

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
INFORMÁTICA INDUSTRIAL

THIAGO STAHLSCHMIDT

CONTROLE DE ADMISSÃO DE CONEXÃO PARA
SERVIDORES DE VÍDEO SOB DEMANDA UTILIZANDO A
TEORIA ASSINTÓTICA DE MUITAS FONTES

DISSERTAÇÃO

CURITIBA

2010

THIAGO STAHLSCHMIDT

**CONTROLE DE ADMISSÃO DE CONEXÃO PARA
SERVIDORES DE VÍDEO SOB DEMANDA UTILIZANDO A
TEORIA ASSINTÓTICA DE MUITAS FONTES**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial como requisito parcial para obtenção do Grau de Mestre em Ciências. Área de concentração: Telemática.

Orientador: Prof. Dr. Walter Godoy Jr.

Co-Orientador: Prof. Dr. Augusto Foronda

CURITIBA

2010

Dissertação de Mestrado sob o título “*Controle de admissão de conexão para servidores de vídeo sob demanda utilizando a teoria assintótica de muitas fontes*”, defendida por Thiago Stahlschmidt, aprovada em 2010, em Curitiba, pela banca examinadora constituída pelos professores:

Prof. Dr. Walter Godoy Jr.
Orientador

Prof. Dr. Emilio Carlos Gomes Wille
Universidade Tecnológica Federal do Paraná

Prof. Dr. Fábio Favarim
Universidade Tecnológica Federal do Paraná

Prof. Dr. Luiz Carlos Pessoa Albini
Universidade Federal do Paraná

RESUMO

STAHLSCHMIDT, Thiago. Controle de admissão de conexão para servidores de vídeo sob demanda utilizando a teoria assintótica de muitas fontes. 2010. 69 f. Dissertação – Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2010.

O presente trabalho propõe uma arquitetura de controle de admissão de conexão para servidores de vídeo sob demanda, que visa introduzir garantias determinísticas para o atraso e aumentar a taxa de utilização do enlace. O sistema tira proveito do fato de que os arquivos de vídeo podem ser pré-processados, para calcular os parâmetros de tempo e espaço, com o objetivo de estimar a banda efetiva de um agregado de *streams* sobre o regime assintótico de muitas fontes. A geração automática dos arquivos de *trace* permite a caracterização total do processo de chegada dos dados, aumentando a precisão do cálculo da taxa de serviço requerida. Para contornar o problema de predição de tráfego, uma vez que o sistema não captura o comportamento dos fluxos de vídeo no domínio do tempo, o enlace é dividido em vários enlaces virtuais, que transportam somente fluxos de vídeos idênticos. A utilização de um mecanismo, que compartilha a banda excedente, diminui o desperdício de recursos e a probabilidade de ocorrer uma violação aos parâmetros de *QoS* impostos. Todo processo de admissão de conexão é auxiliado por um sistema gestor de base de dados, onde uma função decide se deve ou não aceitar uma conexão pela simples comparação das somas das bandas efetivas das conexões com os recursos disponíveis. A eficiência do modelo proposto é analisada e comparada aos modelos determinísticos tradicionais.

Palavras-Chave: Banda Efetiva, CAC, Teoria Assintótica de Muitas Fontes, Vídeo-sob-Demanda, QoS.

ABSTRACT

STAHLSCHMIDT, Thiago. A connection admission control scheme based on the many sources asymptotic for video-on-demand servers. 2010. 69 f. Dissertação – Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2010.

