

Mecanismos de Controle de Acesso em Redes de Alta Velocidade

Kelvin Lopes Dias *José A. Suruagy Monteiro*

Departamento de Informática
Universidade Federal de Pernambuco
50732-970 Recife, Pernambuco
E-mail: {kld,suruagy}@di.ufpe.br

Danilo Florissi

DCC Lab, Columbia University
450 Computer Science Bldg, NYC, NY
10027 USA
E-mail: df@cs.columbia.edu

Resumo

Este artigo analisa um novo mecanismo para o *controle de acesso* de tráfego em redes de alta velocidade, denominado RDMA (*Route Division Multiple Access*). O objetivo do estudo é avaliar RDMA quanto ao suporte oferecido aos requisitos de qualidade de serviço (QoS) das aplicações. A filosofia RDMA baseia-se na utilização de intervalos de tempo (*green-bands*) disponibilizados periodicamente. RDMA é comparado com dois outros mecanismos amplamente difundidos: *Virtual Clock* e *Leaky Bucket*. Em nosso estudo, RDMA apresentou-se satisfatório como mecanismo de controle de acesso para aplicações com requisitos estritos de QoS, representados pelo atraso e variação do atraso (*jitter*). O efeito moldador do tráfego, comum a estes tipos de mecanismos, não ocasiona atrasos e *jitter* excessivos. Além disso, RDMA pode controlar interferências entre fluxos que compartilhem enlaces comuns da rede. O estudo é baseado em simulações.

Abstract

In this article, we present an analysis of a new traffic access control mechanism for high speed networks, called RDMA (*Route Division Multiple Access*). Our objective is to evaluate the support RDMA offers to quality of service (QoS) requirements of applications. RDMA philosophy is based on the use of time intervals (green-bands) available periodically. RDMA is compared to two other well-known mechanisms: *Virtual Clock* and *Leaky Bucket*. In our studies, RDMA showed itself to be satisfactory as access control mechanism for applications with strict QoS requirements, represented by delay and delay variation (*jitter*). The traffic shapping effect, common to these mechanisms, does not cause excessive delays and jitter. Besides that, RDMA can control interferences among flows that share common links.

1 Introdução

As futuras redes de alta velocidade conduzirão uma gama de aplicações, variando desde àquelas pertencentes à comunicação de dados tradicional até aplicações que envolvem transmissão de vídeo de alta qualidade, como: telemedicina, vídeo sob demanda e realidade virtual. Fundamental para o sucesso desta integração de serviços, é o conceito de *controle e gerenciamento de tráfego*, que compreende um conjunto de *funções* cujo objetivo é assegurar que os usuários obtenham a qualidade de serviço (QoS — *Quality of Service*) almejada (por exemplo, atraso máximo fim-a-fim), sobretudo, em períodos de sobrecarga da rede [7].

De uma maneira geral, a QoS para uma nova requisição de conexão é especificada durante o procedimento de configuração desta conexão. A rede tem a oportunidade de estimar os recursos requisitados (*buffers*, largura de banda, processadores) e rejeitar o pedido caso os recursos sejam insuficientes. Para que a rede faça uma estimativa acurada, o usuário deve declarar as características de seu tráfego (por exemplo, as taxas média e máxima de transmissão) no pedido da conexão e, caso a conexão seja aceita, a rede tem a obrigação de garantir a QoS especificada pelo usuário. Este acordo forma a base para o *contrato de tráfego* entre a rede e o usuário. Uma vez que a conexão é estabelecida, a rede deve monitorar a fonte para assegurar que o acordo estabelecido durante a fase de contrato de tráfego seja respeitado.

Mecanismos de *controle de acesso* desempenham a função de monitoração do tráfego admitido pela rede, com o intuito de evitar que fontes mal comportadas (intencionalmente ou não) transmitam além do que foi especificado na fase de contrato de tráfego [14], o que poderia acarretar congestionamento. Os mecanismos para controle de acesso estão presentes na literatura de duas formas: desempenhando o papel de *policimento* ou de *moldagem* do tráfego. Ainda há a possibilidade de uma funcionalidade híbrida, onde os dois papéis podem ser utilizados.

No policiamento, quando um pacote do fluxo viola o contrato feito no estabelecimento da conexão, um policiador pode descartá-lo diretamente ou marcá-lo como sendo um pacote de baixa prioridade, permitindo que este seja descartado no caminho, se a rede estiver congestionada. Na moldagem, ao invés de descartar ou marcar o pacote fora do contrato, o tráfego seria moldado, atrasando os pacotes até que estes obedeçam novamente o contrato. O descarte pode ocorrer quando da ausência de *buffer* livre.

O controle de acesso pode ser tanto implementado na fonte do tráfego [11], como em todos os elementos da rede [12]. Apesar do papel fundamental que o controle de acesso desempenha na prevenção do congestionamento, pode ocasionar conseqüências danosas para a QoS das aplicações, seja pelo descarte de pacotes que ainda estão dentro do contrato mas que violaram os parâmetros utilizados pelos mecanismos para o policiamento, ou por atrasos excessivos sofridos por estes fluxos, no caso da moldagem.

Neste artigo, analisamos um novo mecanismo para o controle de acesso em redes de alta velocidade denominado RDMA [5, 6], do ponto de vista de sua eficácia como moldador do tráfego. O objetivo do artigo é propor uma análise de RDMA quanto ao suporte oferecido às aplicações. RDMA é analisado comparativamente com dois outros mecanismos: *Leaky Bucket* [15], utilizado na arquitetura ATM (*Asynchronous Transfer Mode*)[9], e *Virtual Clock* [17]. Este artigo está organizado da seguinte forma: na seção 2, apresentamos a filosofia de funcionamento do RDMA, juntamente com uma descrição do *Leaky Bucket*

e *Virtual Clock*. Em seguida, na seção 3, os resultados dos experimentos realizados são apresentados e discutidos e, finalmente, na seção 4 apresentamos as conclusões e trabalhos futuros.

2 Mecanismos para o Controle de Acesso

2.1 *Route Division Multiple Access* - RDMA

O RDMA é o mecanismo para controle de acesso adotado pelas redes *Isochronets* [5, 6], uma nova arquitetura para redes de alta velocidade. *Isochronets* dividem a largura de banda da rede entre árvores de roteamento (rotas de todos os nós para um determinado destino). Cada árvore configurada dura por uma banda de tempo (*green-band*) após a qual a próxima banda e árvore correspondente a substitui. Uma seqüência de bandas construtoras de árvores que cobrem cada destino uma e apenas uma vez formam um *ciclo*.

