Nova Estrutura de Adaptação de Taxa de Transmissão para uma Arquitetura de Serviços Diferenciados Escalável

Cleison Caetano dos Santos e Lee Luan Ling

***Resumo*— Em [1] foi proposta uma arquitetura, baseada na filosofia DiffServ, onde roteadores do centro do domínio não necessitam implementar funções específicas de diferenciação de serviço. Neste trabalho, propomos uma diferente estrutura de adaptação de taxa de transmissão para essa arquitetura, que diminui, para quase zero, a perda de pacotes no interior do domínio.**

***Palavras-Chave*—QoS, DiffServ, Escalonamento, Controle em Malha Fechada.**

***Abstract*— In [1] has been proposed an architecture that fulfills the design philosophy of the DiffServ, where core routers do not need to support any specific function for service differentiation. In our work, we consider a different structure of rate adaptation that decrease, for almost zero, the packets loss in the core of the domain.**

***Index Terms*—QoS, DiffServ, Schedule, Feedback Control.**

I. INTRODUÇÃO

A Internet, como foi concebida, caracteriza-se por ser uma rede IP que atende as requisições das aplicações e as encaminham num esquema melhor esforço. Esse modelo apresenta grande simplicidade e robustez, motivo do seu sucesso. No entanto, por si só, não permite o desenvolvimento de aplicações avançadas e a diferenciação de serviços entre usuários. O tráfego de grandes quantidades de informações de voz, vídeo, imagem e dados, faz com que o suporte à qualidade de serviço (QoS) torne-se indispensável na Internet, de forma que se possa controlar melhor este tráfego.

Soluções como o modelo IntServ [2] , DiffServ [3], MPLS [4], engenharia de tráfego [5] e roteamento baseado em restrições [6], tentam empregar QoS de diferentes aspectos. É evidente que para conseguir QoS, todos estes modelos inserem na rede um determinado nível de complexidade. Como e onde esta complexidade é inserida determinam o quão escalável é a solução.

Cleison Caetano dos Santos e Lee Luan Ling, Departamento de Comunicações, Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, Brasil, E-mails: cleison@decom.fee.unicamp.br, lee@decom.fee.unicamp.br.

Decisões complexas sobre o comportamento do tráfego, tomadas no interior da rede, afetam de forma significativa o seu desempenho bem como a sua escalabilidade, aumentando os custos e complexidade.

A arquitetura DiffServ posiciona-se entre os extremos do melhor esforço, sem priorização, e serviços com reserva de recursos e grande sobrecarga de sinalização. Nessa abordagem, uma porção do tráfego é tratada de forma privilegiada sobre o restante.

De acordo com a própria arquitetura DiffServ, por motivo de escalabilidade, as funções de um roteador de núcleo de um domínio resumem-se à atribuição do pacote à classe que lhe proverá o devido serviço e o encaminhamento do mesmo, deixando as tarefas que exigem maior processamento para os roteadores de borda. Contudo, essa filosofia concentra-se em garantir qualidade de serviço em cada nó da rede, supondo-se que a garantia individual proporciona a qualidade no domínio. Assim, todos os nós devem ter uma fila distinta para cada classe, um classificador que coloca o pacote na respectiva fila, bons algoritmos para tratamento de filas e um escalonador que retira pacotes das filas.

C. L. Lee et. al. propuseram uma nova e escalável arquitetura seguindo a mesma filosofia de projeto do modelo DiffServ [1]. Porém, nesse novo modelo proposto, os roteadores de borda mantêm um estado por fluxo agregado enquanto que os roteadores do interior da rede não sofrem modificações. Diferente de outras arquiteturas existentes [7, 8, 9], esse modelo coloca toda a complexidade de diferenciação de serviço na borda da rede. Assim, no núcleo, os roteadores simplesmente encaminham os pacotes para o seu destino num esquema melhor esforço, não necessitando suportar qualquer função específica de diferenciação de serviço. As garantias de QoS dos fluxos agregados são conseguidas controlando a ocupação das filas no interior da rede.

Neste trabalho, analisamos, através de simulações, a arquitetura proposta em [1] em relação a perda desordenada de pacotes no interior do domínio. Isto fere a filosofia DiffServ quanto ao controle de descarte de pacotes. Propomos aqui uma nova estrutura de adaptação de taxa de transmissão capaz de obter controle do tipo amortecido e dessa forma diminuir, para quase zero, a perda de pacotes no

interior da rede. Assim, toda a funcionalidade DiffServ, incluindo a função de controle de descarte de pacotes, passa a ser executada somente na borda.

