

WWTP: Um Escalonador de Pacotes para o Modelo de Diferenciação Proporcional Baseado em Janelas de Medidas

Alencar de Melo Júnior¹, Maurício Ferreira Magalhães², Juan Manuel Adán Coello³

¹Universidade São Francisco (USF)

R. Alexandre Rodrigues Barbosa, 45 – 13251-900 – Itatiba – SP – Brasil

²Universidade Estadual de Campinas (Unicamp)

³Pontifícia Universidade Católica de Campinas (PUC – Campinas)

alencar@saofrancisco.edu.br, mauricio@dca.fee.unicamp.br,
juan@puc-campinas.edu.br

Abstract. *Many applications that adapt themselves to available resources have lately become popular on the Internet, suggesting that absolute QoS levels are not essential for most applications. The Proportional Differentiation Model (PDM) constitutes currently one of the main service proposals for the Next Generation Internet and adds one more possibility for application adaptation. The present work investigates the PDM, situating it along other existing service proposals and presents the Window Waiting-Time Priority (WWTP) packet scheduling algorithm, which enables to reach the PDM objectives with higher accuracy than the traditional Waiting-Time Priority (WTP) and Proportional Average Delay (PAD) algorithms.*

Resumo. *Muitas aplicações que se adaptam aos recursos disponíveis na rede têm se tornado populares ultimamente na Internet, mostrando que níveis absolutos de QoS não são essenciais para a maioria das aplicações. O Modelo de Diferenciação Proporcional (MDP) constitui-se atualmente em uma das principais propostas de serviço para a próxima geração da Internet e pode acrescentar mais uma possibilidade ao espaço de dimensões disponíveis para a adaptação das aplicações. O presente trabalho investiga o MDP, situando-o diante de outras propostas de serviços existentes e propõe o algoritmo para escalonamento de pacotes Window Waiting-Time Priority (WWTP), o qual possibilita atingir os objetivos do modelo com maior precisão do que os tradicionais algoritmos Waiting-Time Priority (WTP) e Proportional Average Delay (PAD).*

1. Introdução

Tem se observado nos últimos anos a tendência das redes isoladas e especializadas, tais como as redes de telefonia, televisão e rádio, convergirem para a Internet, ocasionando um grande aumento de tráfego. O tráfego da Internet aumentou muito em volume, mas também sofreu grandes alterações em sua natureza. Com o surgimento das aplicações multimídia, os computadores passaram a processar mídias contínuas (áudio e vídeo); implementações de aplicações multimídia quando desenvolvidas sobre a Internet melhor-esforço não funcionam de modo adequado, devido principalmente aos grandes atrasos.

Os requisitos dos serviços de comunicação para aplicações multimídia excedem os serviços de comunicação atuais, orientados para dados, quanto ao suporte à Qualidade de Serviço (QoS). O termo QoS é utilizado neste trabalho para caracterizar a habilidade de qualquer elemento da rede em fornecer aos seus usuários serviços com um desempenho superior ao serviço melhor-esforço, levando em consideração os parâmetros típicos de QoS, tais como atraso na entrega dos pacotes, *throughput*, *jitter* (variação do atraso) e descarte de pacotes.

O escalonador de pacotes é um componente vital para qualquer arquitetura Internet orientada à *QoS*, sendo fundamental para assegurar requisitos de *QoS* aos diversos fluxos de pacotes servidos pelos roteadores. O objetivo principal deste trabalho é investigar o Modelo de Diferenciação Proporcional [1], principalmente nos aspectos relacionados ao escalonamento de pacotes, buscando o desenvolvimento de escalonadores que melhor atendam aos seus objetivos.

Na seção 2 discute-se os principais modelos de serviços propostos para as arquiteturas Internet orientadas à *QoS*, bem como aspectos relacionados ao escalonamento de pacotes. O Modelo de Diferenciação Proporcional é abordado na seção 3 e uma revisão das principais propostas para o escalonamento de pacotes no modelo é apresentada na seção 4. A principal proposta deste trabalho, o algoritmo WWTP, é descrito na seção 5. O modelo de simulação e os resultados obtidos são analisados na seção 6. Finalmente, as conclusões são apresentadas.