This work proposes a admission control architecture for video-on-demand servers, which provide deterministic guarantees on the maximum delay of video traffic and increase the network resource utilization. The framework take advantage of the fact that stored video streams can be processed off-line to calculate the space and time parameters in order to estimate its effective bandwidth under the many sources asymptotic regime. The automatic generation of video data representation (packetized and encapsulated in network protocols) in textual format allows a full characterization of the underlying process so as to precisely calculate its bandwidth requirements. To bypass the traffic prediction problem, since the off-line estimation does not capture the exact dynamics of the video stream in the time domain, the link is divided into several virtual links, where only streams with same content are multiplexed (group multiplexing approach). The introduction of a borrowing mechanism to distribute the excess bandwidth to other groups which request service increases the resource utilization and QoS. The admission control is aided by a Database Management System, where a stored procedure decides whether to accept or reject a connection by simple compare the sum of effective bandwidths for the traffic streams with the available resources. The efficiency of the model is analyzed and compared with deterministic approaches.

Palavras-Chave: Effective Bandwidth, CAC, Many Sources Asymptotic, Video-on-Demand, QoS.

LISTA DE FIGURAS

1	Processo básico de <i>streaming</i>	14
2	Transporte do fluxo de vídeo sobre <i>RTP</i>	16
3	Multiplexação estatística dos fluxos de dados	22
4	Banda efetiva de uma janela de tráfego	23
5	Método de Busca por Seção Áurea	32
6	Arquitetura do sistema de Controle de Admissão de Conexão	36
7	Processo de chegada de dados	37
8	Diagrama ER do controle de admissão	41
9	Esquema de controle de tráfego utilizando <i>HTB</i>	43
10	Construção da base de dados que auxilia o controle de admissão	46
11	Processo de admissão de conexão	47
12	Enlace alvo dos experimentos	48
13	Trace de rede utilizado nos experimentos	49
14	Comparação das alocações por taxa de pico, banda efetiva e taxa média ..	50
15	Atraso máximo das alocações por taxa de pico e banda efetiva	51
16	Atraso máximo da alocação por taxa média	52
17	Taxa de utilização das abordagens analisadas	53

LISTA DE TABELAS

1	Conversão do <i>trace</i> de vídeo em <i>trace</i> de rede	38
2	Comportamento dos estimadores selecionados	39
3	Cálculo do número máximo de fluxos em função da <i>QoS</i> desejada	40
4	Capacidade mínima vs número de fluxos admitidos	44
5	Desperdício de recursos com o modelo de multiplexação em grupo	45

LISTA DE SIGLAS

AVI	<i>Audio Video Interleave</i>
CBR	<i>Constant Bitrate</i>
ER	<i>Entity-Relationship</i>
FBM	<i>Fractional Brownian Motion</i>
FIFO	<i>First In, First Out</i>
GA	<i>Gaussian Aproximation</i>
HTB	<i>Hierachical Token Bucket</i>
IID	<i>Independent and Identically Distributed</i>
IP	<i>Internet Protocol</i>
LB	<i>Leaky Bucket</i>
LRD	<i>Long-range Dependency</i>
MBM	<i>Multifractional Brownian Motion</i>
MPEG	<i>Moving Picture Experts Group</i>
MSAEB	<i>Many Sources Asymptotic Effective Bandwidth</i>
MTU	<i>Maximum Transmission Unit</i>
NEB	<i>Norros Effective Bandwidth</i>
NS-2	<i>Network Simulator 2</i>
QoS	<i>Quality of Service</i>
RTCP	<i>Real-time Transport Control Protocol</i>
RTP	<i>Real-time Transport Protocol</i>
SGDB	<i>Sistema Gestor de Base de Dados</i>
SRD	<i>Short-range Dependency</i>
TB	<i>Token Bucket</i>
UDP	<i>User Datagram Protocol</i>
VBR	<i>Variable Bitrate</i>
VoD	<i>Video on Demand</i>

