Acessada em 14 de março de 2004.
- [13] Teitelbaum, B., “Future priorities for Internet2 QoS,” Internet2 QoS WG, Outubro de 2001, <http://www.internet2.edu/qos/wg/papers/qosFuture01.pdf>.
- [14] Teitelbaum, B., “Internet2 QBone: Building a Testbed for IP Differentiated Services”, IEEE Network, Setembro de 1999.
- [15] Thomson, K., Miller, G., & Wilder, R., “Wide-Area Internet Traffic Patterns and Characteristics”, IEEE Network, Novembro de 1997.