A Figura 1 apresenta uma árvore de roteamento, representada por todas as setas orientadas, para uma dada topologia, mostrando o movimento de um quadro, que é transmitido por uma rota da árvore. Esta rota é representada pelas setas tracejadas. O gráfico acima da topologia apresenta o movimento de um quadro transmitido pela origem e representado pela linha dentro da área hachurada (banda de tempo) no gráfico. Em linhas gerais, um quadro transmitido por uma fonte se propagará pela árvore de roteamento até o destino, movendo-se ininterruptamente caso não haja quadros de outro tráfego competindo pela mesma árvore.

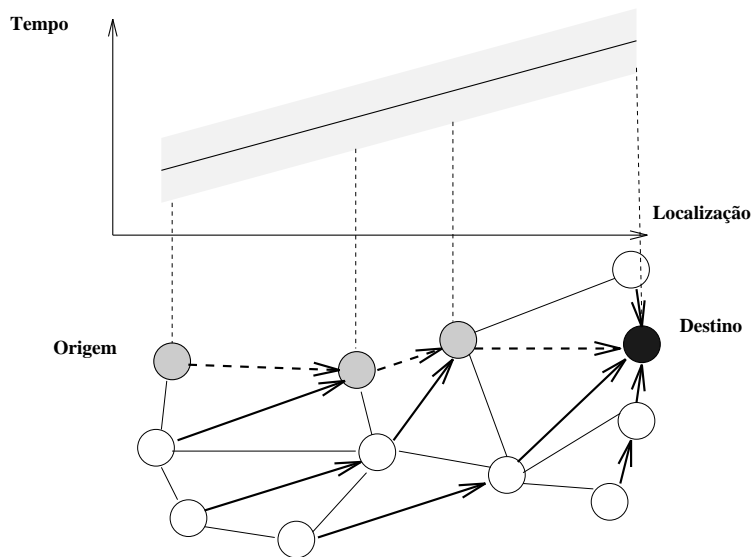


Figura 1: *Isochronets*.

Os nós de comutação intermediários mantêm as bandas de tempo por meio de temporizadores sincronizados para refletir a latência ao longo dos enlaces da árvore. O roteamento ao longo da banda é obtido através da configuração dos recursos dos comutadores

para escalonar quadros que chegam dos enlaces incidentes da árvore para os respectivos enlaces da saída desta árvore.

RDMA implementa a filosofia de controle da rede em *Isochronets*. As Figuras 2 e 3 exemplificam como o controle de acesso é exercido pelo RDMA. O relógio com ciclo de 12 unidades da Figura 2, apresenta o exemplo de alocação de uma banda de contenção (permite o compartilhamento dos recursos por múltiplas fontes) para a árvore de roteamento apresentada na Figura 3, com apenas duas fontes (origens) utilizando a árvore. A Figura 3 mostra a alocação da banda no nó intermediário *C*.

C particiona sua banda entre seus descendentes *A* e *B*. Perceba que os tamanhos alocados para *A* e *B* são diferentes e apenas se sobrepõem parcialmente e quando isto ocorre os quadros de *A* e *B* competem pelo enlace de saída de *C*.

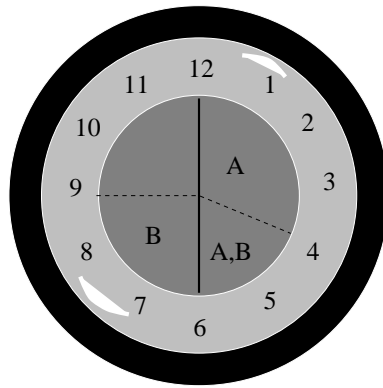


Figura 2: Exemplo de Alocação em RDMA.

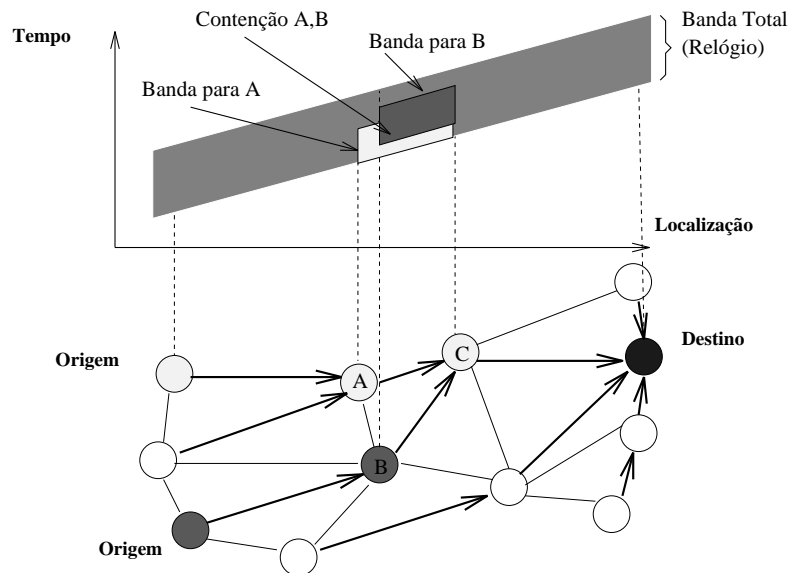


Figura 3: RDMA: Banda de Contenção.

Quando ocorre contenção, o modo de resolução de colisão utilizado é designado por meio de sinais: -, + e ++. Em RDMA-, apenas um dos quadros competindo pelo enlace

prossegue, enquanto os demais são descartados. Em RDMA+, um dos quadros prossegue enquanto os outros são armazenados, mas apenas até o término da duração da banda. RDMA++ opera de forma similar ao RDMA+, contudo, armazena os quadros além do término da banda, reescalando-os durante a próxima ativação da banda.

Existe, também, a possibilidade de controlar o acesso através de bandas de prioridade. Um exemplo pode ser representado pelo último quarto do relógio da Figura 2, isto é, de 9 a 12. A Figura 4 mostra que a origem que utiliza esta banda não sofre contenção no caminho.