SUMÁRIO

1	Introdução	10
1.1	Motivação	10
1.2	Contribuições	11
1.3	Organização	11
2	Fundamentos Conceituais e Trabalhos Relacionados	13
2.1	Introdução	13
2.2	Video sob Demanda	13
2.2.1	Codificação	14
2.2.2	Caracterização do Tráfego	15
2.2.3	Transporte	15
2.3	Qualidade de Serviço	16
2.4	Controle de Tráfego	18
2.5	Modelagem de Tráfego	19
2.6	Controle de Admissão de Conexão	20
3	Teoria de Banda Efetiva	22
3.1	Introdução	22
3.2	Princípio dos Grandes Desvios	23
3.3	Definição Formal de Banda Efetiva	24
3.3.1	Propriedades da Definição	25
3.3.2	Os Parâmetros s (espaço) e t (tempo)	25
3.4	Estimadores de Banda Efetiva	26
3.4.1	Estimador Imediato e em Bloco	27
3.4.2	Estimador de Courcoubetis	27
3.4.3	Estimador de Norros	28
3.4.4	Aproximação Assintótica por Muitas Fontes	29
3.4.5	Aproximação Gaussiana	32

4	Controle de Admissão para Servidores de Vídeo sob Demanda.....	34
4.1	Introdução.....	34
4.2	Arquitetura.....	35
4.3	Processo de Chegada de Dados.....	36
4.4	Construção e Conversão do Trace de Vídeo.....	37
4.5	Seleção do Estimador Adequado.....	38
4.6	Cálculo do Número Máximo de Conexões Admitidas.....	39
4.7	Controle de Admissão Auxiliado por SGBD.....	41
4.8	Controle da Demanda.....	42
4.9	Classificação e Controle de Tráfego.....	42
4.10	Estratégia de Multiplexação por Grupos.....	44
4.11	Funcionamento do Sistema.....	45
5	Simulações.....	48
5.1	Introdução.....	48
5.2	Cenário.....	49
5.3	Ganho Obtido com a Multiplexação Estatística.....	50
5.4	Verificação da Qualidade de Serviço.....	51
5.5	Taxa de Utilização do Enlace.....	52
6	Conclusões.....	54
6.1	Trabalhos Futuros.....	54
	Referências.....	56
	Apêndice A – Simulador de Streams (NS-2).....	61
	Apêndice B – Base de Dados do Controle de Admissão (MySQL).....	66
	Apêndice C – Classificação e Controle de Tráfego (IPTABLES/TC).....	69

1 Introdução

1.1 Motivação

Conhecer a banda requerida pelos fluxos de dados é crucial para fornecer qualidade de serviço (*QoS*) sem que haja desperdício da capacidade do enlace. As abordagens tradicionais para o cálculo da largura de banda necessária para garantir um certo nível de *QoS* são em sua grande maioria inadequadas. O cálculo da taxa média de transferência de dados em intervalos da ordem de minutos, por exemplo, falha em capturar o tráfego em rajada, que é relevante quando se deseja garantir um determinado desempenho. As abordagens baseadas na teoria das filas também não são adequadas, visto que requerem modelos de tráfego elaborados e não podem ser aplicadas efetivamente ao contexto das grandes redes multiserviços (COURCOUBETIS; SIRIS, 1999). Além disso, evidências de comportamento estatístico auto-similar (fractal) em vários tipos de tráfego invalidaram os modelos tradicionais (GROSSGLAUSER; BOLOT, 1996).

O conceito de banda efetiva, um escalar que estima a quantidade de recursos utilizados por um fluxo, é amplamente aceito como um dos mais apropriados para a alocação de recursos e controle de admissão em redes de dados (GIBBENS; KELLY, 1997). Nos enlaces que utilizam serviços com taxa variável de bits (*VBR*) a utilização do conceito permite o transporte de fluxos de dados multiplexados a uma taxa menor que a soma total das taxas de pico sem que a qualidade das conexões seja degradada.