2. Serviços e Escalonamento de Pacotes em Arquiteturas Internet Orientadas à *QoS*

Para atender aos requisitos de *QoS* das novas aplicações para a Internet faz-se necessário empregar soluções mais sofisticadas do que os tradicionais escalonadores FIFO, principalmente quando as flutuações estatísticas das fontes de tráfego causam filas de tamanho considerável nos roteadores. O escalonamento de pacotes desempenha uma função fundamental em qualquer arquitetura Internet orientada à *QoS*, podendo ser classificado em duas famílias:

- **escalonamento nas arquiteturas Internet orientadas à *QoS* quantitativa:** nas arquiteturas deste tipo, por ex. Arquitetura de Serviços Integrados (IntServ) [2], são feitas reservas explícitas de recursos para cada um dos fluxos produzidos pelas aplicações, também chamados de micro-fluxos. O escalonador deve ser capaz de suportar aplicações do tipo tempo-real, oferecendo garantias determinísticas ou estatísticas de limite de desempenho para os micro-fluxos (atraso máximo etc.), requerendo reservas de recursos individuais para os mesmos;
- **escalonamento nas arquiteturas Internet orientadas à *QoS* qualitativa:** nestas arquiteturas, por ex. Arquitetura de Serviços Diferenciados (DiffServ) [3], não são feitas reservas para micro-fluxos individuais; os pacotes dos micro-fluxos individuais são classificados na borda do domínio em agregados de fluxos, sendo fornecidos poucos níveis de diferenciação de serviços. O escalonador normalmente deve suportar aplicações menos exigentes e adaptativas, procurando oferecer garantias relativas de *QoS* aos agregados de fluxos e também evitando a ocorrência de inanição de serviços em algum agregado.

Por ser mais escalável e apresentar um menor custo computacional, a arquitetura DiffServ tornou-se a proposta mais promissora para a próxima geração da Internet. Entre os primeiros serviços a serem propostos para a arquitetura DiffServ podem ser destacados o *Expedited Forward* (EF) [4] e o *Assured Forward* (AF) [5]. O serviço EF garante aos usuários uma largura de banda nominal ao longo de um certo caminho na rede, similar a uma linha privada, buscando minimizar o atraso, o *jitter* e o descarte de pacotes. As garantias de *QoS* oferecidas pelo serviço EF vão na linha da diferenciação de serviços do tipo absoluta, requerendo para isso algum tipo de reserva de banda e fixação de rotas; logo, o serviço EF pode ser pensado como uma extensão para o serviço Garantido [6] da arquitetura IntServ.

O serviço AF define quatro classes de serviço, cada uma com três níveis de precedência para descarte de pacotes. A diferenciação de serviço proposta pelo serviço AF pode ser vista como uma diferenciação relativa, onde os usuários que subscrevem uma classe de serviço mais alta obtêm níveis de *QoS* superiores aos usuários que subscrevem classes mais baixas. A reserva de recursos para o serviço AF não é requerida, contudo alguma forma de provisionamento ou garantias estatísticas para banda e buffer fazem-se necessário [5]. O entendimento e a especificação do serviço AF tem se mostrado até certo ponto vaga, o que certamente dificultou a oferta do serviço; em [7], Kilkki afirma que o modo como se dá o relacionamento entre as quatro classes AF é um assunto ainda aberto e o uso das mesmas é até certo ponto confuso. As colocações de Kilkki podem

ser comprovadas em diversos trabalhos; por exemplo, em [8], o serviço AF é classificado como sendo do tipo diferenciação absoluta e em [1], por sua vez, afirma-se que o serviço AF pode ser visto como um esquema de diferenciação de serviços relativa.

A seguir, discute-se um novo modelo de serviço proposto para a arquitetura DiffServ, o qual tem recebido bastante atenção na literatura. Isto se deve principalmente à clareza de sua especificação e à sua viabilidade de implementação, por meio de mecanismos escaláveis.

3. Modelo de Diferenciação Proporcional

O Modelo de Diferenciação Proporcional (MDP) foi proposto por Dovrolis em [1] e tem por objetivo fornecer um pequeno número de classes de serviço com garantias apenas para o ordenamento relativo do desempenho das classes, considerando parâmetros de *QoS* como atraso de fila e descarte de pacotes. O MDP não requer provisionamento de recursos e a fixação de rotas não é um aspecto importante. As principais características do MDP são:

- **controlabilidade:** do ponto de vista do operador da rede; o operador pode ajustar o espaçamento de *QoS* entre as classes de serviço, a partir de um conjunto de parâmetros;
- **previsibilidade:** do ponto de vista do usuário; deve-se procurar manter uma ordenação consistente entre as classes da *QoS*, conforme os parâmetros especificados pelo operador da rede.

De modo ideal, a controlabilidade e a previsibilidade devem ser mantidas independentemente da distribuição de carga nas diversas classes, a qual é variável, e da escala de tempo.