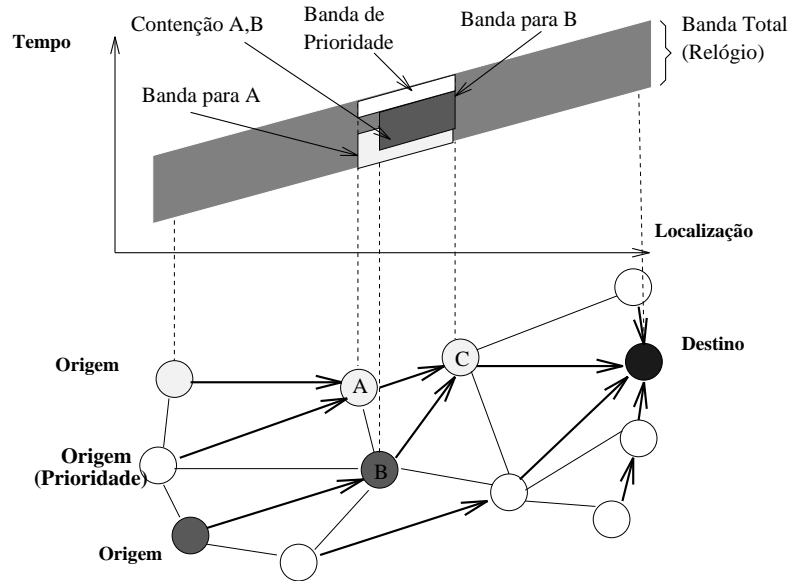


Figura 4: RDMA: Banda de Prioridade.

Para o dimensionamento do RDMA é necessário apenas que se defina a largura da banda para o fluxo de acordo com a taxa média de transmissão exigida pela fonte. A largura de banda deve representar uma porcentagem do ciclo básico RDMA, definido como sendo de $125 \mu s$ [5]. Por exemplo, se um enlace tem 150 Mbps e uma fonte requer 75 Mbps em média, a banda alocada para esta fonte corresponderá a $75/150 = 50\%$ do ciclo RDMA de $125 \mu s$, ou seja, $62,5 \mu s$.

2.2 Leaky Bucket

Proposto inicialmente em [15], é o mecanismo utilizado para o controle de acesso em redes ATM. Consiste em incrementar um contador de créditos, C , de uma unidade a cada D segundos, até que um valor máximo B seja alcançado. O contador é decrementado de uma unidade a cada célula transmitida e a taxa de transmissão é limitada a um valor máximo de $\rho = \frac{1}{D}$ células/s. Com isso, a rajada máxima permitida por este esquema equivale ao valor da variável B mais as células transmitidas no período de enchimento. Quando o contador é zerado, a transmissão é interrompida e duas opções podem ser tomadas quando da chegada de novas células ao *buffer*: o descarte destas células diretamente ou apenas a marcação das células com a indicação de que caso ocorra congestionamento nos

nós seguintes da rede, estas podem ser descartadas. Cada fluxo deve ter seu próprio *Leaky Bucket*, como ilustrado na Figura 5. O *Leaky Bucket* destaca-se, também, pela simplicidade de sua implementação.

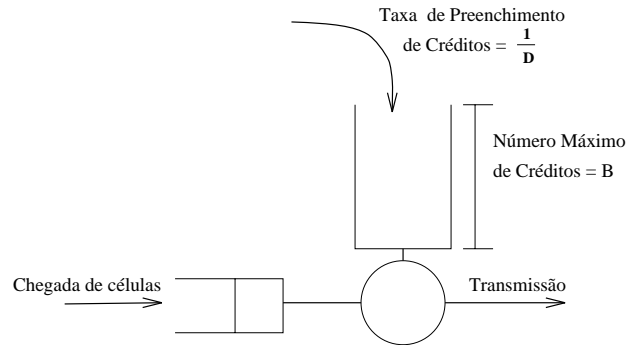


Figura 5: *Leaky Bucket*.

2.3 Virtual Clock

Virtual Clock é um algoritmo para controle de tráfego em redes de alta velocidade que foi projetado como parte de uma nova arquitetura de rede comutada por pacotes, a *Rede de Fluxo (Flow Network)* [17]. Para cada fonte, as necessidades de recursos para um determinado fluxo, são expressas em termos de dois parâmetros: uma taxa média (AR) e um intervalo médio (AI) entre verificações da conformidade do tráfego. O parâmetro AR especifica as necessidades de vazão média da fonte e AI impõe um limite na explosividade da fonte, ditando como os pacotes podem estar espalhados ao longo do tempo para a transmissão. A seleção de AI é uma questão em aberto, podendo assumir a seguinte faixa de valores: $\frac{1}{AR_i} \leq AI \leq \text{Duração Total do Fluxo}$.

Cada fluxo de dados tem uma variável *auxVC* que é incrementada a cada pacote que requisita acesso à rede com o tempo entre chegadas de requisições de transmissões, $\frac{1}{AR_i}$, assumindo-se pacotes com o tamanho constante. A transmissão de pacotes pelo próximo enlace é governada pela ordem crescente das marcas, isto é, o valor do *auxVC* dos fluxos. O descarte de pacotes, quando há escassez de espaço no *buffer*, ocorre para pacotes com o maior valor da variável *auxVC*. Uma outra variável, *virtual clock*, especifica se o fluxo está obedecendo ao contrato de tráfego. Se um fluxo envia pacotes de acordo com a taxa média especificada, a leitura dessa variável deve estar na vizinhança do tempo de um relógio real. Se esta leitura extrapolar um determinado limite, uma ação deve ser tomada. A intenção do mecanismo é emular um sistema de multiplexação por divisão no tempo, TDM – *Time Division Multiplexing*, contudo, obtendo vantagens da multiplexação estatística, o que não ocorre em sistemas TDM.

3 Atraso e Jitter no Acesso à Rede

O modelo utilizado em nossas primeiras simulações é apresentado na Figura 6. Nele, utilizamos uma fonte individual que gera células ATM. Consideramos que os dados da

aplicação são empacotados em 48 bytes, correspondentes ao *payload* (campo da célula destinado aos dados do usuário) padrão de uma célula ATM. O acesso à rede, após o controle, é viabilizado por um enlace de 150Mbps.

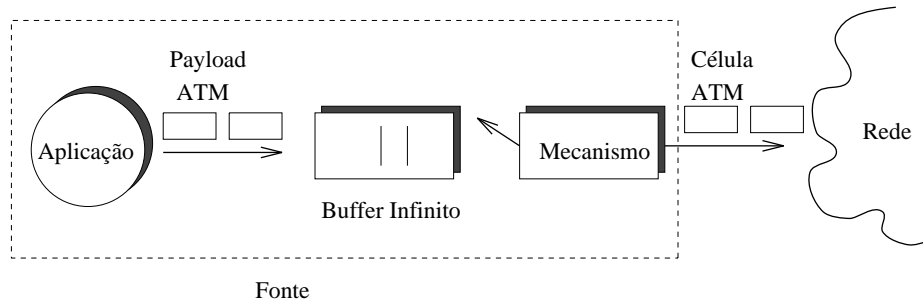


Figura 6: Modelo de Simulação para Fontes Individuais.