Como previsto por Izquierdo e Reeves (1999), o vídeo compactado com taxa variável de bits se espalhou por toda Internet, sendo responsável por grande parte do volume de tráfego das redes (CHENG, 2007). Isto se deve a sua qualidade consistente, suas inerentes vantagens de ganho com a multiplexação estatística e a vasta gama de aplicações que o utilizam, tais como o vídeo sob demanda, vídeo conferência e a televisão digital de alta definição. Devido a crescente importância deste tipo de tráfego, suas restrições temporais e a ausência de garantia de serviço nas redes *IP* (SILVA, 2004), este trabalho propõe uma arquitetura de admissão de conexão que utiliza os resultados da teoria de multiplexação

estatística, quando várias fontes são multiplexadas e um desempenho mínimo é desejado.

1.2 Contribuições

O método proposto neste trabalho se diferencia dos demais, no sentido que não envolve medições periódicas ou um modelo de tráfego específico para determinar a banda efetiva, mas se baseia em arquivos de texto, que detalham a chegada dos dados no tempo (*traces*). A utilização dos arquivos de *trace* são, em geral, restritas as aplicações de simulações, no entanto, neste trabalho eles são utilizados para determinar com maior precisão o valor da banda efetiva.

O processo de geração dos arquivos de *trace* é automatizado de modo a simular a passagem de um fluxo de vídeo por uma rede de pacotes. Dessa forma, a abordagem dispensa a utilização de ferramentas de análise de pacotes e não sofre dos problemas que afetam as abordagens baseadas em medições, tais como, a definição dos intervalos de agregação dos dados (STAHLSCHMIDT; GODOY JR.; FORONDA, 2010).

Através dos *traces* e da teoria assintótica de muitas fontes é calculada a taxa mínima de serviço que garante, com uma certa probabilidade, um atraso máximo na entrega dos dados. A banda efetiva de cada agregado de fluxos de vídeos é pré-calculada, de acordo com uma projeção do crescimento da demanda, e armazenada em uma tabela de banco de dados, onde fornece os subsídios necessários para o controle de admissão.

Uma vez que o sistema não captura o comportamento dinâmico dos fluxos de vídeo no tempo, a multiplexação estatística é considerada somente entre os fluxos que transmitem o mesmo conteúdo. A introdução de um mecanismo de controle de tráfego possibilita contornar o problema da predição dos fluxos através da criação de vários enlaces virtuais, onde é viável prever o tráfego com antecedência e garantir os recursos necessários de acordo com a qualidade de serviço desejada. Além disso, a distribuição da banda excedente entre as classes que demandam serviço, aumenta a taxa de utilização dos recursos ao mesmo tempo em que diminui as chances de ocorrer uma violação aos parâmetros impostos.

1.3 Organização

O restante deste texto está organizado como descrito a seguir. O Capítulo 2 é dedicado a revisão de alguns conceitos e ao estudo de técnicas e mecanismos que tem

por objetivo introduzir qualidade de serviço na transmissão de vídeos sobre redes *IP*. O Capítulo 3 apresenta os conceitos básicos da teoria de banda efetiva e revisa alguns dos principais estimadores disponíveis na literatura. Já no Capítulo 4, as idéias introduzidas nos capítulos anteriores são organizadas para a proposta de uma arquitetura de admissão de conexão para servidores de *streaming* sob demanda. O Capítulo 5 demonstra a validade da arquitetura utilizando *traces* de arquivos de vídeo (*MPEG-4*) e um simulador de redes (*NS-2*). Finalmente, o Capítulo 6 apresenta os resultados do trabalho, discute seus méritos, problemas e identifica algumas possibilidades de desenvolvimentos futuros.

2 Fundamentos Conceituais e Trabalhos Relacionados

2.1 Introdução

Este capítulo tem por objetivo revisar de forma bastante direcionada e sucinta os conceitos que são utilizados ao longo deste trabalho. A medida em que os conceitos são expostos, os esforços realizados anteriormente em soluções correlatas são descritos e analisados.

A revisão de contexto inicia pela descrição do serviço de vídeo sob demanda, sua estrutura, seus requisitos e alguns detalhes relevantes inerentes à codificação e ao transporte dos vídeos sobre uma rede de pacotes. Na seqüência, o conceito de qualidade de serviço é revisto e algumas técnicas que podem ser utilizadas na introdução de garantias determinísticas em redes que operam segundo o paradigma do melhor esforço são analisadas.