No MDP, a rede fornece a garantia de que uma classe de serviço C_i receberá melhor serviço, ou pelo menos serviço não pior, do que uma classe de serviço C_{i-1} em termos de métricas locais (*per-hop*) para atraso de fila e descarte. Considerando que $\delta_0 < \delta_1 < \delta_2 \dots$, com $\delta_0 = 1$, são os parâmetros de diferenciação de *QoS* especificados pelo operador da rede e que dm_i é o atraso médio da fila C_i , o objetivo do MDP pode ser expresso, para as n classes de serviço, da seguinte forma:

$$\delta_0 dm_0 = \delta_1 dm_1 = \dots = \delta_{n-1} dm_{n-1}. \quad (1)$$

As classes mais altas oferecem melhor desempenho aos usuários, sendo que a quantificação do desempenho depende da carga corrente em cada uma das classes. Nem sempre é factível atingir o objetivo, visto que pode-se perceber, até mesmo intuitivamente, que o atraso de cada classe possui um valor mínimo, relacionado com sua carga [9]. Em [10], Dovrolis estende o MDP, de modo a considerar o descarte de pacotes. O modelo proposto faz uso do método RED [11] e trata atrasos e descarte de modo acoplado, de tal modo que as classes mais altas possuem menores atrasos e taxas de descarte do que as classes mais baixas.

Os oito seletores de classe definidos pelo IETF em [12] estão em conformidade com o MDP. As aplicações e os usuários, ao usar serviços MDP, podem adaptar-se dinamicamente, escolhendo a classe de serviço que melhor atende suas necessidades. A classificação dos pacotes para as diferentes classes pode ser feita pela aplicação, pelo *host* origem ou pelos roteadores localizados na fronteira de domínios MDP. Uma política restritiva, baseada ou não em tarifação, deve ser implementada para impedir que todos os usuários escolham a mesma classe de serviço para atender seu tráfego.

A seguir, os principais algoritmos para escalonamento de pacotes no MDP são discutidos.

4. Escalonamento de Pacotes no Modelo de Diferenciação Proporcional

Em [1] mostra-se que abordagens tradicionais para obtenção de diferenciação de serviços, tais como escalonamento baseado em prioridades, tarifação e até mesmo o algoritmo *Weighted Fair Queueing* (WFQ) são inadequadas para o MDP. O WFQ [13] realiza diferenciação de serviço baseado em alocação de recursos, e é inadequado para assegurar que classes de serviço mais altas irão obter melhor serviço do que as classes de serviço mais baixas, independentemente da distribuição de

carga e escala de tempo, devido ao fato do tráfego Internet ser fortemente orientado a rajadas em qualquer escala de tempo. Resultados de simulação exibidos em [14] ilustram a falta de adequação do WFQ para o MDP.

O algoritmo *Waiting-Time Priority* (WTP) [1], proposto inicialmente por Kleinrock como *Time Dependent Priority Queuing* [15], foi o primeiro a ser estudado no contexto do MDP. As prioridades associadas aos pacotes pelo WTP aumentam proporcionalmente com o tempo de espera do pacote na fila, sendo que as prioridades das classes mais altas aumentam com um fator maior. O pacote com a maior prioridade é servido primeiro, de modo não preemptivo. A prioridade de um pacote na cabeça da fila i em um instante de tempo t é dada por:

$$p_i(t) = w_i(t)s_i \quad (2)$$

onde $w_i(t)$ é o tempo de espera na fila do pacote que se encontra na cabeça da fila i e s_i é o parâmetro de diferenciação, o qual é maior para as filas ou classes mais altas. Caso a taxa de utilização do enlace se aproxime de 100%, o algoritmo WTP pode atingir os objetivos do MDP [1][9], até mesmo em curtos intervalos de tempo, fazendo:

$$s_i = \delta_i. \quad (3)$$

Portanto, toda vez que um pacote vai ser transmitido, o escalonador WTP seleciona a classe j não vazia da seguinte forma:

$$j = \arg \max_{i \in B(t)} w_i(t)\delta_i(t) \quad (4)$$

onde $B(t)$ é o conjunto das filas que possuem pacotes a serem transmitidos no instante t .