A seção II descreve resumidamente a arquitetura proposta, maiores detalhes podem ser encontrados em [1]. Na seção III discutimos a nova estrutura de controle de taxa a qual propomos. Na seção IV apresentamos os resultados das simulações. Concluímos o trabalho na seção V.

II. A Arquitetura Proposta

A arquitetura proposta é um mecanismo de controle de tráfego agregado implementado na terceira camada da arquitetura TCP/IP e concentra-se no processo de encaminhamento dos pacotes IP. A Figura 1 mostra esta arquitetura: um conjunto contíguo de nós DS que aplicam um conjunto comum de políticas sobre o tráfego que atravessa o domínio [3].

De acordo com a filosofia da arquitetura DiffServ, os múltiplos domínios DS negociam contrato de serviço (SLA) que visam a garantia mínima de QoS para aplicações dos usuários [3]. Todos os pacotes são policiados nos roteadores de borda para verificar sua conformidade com o SLA.

A. Componentes

Essa arquitetura define três importantes componentes nos nós de borda: o componente de classificação, o componente de condicionamento de tráfego e o componente de controle de ocupação das filas no interior da rede. Os componentes internos a rede são bem mais simples do que os de borda, já que não requer nenhuma funcionalidade especial. O escalonamento de pacotes nesses roteadores é feito com o mais simples e comum FIFO. Este aspecto favorece fortemente a escalabilidade da arquitetura.

A1. Componentes de Classificação

A classificação divide o fluxo de entrada em um conjunto de fluxos de saída por meio de filtros de tráfego baseados ou em vários campos contidos num pacote IP, ou no campo DSCP [3]. Depois de um pacote ser classificado dentro de um fluxo, um peso lhe será atribuído. Esse peso determina o processamento relativo que um pacote receberá no domínio.

A2. Componentes de Condicionamento

Condicionadores podem ser implementados através da combinação dos seguintes componentes DiffServ:

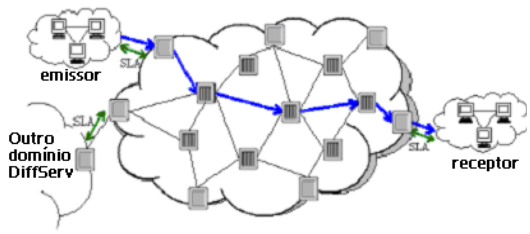

Fig. 1. A arquitetura proposta

1) *Medição*: Esta funcionalidade mede as propriedades temporais do tráfego previamente classificado. Esta informação, confrontada com os perfis de tráfego especificados no SLA, é utilizada para regular o tráfego conforme este esteja dentro ou fora do perfil acordado.

2) *Policiamento*: garante que o tráfego está de acordo com um perfil específico, eliminando ou remarcando para prioridades mais baixas o tráfego não conforme.

3) *Shaping*: Esta funcionalidade regula o tráfego que é submetido a um domínio DS. O *Shaping* pode atrasar os pacotes de forma a ajustar as suas características ao perfil definido.

A3. Componente de Controle

Para cada fluxo, o roteador de ingresso insere um pacote de controle de 32 bytes depois que NCTR pacotes de dados tenham sido enviados. Quando um roteador de egresso recebe um pacote de controle, ele envia esse pacote para o roteador de ingresso, de acordo com o seu endereço fonte.

Definimos ciclo de controle como sendo o intervalo de tempo em que um pacote de controle estará na rede. O início do ciclo é marcado pelo envio deste pacote, e o fim é marcado pela sua chegada no mesmo roteador de ingresso que o enviou. Denominamos *Forward Packet Control* (FCP) o pacote de controle quando este está sendo envido do roteador de ingresso para o roteador de egresso, e de *Backward Packet Control* (BCP), quando faz o caminho inverso.

Um roteador de ingresso pode estimar a ocupação da fila de um fluxo para cada BCP recebido baseado nas informações contidas nesse pacote. A partir daí, é executando um algoritmo de adaptação de taxa discutido em *B*. Uma vez que um roteador de ingresso recebe um BCP e executa o algoritmo de adaptação de taxa, ele envia um outro FCP, mantendo sempre um pacote de controle na rede.