2.2 Video sob Demanda

O vídeo sob demanda (VoD) é uma tecnologia de fluxo de mídia (*streaming*) na qual os arquivos de vídeo permanecem armazenados em um servidor, prontos para serem enviados via rede sob requisição dos clientes. A arquitetura básica do sistema e o processo de *streaming* podem ser vistos na Figura 1.

Um sistema de *VoD* consiste basicamente de um servidor (que armazena o acervo de vídeos), clientes (que fazem requisições remotas para visualizar os vídeos, através de uma interface web ou uma aplicação específica) e uma rede de distribuição de conteúdo (responsável por interconectar clientes e servidores) (PINHO; ISHIKAWA; AMORIM, 2003).

O *VoD* está entre os serviços de multimídia de maior expansão devido a sua vasta aplicabilidade. O vídeo sob demanda pode ser empregado na área de entretenimento, na

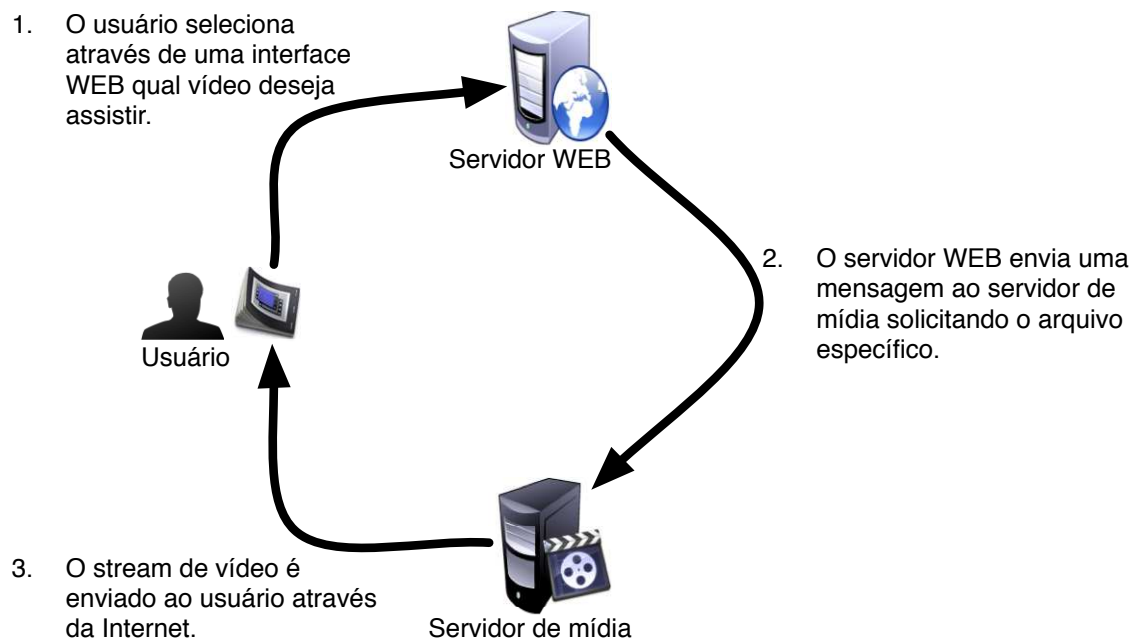


Figura 1: Processo básico de *streaming*

distribuição de filmes, videoclipes e na programação personalizada de televisão, assim como na área educacional, nos projetos de ensino a distância. O sucesso de um sistema de *VoD* depende de um meio de transmissão que atenda requisitos de qualidade de serviço, tais como a largura de banda e o atraso.

2.2.1 Codificação

A compactação dos vídeos é extremamente importante para as aplicações de *VoD*. Primeiramente, porque as informações armazenadas nos servidores de vídeo são extremamente volumosas. Segundo, porque mesmo em uma rede de alta velocidade não seria possível manter o tráfego de informações de vídeo de forma não comprimida por muito tempo.