Resultados de simulações exibidos em [1][9] mostram que o algoritmo WTP pode atingir com precisão o objetivo expresso em (1) apenas quando as taxas de utilização dos enlaces apresentam-se bastante altas, tipicamente acima de 90%. Diante disso, tem-se buscado novos algoritmos de escalonamento para o MDP, que possam atender aos seus objetivos de modo mais preciso, inclusive quando o grau de utilização dos enlaces não se encontrar em níveis elevados. Em [8] é apresentado um algoritmo iterativo baseado no método Gauss-Seidel destinado a obter os parâmetros de diferenciação s_i do algoritmo WTP a partir das cargas atuais de tráfego das classes de serviços. Os resultados obtidos com o método quando os enlaces não estão fortemente utilizados são superiores aos obtidos utilizando a relação (3), atendendo melhor aos objetivos expressos em (1); contudo, a complexidade da proposta torna impraticável sua utilização *on-line* em um ambiente sujeito a inúmeras variações como a Internet. Outros algoritmos seguem a linha de trabalhar com os parâmetros de diferenciação obtidos a partir de (3), mas buscando uma melhor caracterização para o atraso de cada classe, visto que o algoritmo WTP considera apenas os pacotes que se encontram nas cabeças das filas para tomar sua decisão, não levando em consideração, entre outros aspectos, o número total de pacotes que se encontra em cada uma das filas aguardando para serem transmitidos e o tempo gasto para a transmissão dos mesmos.

O algoritmo *Advanced WTP* (AWTP) [16] estende o algoritmo WTP considerando no cálculo das prioridades o tempo de transmissão dos pacotes que estão na cabeça de cada uma das classes, além do tempo de espera dos mesmos. Contudo, do mesmo modo que o WTP, o algoritmo AWTP não leva em consideração o número total de pacotes que aguardam transmissão em cada uma das filas.

O algoritmo *Proportional Average Delay* (PAD) [9] seleciona o pacote a ser enviado através de uma média dos atrasos de fila de todos os pacotes já transmitidos por cada uma das classes até o momento. Quando o escalonador PAD é empregado, uma classe com maior importância pode apresentar maiores atrasos do que uma classe com menor importância em curtos intervalos de tempo. Isso acontece em situações nas quais ocorrem muitas chegadas de pacotes a uma fila durante um intervalo de tempo em que a mesma não teve um pacote transmitido [9]. Neste caso, a média dos atrasos da fila não será atualizada, mas a fila terá acumulado grandes atrasos, o que irá comprometer a ordenação das classes de serviço em um futuro próximo. Em curtos intervalos de

tempo o algoritmo PAD atende apenas de modo razoável o objetivo MDP expresso em (1), visto que o mesmo busca equalizar o atraso médio normalizado de longo prazo das classes de serviço e não o atraso médio normalizado dos últimos pacotes transmitidos.

O escalonador *Hybrid Proportional Delay* (HPD) [9] é um algoritmo híbrido, resultante da combinação dos algoritmos WTP e PAD. Este algoritmo consegue um atendimento mais preciso dos objetivos do MDP em carga baixa e apresenta maior previsibilidade do que o algoritmo PAD, no entanto o mesmo depende de um parâmetro de configuração cujo processo de escolha, com vistas à otimalidade, não é discutido em [9]. Pode-se deduzir que o valor ideal deste parâmetro possui uma estreita relação com a carga atual em cada uma das classes, o que torna a sua obtenção bastante complexa.

Na próxima seção será proposto um novo algoritmo de escalonamento de pacotes para o MDP, o qual busca uma caracterização mais precisa para os atrasos das classes de serviço do que os algoritmos aqui discutidos, com o objetivo de melhor atender aos requisitos do MDP.

5. Algoritmo de Escalonamento WWTP

Consideraremos que a prioridade de um pacote na cabeça da fila j em um instante de tempo t será fornecida por:

$$p_j(t) = WW_j(t)\delta_j \quad (5)$$

onde $WW_j(t)$ é o tempo médio de espera na fila de pacotes j , estimado a partir de uma janela de medida constituída por diversos pacotes a serem transmitidos e/ou já transmitidos e δ_j é o parâmetro de diferenciação da fila j .

O algoritmo PAD considera apenas pacotes já transmitidos e possui uma memória infinita, tornando-se pouco sensível às alterações recentes nas distribuições de carga das classes. Um bom estimador para $WW_j(t)$ deve possuir as seguintes características: 1) trabalhar com uma janela de medida de tamanho limitado, de modo a se tornar mais sensível às alterações nas distribuições de carga das classes e 2) ser atualizado a cada chegada e a cada partida de pacotes, de modo a evitar que uma classe de serviço mais importante possa vir a sofrer maiores atrasos do que uma classe de menor importância. Em [17] é proposto um estimador de atraso que possui estas características. Para cada classe de serviço é definida uma janela de medida para armazenar informações relativas às chegadas e partidas mais recentes dos pacotes. A janela é implementada como uma lista circular com dois ponteiros: hw , que indica a cabeça da janela, posição onde um pacote que chega é armazenado, sobrescrevendo a informação anterior e o ponteiro hq , que indica a cabeça da fila, ou seja, a posição do próximo pacote a ser transmitido. Quando a fila está vazia os ponteiros hw e hq permanecem alinhados. A informação armazenada consiste, para os pacotes a serem transmitidos, de carimbos de tempo do instante em que os pacotes entraram na fila; para os pacotes já transmitidos armazena-se o seu tempo de espera na fila.