O parâmetro NCTR determina a freqüência em que os pacotes de controle serão inseridos no fluxo. A adaptação de taxa é feita a cada ciclo de controle. Assim, é selecionado apenas um único pacote de controle para a verificação do estado da rede, mesmo que sejam recebidos múltiplos BCP em um ciclo. O principal objetivo de se transmitir vários pacotes de controle em um ciclo é fazer a arquitetura robusta a congestionamento, o que poderia causar perda de pacote no interior da rede.

B. Diferenciação de Serviços

Nesta arquitetura, n fluxos competem por uma limitada taxa de serviço. Cada fluxo obtém uma taxa compartilhada que é proporcional a sua ocupação na fila. A Figura 2 mostra um modelo simples de um fluxo i onde os roteadores do interior da rede são representados por um único roteador FIFO.

Fig. 2 – Modelo para um fluxo

Seja t_d o atraso de um pacote na fila do fluxo i, e u_i a taxa de serviço compartilhada. De acordo com o resultado de Little [10], a ocupação da fila para esse fluxo pode ser deduzida como:

$$q_i = u_i \times t_d . \quad (1)$$

Dessa equação podemos perceber que se o roteador de ingresso é capaz de obter o valor de t_d e u_i, então ele pode estimar a ocupação da fila do fluxo i.

O tempo de ida e volta (rtt) de um pacote consiste do atraso constante referente a propagação $rtt_{\min}$ e do atraso variável t_d referente ao tempo de processamento nos nós do interior da rede. O valor de $rtt_{\min}$ é assumido como sendo o menor rtt medido e esse valor é sempre atualizado. Então o atraso de um pacote na fila t_d pode ser obtido subtraindo o rtt medido do $rtt_{\min}$.

O roteador de borda de um fluxo deve medir o rtt periodicamente para atualizar o estado da rede. Isso é feito a cada chegada do primeiro BCP de um ciclo. A taxa de serviço compartilhada também é calculada para cada primeiro BCP recebido, baseada nos dados contidos nesse pacote:

$$u_i = (\lambda_i . N_{ctrl}) / (N_{ctrl} + \lambda_i . t_d) . \quad (2)$$

Assim, um roteador de borda pode estimar a ocupação da fila de um determinado fluxo para cada BCP recebido.

Seja w_i o peso associado ao fluxo i. Se a razão esperada das taxas compartilhadas dos m fluxos é $w_1 : w_2 : \ldots : w_m$, essa razão pode ser conseguida mantendo a ocupação das filas desses m fluxos no interior da rede nessa mesma razão. Isto significa que um roteador de borda pode conseguir a taxa compartilhada de um fluxo controlando uma certa quantidade Q extra de dados no interior da rede.

É fácil observar que se mantivermos a taxa de envio de pacotes λ_i da fonte por um intervalo de tempo Δt, alcançaremos o objetivo de ocupação da fila desde que:

$$\lambda_i . \Delta t + q_i - u_i . \Delta t = Q . \quad (3)$$

Observe que o Δt pode ser qualquer. Podemos portanto, definir que o objetivo de ocupação será atingido no final do ciclo de controle.

Substituindo (1) em (3), fazendo Δt igual ao novo rtt do BCP que está chegando e fazendo algumas manipulações algébricas, chegaremos à equação utilizada para o cálculo da nova taxa:

$$\lambda_i = u_i . (rtt_{\min} + Q_i / u_i) / rtt_{new} . \quad (4)$$

III. Adaptação de Taxa Proposta

Como discutido anteriormente, a cada ciclo o algoritmo de adaptação de taxa é executado. Isso é feito a cada chegada do primeiro BCP de um ciclo. Em [1], a fim de calcular a taxa de transmissão, os autores fizeram Δt igual ao rtt extraído do novo BCP que acabou de chegar. Eles assumiram, com isso, que o pacote de controle a ser enviado gastará exatamente o mesmo tempo de ida e volta gasto pelo pacote de controle do ciclo anterior. Foi assumido também que o objetivo de ocupação da fila será atingido ao final do ciclo atual, ou seja, no momento em que o BCP, correspondente ao FCP que está sendo enviado, estiver chegando.

Um dos problemas deste método, como mostra os resultados das simulações na seção IV, é que a taxa de transmissão estimada oscila largamente em torno do valor de objetivo, ciclo após ciclo. Esta oscilação somente pode ser amortecida sacrificando o desempenho da rede, isto é, descartando pacotes no núcleo, sem levar em consideração as prioridades de descarte.