A coleta das métricas em estudo, como *jitter* e atraso, é efetuada na saída da fonte, isto é, após as células terem passado pelos mecanismos e terem sido transmitidas. Os seguintes parâmetros foram considerados: atraso médio de acesso, definido como a diferença entre o momento no qual uma célula teve acesso à rede (foi transmitida pela fonte) e o momento no qual esta célula foi gerada pela aplicação, o *jitter*, definido como a diferença entre os atrasos médios (como definido anteriormente) sofridos por duas células consecutivas na saída do mecanismo e, finalmente, a ocupação média no *buffer* infinito do modelo da Figura 6. Os modelos de simulação foram implementados com a utilização da ferramenta BONEs (*Block Oriented Network Simulator*) Designer [1].

3.1 Análise para Fonte de Voz

Utilizamos o modelo ON-OFF de acordo com as conclusões em [3] para modelar a fonte de voz. A seguir, apresentamos as principais características desta fonte, especificadas em termos de células: taxa de pico = 167 células/s, taxa média = 60 células/s, nível de explosividade (*burstiness*)* = 2,78 e comprimento médio da rajada = 60 células.

No caso do RDMA precisamos especificar a largura de banda necessária para acomodar a média do tráfego prometida pela fonte, que é de 60 células/s. Para um enlace com capacidade de 150Mbps, a banda ocupará $1,5 \times 10^{-2}\%$ do ciclo básico RDMA, ou seja, $1,9 \times 10^{-2} \mu s$ dos $125 \mu s$ correspondentes ao ciclo. É importante ressaltar que neste trabalho não consideramos uma característica nova de RDMA que é a possibilidade abrir a banda de transmissão sincronamente com a geração de dados pela aplicação [5]. Assim, os dados podem chegar durante a banda ociosa de RDMA.

Para o *Virtual Clock* o valor médio da taxa de célula prometida pela fonte, 60 células/s, será utilizada como seu parâmetro de taxa (AR). O tempo entre verificações, AI, é configurado com o tempo médio no qual a fonte transmite (estado ON): 360ms ou β^{-1} [3].

No *Leaky Bucket* o número máximo de créditos que o mecanismo pode armazenar, corresponde ao comprimento médio de rajada especificado pela fonte de voz, 60 células. A taxa de preenchimento de créditos corresponde à taxa média de transmissão da fonte

*taxa de pico/taxa média

de voz, 60 células/s, multiplicada por um fator C (utilizado para evitar atrasos excessivos quando a fonte transmite acima da taxa média).

A Tabela 1 apresenta os resultados do estudo para a fonte de voz. Os resultados, deste e demais experimentos do artigo, foram obtidos utilizando-se a técnica de médias em lote (*Batch Means*). Os valores médios apresentados, para o estudo da fonte de voz, foram obtidos com precisão dentro de 6% para um nível de confiança de 95%.

<i>Mecanismo</i>	<i>Atraso Médio</i>	<i>Jitter Máximo</i>	<i>Jitter Médio</i>	<i>Ocupação Média</i>
RDMA	0,0001 s	0,000116 s	0,000009 s	0,5 célula
<i>Leaky Bucket</i> C=1,2	1,0058 s	0,0138 s	0,00523 s	65,57 células
<i>Leaky Bucket</i> C=1,4	0,2930 s	0,0119 s	0,00242 s	22,29 células
<i>Virtual Clock</i>	0,2814 s	0,6501 s	0,0335 s	27,22 células
<i>Virtual Clock</i> C = 1,2	0,1930 s	0,6501 s	0,0144 s	19,03 células

Tabela 1: Resultados para a Fonte de Voz.

Dentre os atrasos, o menor foi introduzido pelo RDMA, o maior para *Leaky Bucket*, mesmo com a utilização da taxa de preenchimento de créditos maior que a taxa média especificada pela fonte. O atraso sofrido para o fluxo de voz, utilizando-se RDMA, é introduzido exclusivamente pela possibilidade de células requisitarem transmissão quando a banda estiver fechada/não ativa. Assim, o tempo máximo de permanência que células deste fluxo podem experimentar é ditado pela duração da banda ociosa mais o tempo de transmissão da célula.

O atraso introduzido pelo *Leaky Bucket* é devido à falta de créditos, quando da chegada de uma célula para transmissão. Quando isto ocorre, o tráfego pode ficar limitado a uma taxa constante de transmissão definida pela taxa de preenchimento de créditos. Mesmo com valores mais altos para a taxa de preenchimento de créditos, o *Leaky Bucket* ainda introduz atrasos médios maiores do que os introduzidos pelo RDMA e *Virtual Clock*. Uma constatação similar do atraso excessivo gerado pelo *Leaky Bucket*, ainda na estação de trabalho, também foi apresentada em [10].

O atraso médio do *Virtual Clock* depende da escolha do tempo entre verificações do fluxo e da própria taxa média especificada para o fluxo. Em conjunto, eles ditam o número de células após o qual o fluxo deve ser verificado. No *Virtual Clock*, assim como no *Leaky Bucket*, pode ocorrer um atraso excessivo de células, mesmo que estas estejam dentro da taxa média de transmissão de longo prazo especificada pela fonte. Este atraso pode ser reduzido de duas formas: se o tempo entre verificações for relativamente alto, ou se utilizarmos um artifício semelhante ao utilizado pelo *Leaky Bucket*, isto é, utilizarmos um fator maior que um que multiplica a taxa média utilizada como parâmetro do mecanismo. Ambas as saídas podem causar efeitos colaterais como, por exemplo, a injeção de rajadas excessivas na rede (isto também é verdade no caso do *Leaky Bucket*), apesar de poderem reduzir o atraso sofrido já na própria fonte.

Uma outra forma de verificar o efeito negativo causado pelos mecanismos para a fonte de voz, é através da observação do número médio de células que ficam no *buffer* de controle, aguardando permissão para serem transmitidas. Esta é outra forma de analisar o atraso introduzido pelos mecanismos. A grande diferença entre a ocupação média de RDMA e

dos demais mecanismos é decorrente da baixa vazão exigida pela fonte: o tempo máximo de espera pela banda é de aproximadamente $125\mu s$ e o tempo entre chegadas de células é de $\frac{1}{60 \text{ células/s}} = 16\text{ms}$.