Para cada janela W_j de tamanho $\|W_j\|$, pacote i , $p_j^i \in W_j$, é usada a seguinte notação: ts_j^i é o carimbo de tempo do pacote i , d_j^i é o tempo de espera do pacote i , B_j é o conjunto de pacotes que aguarda transmissão em W_j e n_j é o tamanho de B_j . O tempo médio de espera para a classe de serviço ou janela j é fornecido por:

$$WW_j(t) = \frac{1}{\|W_j\|} \sum_{i \in W_j} d_j^i = \frac{1}{\|W_j\|} \left(\sum_{i \in B_j} d_j^i + \sum_{i \notin B_j} d_j^i \right)$$

$$WW_j(t) = \frac{1}{\|W_j\|} \left(n_j \cdot t - \sum_{i \in B_j} ts_j^i + \sum_{i \notin B_j} d_j^i \right)$$

$$WW_j(t) = \frac{1}{\|W_j\|} (n_j \cdot t - S_j + D_j^{dep}) \quad (6)$$

onde $j = 0, \dots, n-1$, considerando n classes de serviço e $i = 0, \dots, \|W_j\| - 1$. Os ponteiros e as variáveis são atualizados sempre que um pacote chega ou é transmitido de cada uma das classes, conforme os procedimentos exibidos na Figura 1.

O escalonador que faz uso do estimador (6) será chamado de **Window Waiting-Time Priority (WWTP)**.

Chegada de pacotes	Partida de pacotes
$n_j \leftarrow n_j + 1$	$n_j \leftarrow n_j - 1$
$S_j \leftarrow S_j + clock$	$S_j \leftarrow S_j - ts_j^{hq}$
$D_j^{dep} \leftarrow D_j^{dep} - d_j^{hw}$	$d_j^{hq} \leftarrow clock - ts_j^{hq}$
$hw \leftarrow (hw + 1) \bmod \ W_j\ $	$D_j^{dep} \leftarrow D_j^{dep} + d_j^{hq}$
	$hq \leftarrow (hq + 1) \bmod \ W_j\ $

Figura 1 – Procedimentos realizados após a chegada e a partida de pacotes

O objetivo do MDP expresso em (1) pode ser reescrito da seguinte forma:

$$|\delta_i dm_i - \delta_j dm_j| \rightarrow 0, \forall i, j \quad (7)$$

Logo, uma medida para o nível de atendimento da *QoS* no MDP pode ser definida do seguinte modo:

$$NívelQoS = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} |\delta_i dm_i - \delta_j dm_j| \quad (8)$$

Quanto mais próximos de zero estiverem os valores de *NívelQoS*, melhor será o atendimento do objetivo expresso em (1).

6. Simulação

Com o objetivo de avaliar o algoritmo WWTP proposto, realizou-se simulações empregando o simulador LBNL NS [18] (pacote “tudo-em-um”, versão 2.27), modificado para suportar os algoritmos de escalonamento WTP, PAD e WWTP, bem como os aspectos relacionados à medida do nível de *QoS*. O modelo de simulação e os resultados obtidos são apresentados a seguir.

6.1. Modelo de Simulação

O domínio DiffServ simulado encontra-se representado na Figura 2, onde *RB1* e *RB2* são roteadores de borda do domínio DiffServ, *RN* é um roteador do núcleo do domínio, *S0*, *S1* e *S2* são os nós aos quais estão associadas as fontes de tráfego e *D* é o nó destinatário do tráfego gerado. O atraso de propagação e a capacidade de todos os enlaces encontram-se indicados próximos dos mesmos, sendo que os três enlaces que interconectam os nós *S0*, *S1* e *S2* ao roteador de borda *RB1* possuem características idênticas.

Associou-se a cada um dos nós *S0*, *S1* e *S2* uma fonte de tráfego do tipo Pareto com as seguintes características:

- nó $S0$: tráfego pertencente à classe C_0 (50% do tráfego total pertence a esta classe), tempo de rajada = 500 ms, tempo de ociosidade = 500 ms e $\alpha = 1,3$;
- nó $S1$: tráfego pertencente à classe C_1 (30% do tráfego total pertence a esta classe), tempo de rajada = 500 ms, tempo de ociosidade = 500 ms e $\alpha = 1,3$;
- nó $S2$: tráfego pertencente à classe C_2 (20% do tráfego pertence a esta classe), tempo de rajada = 750 ms, tempo de ociosidade = 250 ms e $\alpha = 1,3$.