Nesta seção, propomos uma nova alternativa para a adaptação de taxa capaz de eliminar satisfatoriamente a oscilação da taxa de transmissão. Assim, conseguimos eliminar totalmente a perda de pacotes em poucos ciclos. O intervalo de tempo desses ciclos nos quais temporariamente a taxa oscila e ocorrem perdas de pacotes é chamado de estado transiente. O intervalo de tempo seguinte, no qual a taxa de transmissão segue continuamente o mesmo padrão no tempo, é chamado de estado estacionário.

A principal característica desta nova estrutura é o preciso processo de predição baseado no fato de que o objetivo de ocupação será atingindo somente no final do próximo ciclo de controle. Sob essa observação, é possível deduzir a ocupação da fila ao final do ciclo atual e então, estimar o rtt do pacote de controle do próximo ciclo. Com isso, a nova taxa de transmissão pode ser calculada de maneira mais precisa, tornando a oscilação menor possível.

A ocupação da fila ao final de um ciclo é calculada como:

$$q_n = \lambda_n . \Delta t - u_n . \Delta t , \quad (5)$$

onde λ_n, u_n e Δt são a taxa de transmissão, a taxa de serviço da fila durante o ciclo n e a duração do ciclo n, respectivamente.

Neste trabalho, para supor o valor da ocupação da fila ao final do ciclo, fizemos Δt igual ao rtt do pacote de controle recebido e calculamos u_n com base nas informações contidas nesse pacote, conforme (2). Consideramos assim que o FCP enviado no ciclo atual (n) encontrará o estado da rede com as mesmas condições do ciclo anterior (*n-1*).

O rtt do pacote de controle do próximo ciclo pode ser calculado desde que a ocupação da fila ao final do ciclo atual tenha sido estimada. rtt_{n+1} é dado por:

$$rtt_{n+1} = (\lambda_n . rtt_{new} - u_n . rtt_{new}) / u_n + rtt_{\min} . \quad (6)$$

O que esperamos conseguir aqui é uma situação de controle amortecido, onde a taxa de transmissão, após um regime transitório, permaneça praticamente constante e igual ao valor de objetivo. Matematicamente, isto significa que:

$$\lambda_{n+1} = \lambda_n , \quad (7)$$

$$u_{n+1} = u_n \,. \tag{8}$$

Reescrevendo (3) para o próximo ciclo, temos:

$$\lambda_{n+1}.rtt_{n+1} + q_n - u_{n+1}.rtt_{n+1} = Q \,. \tag{9}$$

Substituindo (5), (6), (7) e (8) em (9) e fazendo algumas manipulações algébricas, chegaremos a uma nova equação de adaptação de taxa em que o controle é do tipo amortecido:

$$\lambda = \frac{u.t_d + \sqrt{(u.t_d)^2 + 4.rtt.u.(Q + u.rtt_{\min})}}{2.rtt}. \tag{10}$$

IV. SIMULAÇÕES

Para comparar o comportamento do sistema ao utilizarmos os algoritmos de controle de adaptação de taxa (4) e (10), reproduzimos o primeiro cenário de simulação apresentado em [1]. Para tanto usamos o simulador Opnet versão 7.0 [11].

A configuração deste simples cenário é mostrada na Figura 3, onde dez fontes enviam pacotes durante 1000 segundos, através de um link comum com capacidade limitada. Assumimos que as fontes sempre têm dados a serem enviados e o tamanho dos pacotes de dados é fixo e igual a 1k bytes. NCTRL foi configurado para 10. A Tabela I lista o peso e a taxa de transmissão ideal para cada conexão.

A. Experiência 1: Buffer de Tamanho Infinito

Nessa primeira situação, estabelecemos o objetivo total de ocupação do *buffer* como sendo igual a 100 pacotes. Porém, mantivemos os roteadores de centro com tamanho de *buffer* infinito.

A Figura 4 mostra como a taxa de transmissão do fluxo 1 (*f1*), observada no Roteador E_1, varia de ciclo em ciclo quando (4) é usado para a estimação de taxa de transmissão. Observe que este mecanismo de controle é incapaz de manter a taxa de transmissão estável. Esta taxa oscila com grande amplitude em torno da taxa ideal.