Dentre os mecanismos, apenas o RDMA pode alcançar os objetivos de atraso fim-a-fim especificados para fonte de voz, já o *Virtual Clock* e o *Leaky Bucket* necessitam da utilização de valores maiores do que aqueles que servem como dimensionamento padrão. Isto é, os valores médios especificados pela fonte, pois já na periferia podem ter atrasos significativamente altos. Valores desejáveis para o atraso fim-a-fim de fluxos de voz para a maioria dos usuários estão entre 0 e 150ms [9].

Assim como no caso do cálculo do atraso, o *jitter* sofrido pelo fluxo de voz, para RDMA, depende da largura da banda ociosa, tanto o *jitter* máximo quanto o médio são menores em RDMA do que nos demais mecanismos. A maior diferença entre duas células consecutivas, ocorre quando a predecessora, utilizada no cálculo do *jitter*, chega no momento em que a banda ativa se fecha, dessa forma, terá que esperar até a próxima banda ativa para ser transmitida.

3.2 Análise para Fonte de Vídeo

Em nossos estudos, utilizamos o modelo apresentado em [8], descrito por um processo auto-regressivo de primeira ordem, gerado pela relação: $\lambda(n) = a\lambda(n-1) + b\omega(\eta)$, onde $\lambda(n)$ representa a taxa de bit de uma única fonte durante o n-ésimo quadro, em bits/*pixel*. $\omega(\eta)$ é a seqüência de variáveis aleatórias Gaussianas independentes com variância e média iguais a 1 e η , respectivamente, com a e b sendo constantes. Casando este modelo com os dados experimentais obtidos de uma seqüência representativa de aplicações de videofonia, foram obtidos os seguintes valores para a, b e η [8]: $a \cong 0,8781$; $b \cong 0,1108$; $\eta \cong 0,572$. A fonte de vídeo apresenta uma taxa média de 3,9Mbps (taxa média, λ , igual a 0,52 bit/*pixel*), taxa de pico igual a 10,68Mbps (1,41 bit/*pixel*) e 250.000 pixels por quadro [8].

Para o RDMA necessitamos especificar apenas a largura de banda necessária para acomodar a média do tráfego prometida pela fonte, que é de $\frac{3,9Mbps}{8bits \times 48bytes} = 10156$ células/s, utilizando-se um *payload* de 48 bytes para a célula ATM. Para um enlace com capacidade de 150Mbps, a banda ocupará 2,6% do ciclo básico RDMA, ou seja, $3,25\mu s$ dos $125\mu s$ correspondentes ao ciclo.

No *Virtual Clock* o valor médio da taxa de célula prometida pela fonte, 10156 células/s, será utilizada como seu parâmetro de taxa, AR. O tempo entre verificações foi escolhido para corresponder à periodicidade dos quadros de vídeo, isto é, 33ms.

Com o *Leaky Bucket*, para o número máximo de créditos que o mecanismo poderia armazenar, utilizamos duas opções: o número médio e máximo de células por quadro. Esses valores correspondem a: $\frac{0,52 \times 250000}{8 \times 48} = 339$ e $\frac{1,41 \times 250000}{8 \times 48} = 918$ células, respectivamente. A taxa de preenchimento corresponde à taxa média de transmissão da fonte de vídeo, 10156 células/s.

Os resultados para vídeo são apresentados na Tabela 2 e foram obtidos com precisão dentro de 8% em relação aos correspondentes valores médios, para um nível de confiança de 95%. *Leaky Bucket* continua apresentando o maior atraso mesmo para taxas de preenchimento de créditos maiores do que a taxa média de transmissão especificada pela fonte. *Virtual Clock* apresenta um atraso menor do que RDMA, apenas quando utiliza um fator multiplicativo para a taxa média do mecanismo.

<i>Mecanismo</i>	<i>Atraso Médio</i>	<i>Jitter Máximo</i>	<i>Jitter Médio</i>	<i>Ocupação Média</i>
RDMA	0,00787 s	0,0001217 s	0,000041 s	125,65 células
<i>Leaky Bucket</i> C=1,15	0,16115 s	0,000086 s	0,000010 s	1.282 células
<i>Leaky Bucket</i> C=1,15 Créditos=918	0,13198 s	0,000086 s	0,000008 s	947,34 células
<i>Leaky Bucket</i> C=1,4	0,0166 s	0,00007 s	0,000002 s	210,79 células
<i>Virtual Clock</i>	0,00904 s	0,03300 s	0,000102 s	88,22 células
<i>Virtual Clock</i> C=1,2	0,00456 s	0,03300 s	0,000027 s	45,19 células

Tabela 2: Resultados para a Fonte de Vídeo.

O *jitter* máximo introduzido pelos mecanismos está relacionado, novamente, a valores de alguns parâmetros utilizados para configurá-los. Por exemplo, o *jitter* máximo do *Leaky Bucket* depende do tempo entre chegadas de créditos. No caso do *Virtual Clock*, corresponde ao atraso máximo que as células da fonte de vídeo podem sofrer. Mesmo com a utilização do fator C=1,2 para o *Virtual Clock*, não evitou-se que o *jitter* máximo sofrido fosse igual ao do *Virtual Clock* com o dimensionamento padrão. Para RDMA, da mesma forma que para a fonte de voz, pode-se alcançar um *jitter* máximo correspondente ao tempo de duração da banda ociosa.

Perceba que para o caso de vídeo, o *jitter* máximo sofrido por *Leaky Bucket* é menor que o introduzido pelo RDMA. Isto é verdade, porque o tempo entre chegadas de créditos é, consideravelmente, menor que o tempo de espera pela banda ativa, que no pior caso corresponde a 121,7 μ s. Os valores para o *jitter médio* seguem o mesmo comportamento do *jitter* máximo. A ocupação média em todos os mecanismos foi relativamente alta. No caso do RDMA ocorre devido ao número médio de células que aguardam a banda ativa, por chegarem durante a banda ociosa do ciclo RDMA.

De uma maneira geral, RDMA provê resultados satisfatórios para os fluxos de voz e vídeo modelados. Tendo como ponto favorável a filosofia de controle periódico, adequada para tráfego isócrono.

3.3 Análise do Atraso com Tráfego de Interferência

O objetivo deste estudo é verificar o atraso sofrido por um fluxo bem comportado (Tráfego Constante), quando este compete com um outro fluxo (Tráfego de Fundo Variável) pelo mesmo enlace de saída do nó periférico. Utilizamos um MMPP (*Markov Modulated Poisson Process*) de dois estados [2] para modelar tanto o tráfego constante quanto o tráfego de fundo variável. Os mecanismos têm seus parâmetros configurados para suportar uma carga de 50% referente à capacidade total do enlace de saída, definida para este estudo como sendo de 75Mbps, inicialmente. O tráfego de fundo, varia desde a região com a carga abaixo da utilizada para dimensionar os mecanismos até a carga acima desta. A Figura 7 apresenta o modelo utilizado neste estudo. Neste modelo, as filas de transmissão e a que armazena o tráfego constante, são consideradas infinitas para que não ocorram perdas de células em função da falta de espaço de armazenamento.