O principal algoritmo de codificação de vídeo é o *MPEG*, que possui versões conhecidas como *MPEG-1*, *MPEG-2* e *MPEG-4* (RICHARDSON, 2003). O *MPEG-4* foi desenvolvido para uso na Internet, mas seu uso foi também suportado pelas novas mídias de alta definição, como o *Blu-ray*, telefones celulares e diversos padrões de transmissão de TV digital, especialmente os de alta definição (*HDTV*) (AVRAMOVA et al., 2009).

O padrão *MPEG-4* começou a ser concebido, pelo grupo *MPEG*, em julho de 1993, tendo sido aprovado como padrão internacional em 2000. Assim como os padrões anteriores, o *MPEG-4* permite estabelecer diversos valores de taxas de compressão, con-

forme a aplicação a ser efetuada, podendo comprimir mais ou menos o conteúdo original. No entanto, ao contrário do *MPEG-2*, cuja qualidade é mais ou menos fixa em torno do padrão de qualidade do *DVD*, no *MPEG-4* essa variação é bem maior, uma gama muito grande de valores pode ser utilizada, permitindo a visualização do vídeo mesmo nos meios de transmissão com baixa capacidade.

Assim como suas versões anteriores, o *MPEG-4* é na verdade o nome dado a um conjunto de tópicos denominados "partes". Cada parte aborda um aspecto diferente do padrão. Assim, por exemplo, no *MPEG-4* há atualmente um total de 16 partes. A parte 1 descreve a sincronização de áudio e vídeo, a parte 2 o processo de compressão do vídeo, a parte 3 o processo de compressão do áudio, a parte 4 procedimentos para verificar a conformidade de determinada amostra com outras partes do padrão, a parte 5 software para demonstrar e ilustrar determinadas partes do padrão e assim por diante (RICHARDSON, 2003). Maiores informações sobre esse padrão de compressão de dados digitais de áudio e vídeo podem ser encontradas no livro de Pereira e Ebrahimi (2002).

2.2.2 Caracterização do Tráfego

Os vídeos armazenados no servidor podem ser codificados basicamente de duas formas. Uma forma é gerando uma saída a uma taxa de bits constante (*CBR*). Essa codificação facilita o provisionamento de banda pois gera dados há uma taxa fixa e previsível. Todavia, esse tipo de codificação possui a desvantagem de desnivelar a qualidade do vídeo gerado, pois ao compactar cenas complexas, para não exceder a taxa de bits, a qualidade é sacrificada, enquanto que as cenas mais simples são codificadas com uma qualidade maior que o necessário resultando em desperdício de dados.

Já na codificação com taxa variável de bits (*VBR*) a taxa do vídeo gerado varia de acordo com a complexidade da cena, propiciando uma melhor qualidade. A transmissão de vídeos *VBR* faz com que a taxa de pico da fonte atinja valores consideravelmente maiores que a média e isso possibilita a obtenção de ganhos com a multiplexação estatística. No entanto, o ganho estatístico apresenta desafios para o provisionamento de recursos (CALLADO; KAMIENSKI; KELNER, 2002).

2.2.3 Transporte

O *Real-time Transport Protocol (RTP)* é um protocolo extensivamente utilizado em sistemas de comunicação e entretenimento que envolvem a distribuição de conteúdo

multímídia através da Internet. O *RTP* fornece suporte ao transporte de dados de tempo real, como fluxos de áudio e vídeo. Os serviços oferecidos pelo protocolo incluem marcação de tempo, detecção de perda e identificação do conteúdo (SCHULZRINNE et al., 2003).