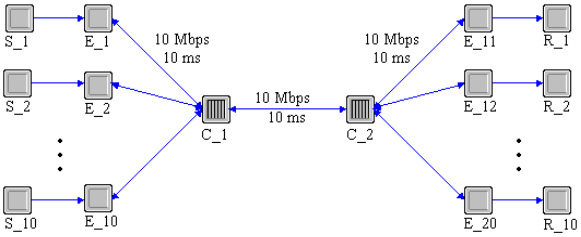

Fig. 3. Cenário de simulação

TABELA I
PARAMETROS DA SIMULAÇÃO

Fluxo	*f1 f6*	*f2 f7*	*f3 f8*	*f4 f9*	*f5 f10*
Peso(w_i)	1	2	3	4	5
Taxa Ideal	333.33	666.66	999.99	1333.33	1666.66

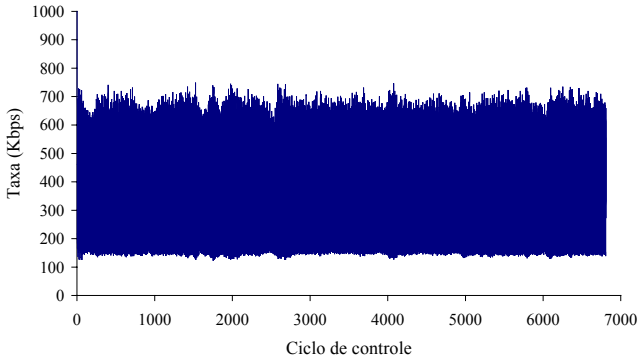

Fig. 4. Taxa do fluxo 1 vs ciclo de controle

Por outro lado, a Figura 5 mostra a taxa de transmissão conseguida pelos 10 fluxos (*f1* a *f10*) quando (10) é aplicado para a atualização da taxa. Através dessa figura é possível observar que o controle estabelecido por esse novo algoritmo é do tipo amortecido, isto é, a taxa de transmissão de cada fluxo, após o regime transitório, tende ao seu valor de objetivo, mantendo-se constantemente bem próximo a esse valor.

A Tabela II resume a taxa média e o desvio padrão para cada fluxo quando (4) e (10) é aplicado.

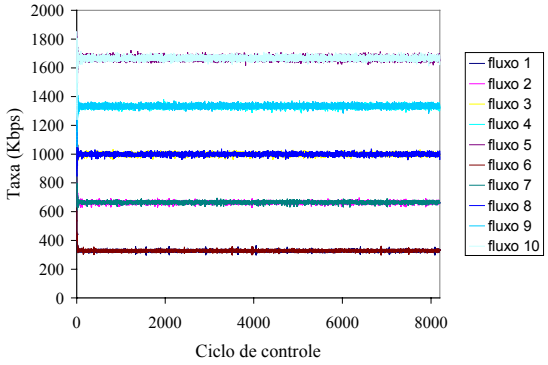

Fig. 5. Taxa dos fluxos vs ciclo de controle

TABELA II
MEDIA E DESVIO PADRÃO

	Equação (4)		Equação (10)	
	Média	Desvio Padrão	Média	Desvio Padrão
Fluxo 1	370,1	193,6	331,7	5,4
Fluxo 2	744,0	389,4	664,4	7
Fluxo 3	1117,2	584,8	997,1	8,5
Fluxo 4	1489,9	779,5	1329,8	10,7
Fluxo 5	1863,6	975,4	1662,5	12,7
Fluxo 6	370,0	193,3	332	5,1
Fluxo 7	742,5	387,7	664,8	6,9
Fluxo 8	1115,7	583,0	997,7	8,8
Fluxo 9	1491,4	781,0	1330,7	10,8
Fluxo 10	1867,4	978,9	1663,7	12,6

As Figura 6 e Figura 7 mostram a situação do *buffer* para os dois casos em questão. A mesma conclusão sobre o comportamento das taxas de transmissão podem ser tiradas a cerca da ocupação do *buffer*. Isto é, enquanto altas oscilações são observadas no método proposto em [1], nossa estrutura de controle oferece rápida convergência para o valor de ocupação ideal.

B. Experiência 2: Buffer de Tamanho Limitado

O cenário para a segunda fase de simulações é idêntico ao da primeira, exceto pelo fato de limitarmos o tamanho dos *buffers* dos roteadores de centro ao valor de objetivo total de ocupação, igual a 100 pacotes.