Além do estudo do efeito causado por um tráfego de fundo/interferência, verificamos

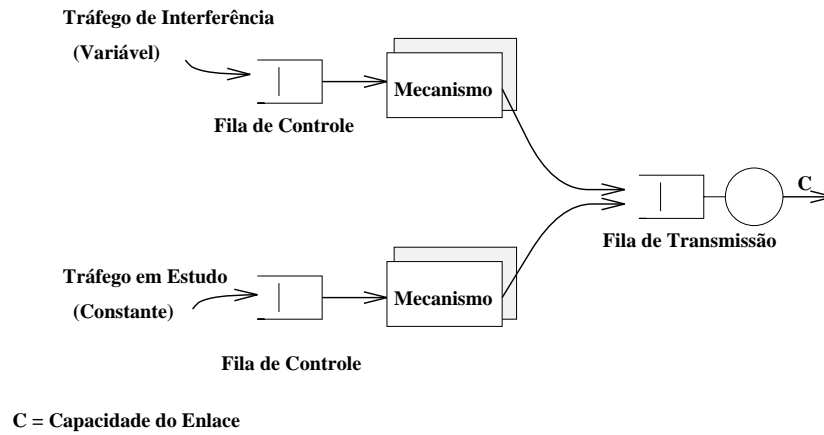


Figura 7: Modelo com Tráfego de Interferência.

a influência da variação da capacidade do enlace no atraso introduzido pelo RDMA. Com este fim, dobramos a capacidade para 150Mbps. O dimensionamento feito para os demais mecanismos não precisa variar com a capacidade do enlace, pois independem dela e precisam apenas de informações do tráfego, que permaneceu o mesmo. Entretanto, o dimensionamento RDMA precisa da informação do enlace. Para esta nova capacidade de 150Mbps, RDMA é dimensionado para um quarto do ciclo de $125\mu s$, ao invés da metade do ciclo, como feito para o experimento com o enlace de 75Mbps. Os resultados das simulações, apresentados na Figura 8, foram obtidos considerando um intervalo com um nível de confiança de 95%. O dimensionamento dos demais mecanismos segue a mesma filosofia do dimensionamento para fonte individual de voz. O valor utilizado pelo fator C do *Leaky Bucket* foi 1,4. Para o *Virtual Clock* o período máximo durante o qual o fluxo pode ser penalizado corresponde a $0,1ms$.

Nas simulações feitas para o enlace de 150Mbps, RDMA apresenta o menor atraso, juntamente com *Virtual Clock*. O *Leaky Bucket* apresenta o maior atraso e, para esta capacidade de enlace, o tráfego de fundo não ocasiona um maior atraso para o tráfego constante à medida que a carga daquele aumenta.

Quando a capacidade do enlace torna-se 75Mbps, temos resultados diferentes dos obtidos anteriormente. A diferença mais significativa foi observada no comportamento do atraso para RDMA. Ocorreu um aumento considerável se comparado com o atraso sofrido pelos demais mecanismos para esta nova capacidade do enlace, embora a largura de banda RDMA para um enlace de 75Mbps corresponda a 50% do ciclo de $125\mu s$, e a largura de banda para o enlace de 150Mbps, corresponda à apenas 25% do mesmo ciclo. Podemos atribuir este atraso significativo, à redução da capacidade do enlace. Nesta configuração, o RDMA terá um maior acúmulo de células aguardando pela banda ativa, uma vez que, não conseguirá dar vazão para todas as células que aguardam pela transmissão. É importante frisar que estes atrasos podem ser naturais para a periferia de uma rede que utiliza RDMA. Contudo, à medida que o fluxo trafega pela rede, não sofrerá tanto com o atraso, uma vez que, as bandas nos respectivos nós da árvore RDMA, estarão sincronizadas [4]. Outra observação importante é que no caso do *Leaky Bucket*, o servidor continua funcionando mesmo quando o tráfego está ocioso. Já RDMA, desperdiçará a sua banda ociosa.

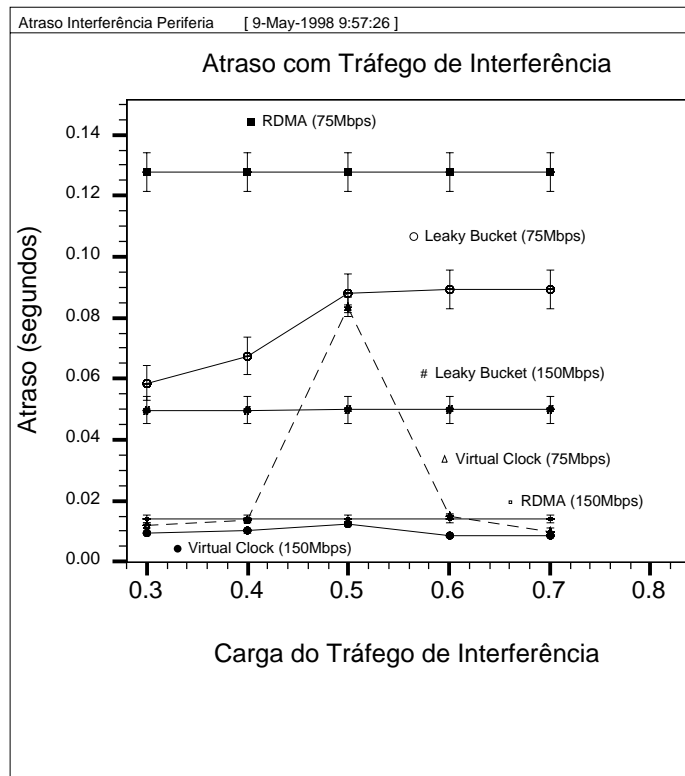


Figura 8: Atraso com Tráfego de Interferência.

O *Leaky Bucket* também foi afetado com a redução da capacidade do enlace para 75Mbps. Isto pode ser explicado pelo acúmulo de células na fila de transmissão. À medida que o tráfego de fundo aumenta sua taxa de transmissão, mais células chegarão e serão transmitidas antes das células pertencentes ao tráfego constante, uma vez que a fila de transmissão implementa simplesmente a disciplina FCFS (*First Come, First Served*).