Os parâmetros de diferenciação utilizados foram os seguintes: $\delta_0 = 1$, $\delta_1 = 2$ e $\delta_2 = 4$. A distribuição de Pareto é caracterizada por uma variabilidade extrema; valores típicos para o parâmetro α na Web se situam entre 0,8 e 1,3 [19].

O tráfego gerado nos nós $S0$, $S1$ e $S2$ é marcado com os marcadores DiffServ 0, 1 e 2, respectivamente. No roteador de borda $RB1$ o policiamento do tráfego é realizado por meio de um policiador *Token-Bucket* [20] configurado de modo a não alterar a marcação DiffServ original dos pacotes. Os enlaces internos ao domínio DiffServ são configurados com 3 filas físicas e 1 nível de precedência cada, todas com tamanho máximo igual a 100 pacotes e política de descarte *droptail*. No roteador RN , o marcador DiffServ de cada pacote identifica a classe e a fila física na qual o mesmo será inserido; por exemplo, o pacote com marcador DiffServ 0 pertence à classe C_0 e será inserido na fila física 0. Utilizou-se um tamanho de pacote igual a 600 bytes.

As três fontes de tráfego utilizam 85% da capacidade do enlace que interconecta os roteadores RN e $RB2$; a opção por esse valor se deve ao fato de que quando a taxa de utilização atinge valores de até 70%, o desempenho do algoritmo WTP desvia-se bastante dos objetivos expressos em (1) e quando a taxa de utilização do enlace aproxima-se de 100%, o algoritmo WTP alcança os objetivos com bastante precisão [9]. Portanto, a taxa de ocupação do enlace de 85% representa o ponto médio entre estes dois extremos, possibilitando um cenário mais realista para a comparação dos três algoritmos.

No algoritmo WWTP, o tamanho da janela de medida utilizado foi de 400 pacotes.

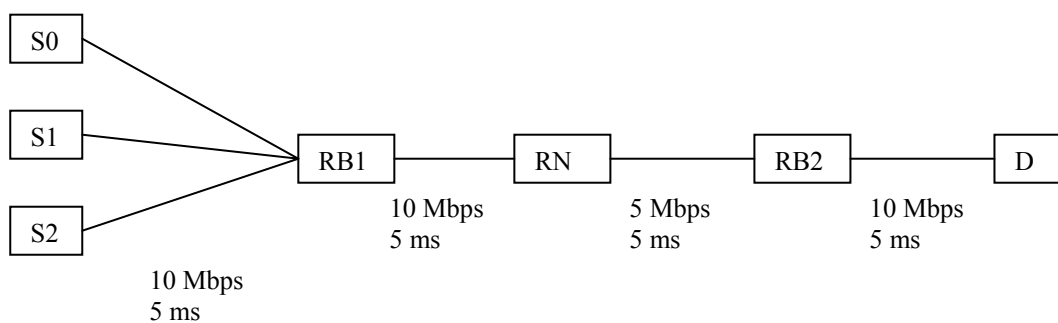


Figura 2 – Topologia do domínio DiffServ simulado

6.2. Resultados da Simulação

Foram realizadas duas sessões de simulação para o cenário descrito anteriormente. Cada sessão consiste em executar três experimentos, apenas alterando a cada experimento o escalonador para o enlace situado entre os roteadores RN e $RB2$. A parte (a) da Figura 3, exibe os resultados obtidos na primeira sessão, onde cada série de pacotes corresponde a 10.000 pacotes, a partir da qual são calculados os atrasos de fila e o valor correspondente de *NívelQoS*; a parte (b) da Figura 3 mostra os resultados obtidos na segunda sessão, na qual *NívelQoS* é calculado a partir de uma série constituída por 2.000 pacotes.