Procedemos as simulações de acordo com o que foi feito na primeira fase e os resultados apresentados nas Figura 8 e Figura 9 mostram a taxa de chegada de pacotes em cada receptor utilizando (4) e (10), respectivamente, como algoritmos de cálculos da taxa de transmissão. Uma observação interessante é que a oscilação de taxa de transmissão observada anteriormente (Fig. 4) no Roteador de Egresso não é mais observada no seu correspondente receptor (Fig. 8). De fato, este fenômeno é conseqüência da perda de pacotes devido ao enchimento do *buffer* no primeiro Roteador do núcleo.

O mecanismo de controle proposto neste trabalho excede em qualidade o proposto em [1] em termos de taxa de perda de pacotes. Para mostrar isso, as Figuras 10 e 11 comparam como a taxa de perda de pacotes varia no tempo para estes dois métodos.

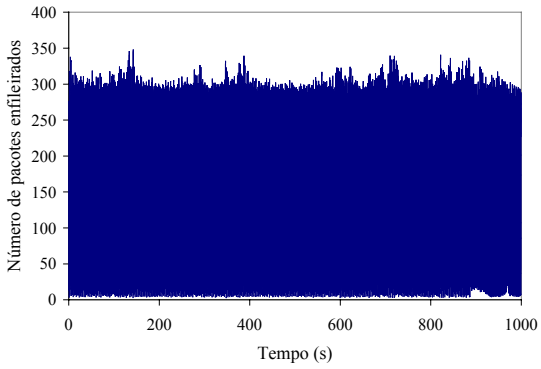

Fig. 6. Número de pacotes enfileirados vs tempo

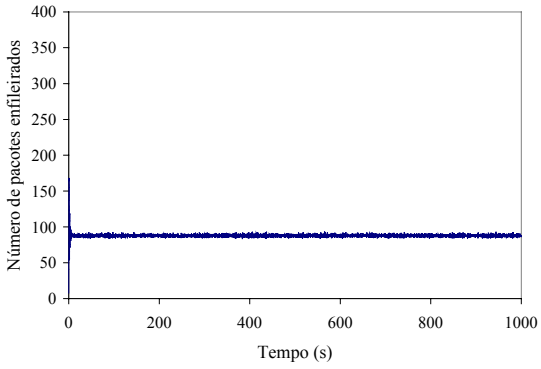

Fig. 7. Número de pacotes enfileirados vs tempo

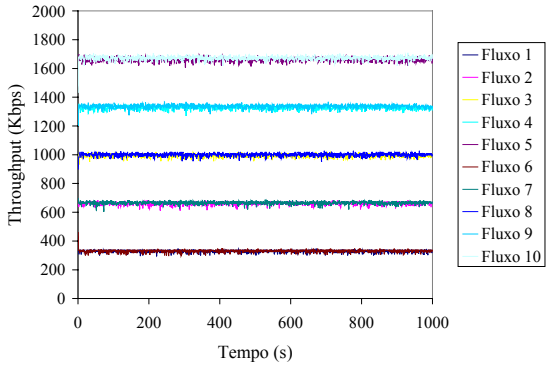

Fig. 8. Processamento dos fluxos vs tempo

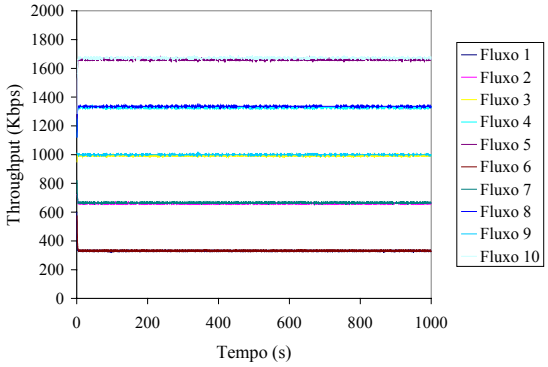

Fig. 9. Processamento dos fluxos vs tempo

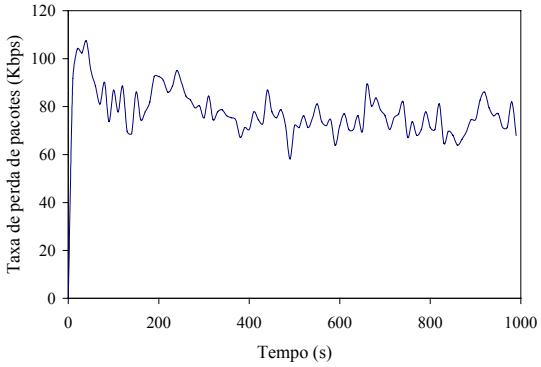

Fig. 10. Taxa de perda de pacotes vs tempo

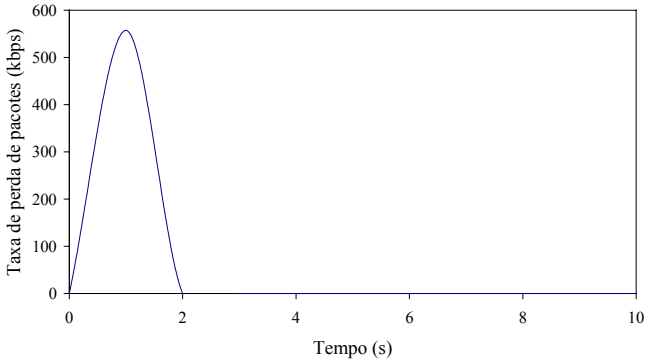

Fig. 11. Taxa de perda de pacotes vs tempo

Como pode ser observado, o método proposto em [1] não consegue evitar a perda constante de pacote durante todo o período de simulação. É válido ressaltar que não existe nenhum controle de prioridade de descarte no núcleo do domínio e a perda descontrolada de pacotes pode denegrir a qualidade de serviço prevista pelos SLAs. Por outro lado, com o nosso mecanismo de adaptação de taxa, conseguimos eliminar totalmente a perda de pacotes no núcleo do domínio, depois de um curto intervalo transiente como ilustrado pela Figura 11. Assim podemos ter controle sobre o descarte de pacotes nas bordas do domínio.

V. CONCLUSÕES