Para nosso estudo, as perdas só ocorrem para a fila de controle do *Leaky Bucket* que serve ao tráfego de fundo, quando sua taxa de transmissão ultrapassa a taxa correspondente à taxa utilizada para dimensioná-lo. Tanto a fila de transmissão, quanto a de controle do tráfego constante, são suficientemente altas para evitar perdas. Entretanto, o atraso sofrido pelo fluxo constante, não sobe indefinidamente pelo fato do *Leaky Bucket*, para o tráfego variável, inibir a entrada de tráfego além do permitido pelo número máximo de créditos e pela taxa de preenchimento, o que permite este efeito suavizador na curva (Figura 8). A partir deste ponto, o tráfego do fluxo variável chegará à fila de transmissão, aproximadamente, como um tráfego CBR (*Constant Bit Rate*).

O *Virtual Clock* para o enlace de 75Mbps, apresenta um crescimento na metade da curva. Neste ponto, tanto o tráfego de fundo quanto o tráfego constante, estão dentro do contrato feito com a rede no momento da admissão e, da mesma forma que o ocorrido para o *Leaky Bucket*, a fila de transmissão para o *Virtual Clock* começa a não ser mais desprezível, o que ocasiona um maior atraso para o fluxo constante.

No caso do *Virtual Clock* é interessante perceber que, diferente do comportamento do *Leaky Bucket* para cargas altas, o atraso do fluxo constante sofre uma redução quando a taxa de transmissão do fluxo variável aumenta. Neste momento, além das penalidades

que o fluxo variável sofre, cada vez maiores com o aumento da taxa de transmissão, a disciplina de serviço [16] começa a surtir efeito na diminuição do atraso sofrido pelo fluxo constante. Apenas atrasando o fluxo de fundo não implica na redução do atraso do fluxo constante, uma vez que, quando habilitado para transmitir, o fluxo de fundo entraria na rede com rajadas cada vez maiores, aumentando assim o atraso do fluxo constante, caso a disciplina de serviço não fosse parte integrante do algoritmo do mecanismo. Dessa forma, a transmissão final pelo enlace é governada pela ordem crescente das marcas que cada pacote possui, onde os pacotes do fluxo com carga acima da prometida teriam valores de marcas maiores, perdendo dessa forma, a prioridade para acessar o enlace para os pacotes com valores de marcas menores, pertencentes ao fluxo constante.

Finalmente, a Figura 9 apresenta o atraso sofrido por um fluxo MMPP em função da variação de sua carga.

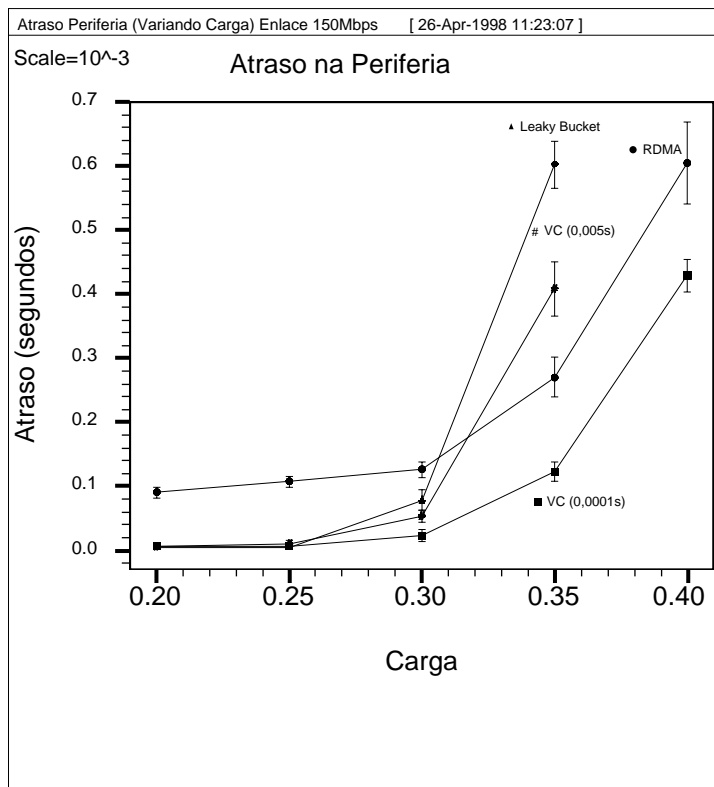


Figura 9: Atraso na Periferia.

As cargas oferecidas foram obtidas com relação a uma capacidade de enlace de 75Mbps e o enlace do simulador utilizou uma capacidade de 150Mbps [4]. Os mecanismos são configurados com os mesmos valores utilizados para dimensioná-los no estudo do efeito do tráfego de interferência descrito anteriormente, isto é, levam em consideração que a rede deve controlar um tráfego que requisita 50% de 75Mbps. Como feito no experimento anterior, o dimensionamento de RDMA utiliza apenas 25% de seu ciclo de 125 μ s, uma vez que, a capacidade do enlace simulado é de 150Mbps. Neste estudo variamos, também, o parâmetro que determina o tempo máximo de penalização para o fluxo do *Virtual Clock*, com valores de 0,1ms (VC(0,0001s)) e 5ms (VC(0,005s)).

O maior atraso sofrido pelo RDMA, para as cargas mais baixas, é explicado pela espera da banda ativa. Este atraso não ocorre para os demais mecanismos, uma vez que, tanto *Leaky Bucket* quanto *Virtual Clock* têm seus parâmetros configurados para suportar uma carga superior do tráfego, não sofrendo da falta de créditos ou de penalizações excessivas pelo fato do tráfego estar acima da média especificada, respectivamente. Entretanto, observe que à medida que a carga se aproxima daquela que serviu para dimensionar os mecanismos (50%), *Leaky Bucket* e *Virtual Clock* com valor máximo de penalização igual a 5ms, começam a sofrer um atraso maior do que o do RDMA. Este comportamento ocorre pelo fato de começar a ocorrer falta de créditos para o *Leaky Bucket* e aumento no tempo de espera na fila de penalização para o *Virtual Clock*. A curva do *Virtual Clock*, configurado com uma penalidade máxima para seu fluxo de 0,1ms, continua com um atraso sempre inferior ao de RDMA.

4 Conclusão e Trabalhos Futuros

Para o estudo do emprego do RDMA na fonte do tráfego, a provisão de QoS para as aplicações que envolvem áudio e vídeo, descritas em termos do atraso e *jitter* introduzidos pelo efeito controlador do mecanismo foi satisfatória. O RDMA necessita que seja especificada a taxa média de transmissão da fonte (um dos parâmetros especificados durante o contrato de tráfego no momento em que a rede decide se deve ou não admitir o novo fluxo). Para RDMA, a utilização deste único parâmetro de tráfego para o dimensionamento do mecanismo, não prejudicou a QoS percebida por estes tipos de aplicações, em termos de atraso e *jitter*.