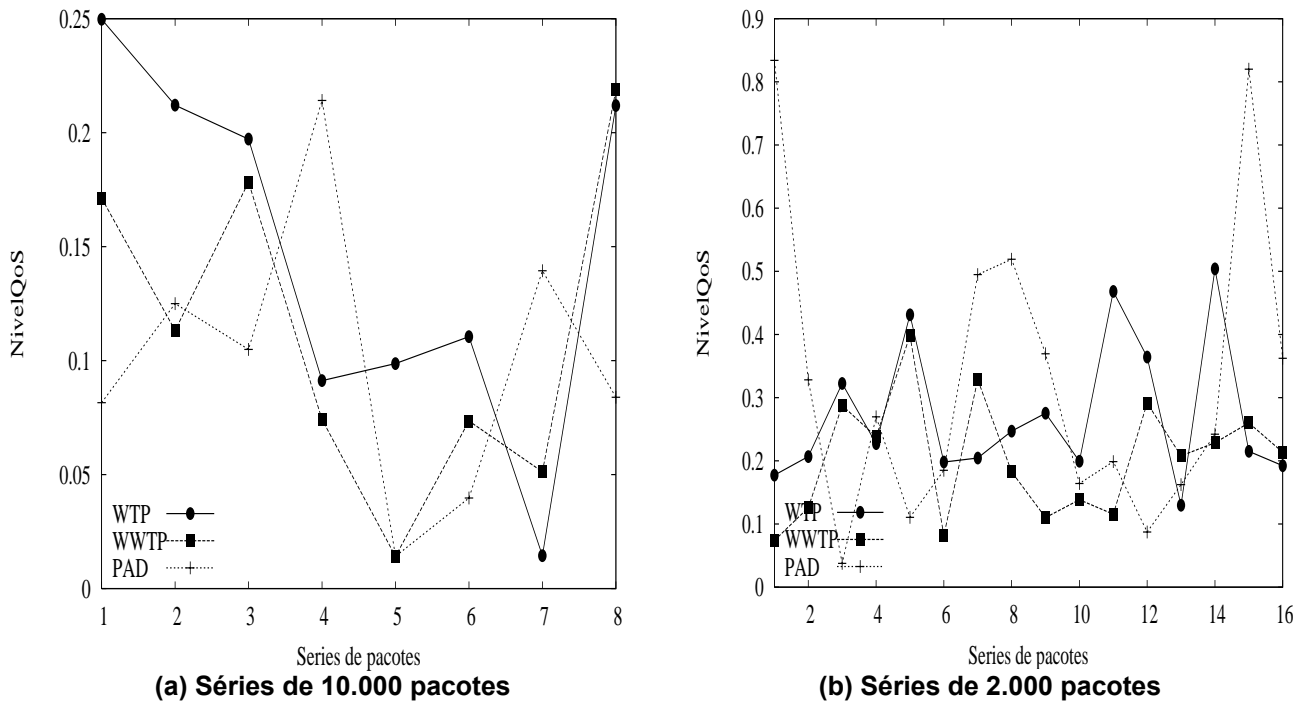


Figura 3 – Alguns resultados obtidos com a simulação

Os resultados obtidos confirmam que quanto menor for a escala de tempo, mais difícil torna-se atender aos objetivos do MDP, visto que os valores de *NivelQoS* são bem maiores quando calculados para as séries constituídas por 2.000 pacotes do que para as séries constituídas por 10.000 pacotes. O algoritmo WWTP apresentou um bom desempenho nas duas sessões de simulação, contudo para as séries de 2.000 pacotes o seu desempenho superou com folga os algoritmos WTP e PAD, mostrando uma maior previsibilidade em intervalos de tempo menores, ou seja, uma ordenação mais consistente entre as classes de serviço. Pode ser observado também que o desempenho do algoritmo PAD tornou-se um tanto quanto instável em curtos intervalos de tempo, apresentando valores bem mais elevados para *NivelQoS* em determinados momentos, o que contribui para o aumento do *jitter*, devido ao fato de seu estimador de atraso trabalhar com memória infinita, tornando-o pouco sensível às alterações mais recentes de carga nas classes de serviço. Portanto, os resultados iniciais obtidos indicam possivelmente a superioridade do algoritmo WWTP sobre os algoritmos WTP e PAD quando a taxa de utilização do enlace encontra-se ao redor de 85%, um cenário bastante realista para a comparação dos três algoritmos, principalmente quando são considerados intervalos de tempo mais curtos.

7. Conclusões

A popularidade crescente de aplicações adaptativas na Internet (telefonia, rádio etc.) usando a infraestrutura melhor-esforço original mostra que níveis absolutos de *QoS* não são essenciais para a maioria das aplicações. O MDP acrescenta mais uma possibilidade ao espaço de dimensões disponíveis para a adaptação das aplicações. Devido à clareza de sua especificação e à viabilidade de sua implementação, o MDP constitui-se atualmente em uma das principais propostas de serviço para a próxima geração da Internet. Neste contexto, o algoritmo proposto (WWTP) contribui para o esforço de viabilização do MDP e para a oferta de diferenciação de serviços aos usuários da Internet.