Nesse trabalho analisamos o desempenho da nova arquitetura DiffServ proposta em [1] com relação a estabilidade da taxa de transmissão estimada, bem como a perda de pacotes no interior do domínio. Propomos uma diferente estrutura de controle adaptativo de taxa de transmissão para essa arquitetura e através de simulações mostramos que esse novo algoritmo é capaz de obter melhor controle sobre as taxa de transmissão dos fluxos e dessa forma diminuir totalmente a perda de pacotes no interior da rede, depois de um curto intervalo transiente. Assim, conseguimos passar para as bordas do domínio todas as funcionalidades DiffServ, incluindo a função de controle de descarte de pacotes.

REFERÊNCIAS

[1] C. L. Lee, J. R. Chen, and Y. C. Chen, "*A scalable architecture for Differentiated Services"*, IEEE Conference on Distributed Computing Systems, 2002.

[2] R. Braden, D. Clark, and S. Shenker, *Integrated services in the Internet architecture: an overview*, RFC 1633, 1994.

[3] S. Blake, et al, *Architecture for differentiated services*, RFC 2475,1998.

[4] E. Rosen, A. Viswanathan, and R. Callon, *Multiprotocol label switching architecture*, Internet Draft, draft-ietf-mpls-arch-07.txt, 2000.

[5] D. Awduche et al., *Requirements for traffic engineering over MPLS*, Internet Draft, draft-ietf-mpls-traffic-eng.00.txt, 1998.

[6] E. Crawley et al., *A framework for QoS-based routing in the Internet*, RFC 2386, 1998.

[7] J. Crowcroft and P. Oechslin, "*Differentiated end-to-end Internet services using a weighted proportional fair sharing TCP"*, ACM Computer Communication Review, vol. 28, n.9, pp. 53-69, 1998.

[8] R. Sivakumar, et. al., "*Achieving per-flow weighted rate fairness in a core stateless network"*, IEEE Conference on Distributed Computing Systems, pp. 188-196, 2000.

[9] L. Stoica, S. Shenker and H. Zhang, *"Core-stateless fair queueing: Achieving approximately fair bandwidth allocations in high speed networks"*, ACM SIGCOMM, pp.118-130, 1998.

[10] L. Kleinrock, *Queueing Systems* Volume I, Theory, Wiley-Interscience, 1975.

[11] OPNET Technologies, Inc - OPNET Modeler (version 7.0), www.opnet.com.