Os demais mecanismos, *Leaky Bucket* e *Virtual Clock*, podem penalizar excessivamente pacotes do fluxo, mesmo para o caso onde o fluxo não esteja violando o contrato feito com a rede. Para evitar este problema é necessário que os parâmetros utilizados para dimensionar *Leaky Bucket* e *Virtual Clock* sejam superestimados. Este tipo de solução apesar de diminuir o atraso e o *jitter* introduzidos por tais mecanismos, tem o efeito colateral de deslocar o problema para elementos internos da rede, proporcionando maiores chances para a ocorrência de congestionamento [4].

Ao contrário de RDMA, os demais mecanismos necessitam de mais de um parâmetro para sua configuração, obtidos a partir do descritor de tráfego cedido pela fonte e, além disso, alguns parâmetros devem ser obtidos subjetivamente, como: os fatores usados para superdimensionar as taxas de preenchimento de créditos do *Leaky Bucket* e a taxa média de transmissão utilizada pelo *Virtual Clock* e o tempo entre verificações e de penalização que os pacotes de um fluxo devem sofrer. Como verificado nos resultados dos experimentos, tais mecanismos só obtiveram melhores resultados que os de RDMA quando seus parâmetros foram superdimensionados. Com relação aos aspectos de implementação, apesar de não ser tão simples quanto o *Leaky Bucket*, o RDMA pode ser implementado em *hardware* [5].

RDMA é um mecanismo que pode desempenhar várias das funções de gerenciamento de tráfego em redes de alta velocidade. Além do controle de acesso, o gerenciamento de fila (disciplina de serviço) está intrínseco à escolha da banda. Por exemplo, fluxos com alta prioridade podem ser escalonados dentro de bandas de prioridades. Como verificado nos experimentos envolvendo o efeito da interferência entre fluxos que competem pelo mesmo

enlace de saída na periferia da rede, o fluxo analisado não sofre qualquer interferência, com isso o atraso sofrido em RDMA pode depender exclusivamente do efeito moldador do mecanismo. O gargalo para o RDMA reside na periferia da rede, isto é, para que um determinado tráfego tenha acesso à rede, deve esperar por sua banda ativa. Este será um componente previsível do atraso que pode ser introduzido para o tráfego isócrono e, portanto, o atraso fim-a-fim sofrido por estes tipos de fluxo pode ser facilmente controlado.

Podemos concluir que o RDMA pode ser caracterizado como um mecanismo eficaz para o controle de acesso para aplicações de tempo real em redes de alta velocidade, como verificado nos experimentos envolvendo áudio e vídeo, bem como, RDMA pode evitar a interferência entre fluxos. Pesquisas futuras neste sentido, compreendem a análise de RDMA com modelos de *buffers* finitos e modelos de tráfego mais realistas, como os modelos auto-similares ou fractais [13], a verificação da probabilidade de perda de células/pacotes na utilização dos modos de resolução de colisão, bem como, novos estudos, agora, relacionados à eficácia do controle admissão de novas conexões (CAC – *Connection Admission Control*) intrínseco ao RDMA através da alocação de bandas de tempo, comparativamente com os métodos de CAC empregados no ATM.

Referências

- [1] Alta Group of Cadence Systems, Inc. *BONeS Designer Release 3.6. Modelling Reference Guide*, March 1996.
- [2] A. Baiocchi, N. Melazzi, A. Roveri, and R. Winkler. Modeling issues on an ATM Multiplexer within a Bursty Traffic Environment. *INFOCOM'91*, pages 83–91, 1991.
- [3] P. T. Brady. A Model for Generating ON-OFF Patterns in Two-Way Conversations. *Bell System Technology Journal*, 48:2445–2472, 1969.
- [4] K. L. Dias. Mecanismos de Controle de Acesso em Redes de Alta Velocidade: Um Estudo Comparativo de RDMA com *Leaky Bucket* e *Virtual Clock*. Master's thesis, Universidade Federal de Pernambuco, Junho 1998.
- [5] D. Florissi. *Isochronets: A High Speed Switching Network Architecture*. PhD thesis, Computer Science Department, Columbia University, 1995.
- [6] D. Florissi and Y. Yemini. On the Design and Implementation of the Isoswitch, a Multiprotocol High-speed Switch. *XIV Brazilian Symposium on Computer Networks*, pages 592–610, May 1996.
- [7] R. Jain. Congestion Control and Traffic Management in ATM Networks: Recent Advances and a Survey. *Computer Network and ISDN Systems*, 28:1723–1738, 1996.
- [8] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, and J. D. Robbins. Performance Models of Statistical Multiplexing in Packet Video Communications. *IEEE Trans. Comm.*, 36(7):834–844, July 1988.
- [9] R. O. Onvural. *Asynchronous Transfer Mode Networks*. Artech House, Second edition, 1995.
- [10] B. V. Patel and C. C. Bisdikian. End-Station Performance under Leaky Bucket Traffic Shapping. *IEEE Network*, 10(5):40–47, sep/oct 1996.

- [11] S. Radhakrishnan, S. V. Raghavan, and A. K. Agrawala. A Flexible Traffic Shaper for High Speed Networks: Design and Comparative Study with Leaky Bucket. *Comp. Net. and ISDN Syst.*, 28(4):453–469, Feb. 1996.
- [12] J. Rexford, F. Bonomi, A. Greenberg, and A. Wong. Scalable Architectures for Integrated Traffic Shaping and Link Scheduling in High-Speed ATM Switches. *IEEE JSAC*, 15(5):938–950, June 1997.
- [13] B. K. Ryu. *Fractal Network Traffic: From Understanding to Implications*. PhD thesis, CTR, Columbia University, May 1996.
- [14] M. Schwartz. *Broadband Integrated Networks*. Prentice Hall PTR, 1996.
- [15] J. S. Turner. New Directions in Communications (or Wich Way in the Information Age?). *IEEE Comm. Mag.*, 24(10):8–15, Oct. 1986.
- [16] H. Zhang. Services Disciplines for Guaranteed Performance Service in Packet-Switching Networks. *Proceedings of the IEEE*, 83(10), October 1995.
- [17] L. Zhang. Virtual Clock: A New Traffic Control Algorithm for Packet Switching Networks. In Proc. of *ACM SIGCOMM'90*, pag. 19-29, Sep. 1990.