Os resultados iniciais das simulações realizadas são encorajadores e sugerem que as soluções propostas pelo algoritmo WWTP para as duas principais deficiências do algoritmo PAD são viáveis: ao empregar uma janela de medida de tamanho limitado torna-se mais sensível às alterações recentes nas distribuições de carga das classes e ao ter seu estimador de atraso atualizado

a cada chegada e a cada partida de pacotes, evita que classes de mais alta importância apresentem em alguns momentos atrasos superiores aos de classes de menor importância. Portanto, a estratégia de desenvolver escalonadores de pacotes para o MDP baseados em janelas de medidas mostrou-se bastante promissora.

Pretendemos na seqüência dos trabalhos considerar outras variáveis no processo de estimação de atraso do algoritmo WWTP, bem como empregar algumas das idéias para obtenção de escalonadores de pacotes apresentadas em [21], baseadas em aprendizado de máquina, buscando obter escalonadores para o MDP que venham a apresentar uma aderência aos seus objetivos ainda maior.

8. Referências

- [1] Dovrolis, C. & Ramanathan, P. A Case for Relative Differentiated Services and the Proportional Differentiation Model, *IEEE Network Magazine*, vol. 13, pp. 26-34, Sept./Oct. 1999.
- [2] Braden, R., Clark, D. & Shenker, S. Integrated Services in the Internet Architecture: an Overview, Network Working Group, RFC 1633, NIC, June 1994.
- [3] Blake, S. et al. An Architecture for Differentiated Services, RFC 2475, IETF DiffServ Working Group.
- [4] Jacobson, V., Nichols, K. & Poduri, K. An Expedited Forwarding PHB, RFC 2598, June 1999.
- [5] Heinanen, J., Baker, F., Weiss, W. & Wroclawski, J. Assured Forwarding PHB Group, RFC 2597, June 1999.
- [6] Shenker, S., Partridge, C. & Guerin, R. Specification of Guaranteed Quality of Service, RFC 2212, September 1997.
- [7] Kilkki, K. Differentiated Services for the Internet, USA: Macmillan Technical Publishing, 1999.
- [8] Leung, M.K.H., Lui, J.C.S. & Yau, D.K.Y. Characterization and Performance Evaluation for Proportional Delay Differentiated Services, *Proceedings of the 2000 International Conference on Network Protocols*, Osaka, Japan, November 14-17, 2000.
- [9] Dovrolis, C., Stiliadis, D. & Ramanathan, P. Proportional Differentiated Services: Delay Differentiation and Packet Scheduling, *IEEE/ACM Transactions on Networking*, vol. 10, no. 1, pp. 12-26, February 2002.
- [10] Dovrolis, C. & Ramanathan, P. Proportional Differentiated Services, Part II: Loss Rate Differentiation and Packet Dropping, *IEEE/IFIP International Workshop Quality of Service (IWQoS)*, June 2000, pp.52-61.
- [11] Floyd, S. & Jacobson, V. Random Early Detection Gateways for Congestion Avoidance, *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397-413, August 1993.
- [12] Nichols, K., Blake, S., Baker, F. & Black, D. L. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, IETF RFC 2474, Dec. 1998.
- [13] Parekh, A.K. & Gallager, R.G. A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks – The Multiple Node Case. *IEEE/ACM Transactions on Networking*, pp. 137-150, April 1994.
- [14] Dovrolis, C. & Stiliadis, D. Relative Differentiated Services in the Internet: Issues and Mechanisms, *ACM SIGMETRICS*, May 1999.
- [15] Kleinrock, L. A Delay Dependent Queue Discipline, *Journal of the ACM*, vol. 14, no. 2, pp. 242-261, 1967.
- [16] Lai, Y.C. & Li, W.H. A Novel Scheduler for Proportional Delay Differentiation by Considering Packet Transmission Time, *IEEE Communications Letters*, vol. 7, no. 4, April 2003.

- [17] Salles, R.M. & Barria, J.A. Utility-Based Scheduling Disciplines for Adaptive Applications Over the Internet, IEEE Communications Letters, vol. 6, no. 5, May 2002.
- [18] NS – Network Simulator Version 2, available at <http://www.isi.edu/nsnam/ns>
- [19] Crovella, M.E. & Bestavros, A. Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes, ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp. 160-169, 1996.
- [20] Keshav, S. An Engineering Approach to Computer Networking: ATM Networks, the Internet, and the Telephone Network, Addison-Wesley, 1997.
- [21] Melo Jr., A. & Adán Coello, J.M. Packet Scheduling Based on Learning in the Next Generation Internet Architectures, Proceedings of The Fifth IEEE Symposium on Computers and Communication, pp. 673-678, Antibes-Juan Les Pins, France, July 3-7, 2000.