O protocolo *RTCP*, definido também através da *RFC3550*, auxilia o *RTP*, fornecendo estatísticas out-of-band (comunicação que ocorre fora da conexão previamente estabelecida) sobre a qualidade de serviço da transmissão para os participantes da sessão. Normalmente o *RTP* utiliza uma porta par para enviar o fluxo de mídia e o *RTCP* utiliza a próxima porta ímpar disponível.

O *RTP* é transportado sobre o *UDP* (KUROSE; ROSS, 2009), um protocolo não orientado a conexão, porque para o transporte de dados em tempo real o tempo de entrega é mais importante que a confiabilidade, ou seja, retransmissões são indesejáveis. A Figura 2 mostra o encapsulamento do fluxo de vídeo para o transporte.

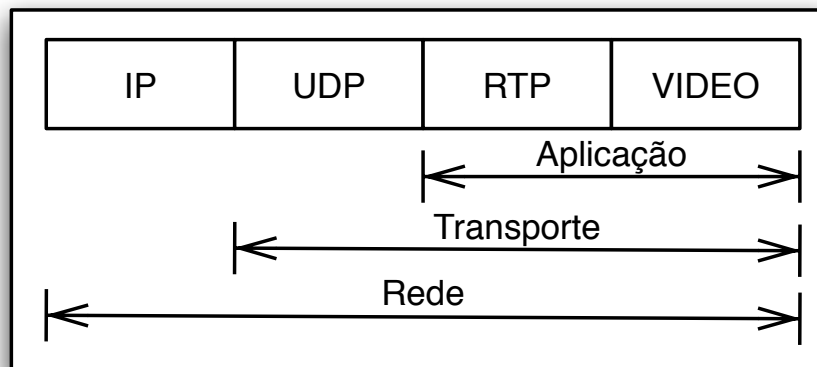


Figura 2: Transporte do fluxo de vídeo sobre *RTP*

2.3 Qualidade de Serviço

Atualmente, a maioria das redes *IP* não fornece qualquer garantia de qualidade de serviço. Essas redes são baseadas na política do melhor-esforço, a qual é apropriada para a maioria das aplicações de dados, mas não é adequada para as aplicações que possuem restrições temporais (KUROSE; ROSS, 2009). Para ampliar o suporte a esses serviços o conceito de qualidade de serviço (*QoS*) deve ser incorporado a essas redes (WYDROWSKI; ZUKERMAN, 2002).

O objetivo da introdução do conceito é privilegiar algumas aplicações, garantindo que certos parâmetros desses fluxos sejam mantidos em um nível mínimo sob quaisquer

circunstâncias. Os principais parâmetros de *QoS* são:

- Atraso (latência)
- Variação do atraso (*jitter*)
- Perda de Pacotes (confiabilidade)
- Vazão (capacidade)

O atraso pode ser entendido como o somatório dos atrasos impostos pela rede e pelos equipamentos utilizados na comunicação. Do ponto de vista da camada superior, a latência representa um tempo de resposta, que é o tempo de entrega da informação para a aplicação. Como os atrasos de fila nos roteadores são aleatórios, o tempo decorrido entre o momento em que um pacote é gerado na fonte e o momento em que ele é recebido no destinatário pode variar de pacote para pacote. Esse fenômeno é denominado de variação do atraso. A perda de pacotes em redes *IP* ocorre principalmente devido à erros na camada de enlace e ao congestionamento dos ativos de rede. As aplicações sempre precisam de vazão e, assim sendo, esse é o parâmetro mais básico e certamente mais presente nas especificações de *QoS* (SILVA, 2004).

De um modo geral, todos os parâmetros são importantes para a qualidade do serviço. Na prática, a relevância dos parâmetros varia de acordo com a necessidade de cada aplicação, ou seja, alguns parâmetros são mais importantes que outros. Diversos fatores podem afetar os parâmetros listados, o atraso, por exemplo, pode ser introduzido por um roteador ou até mesmo pela forma como a pilha de protocolos foi implementada no sistema operacional.

Existem diversas técnicas que podem ser utilizadas para introduzir *QoS* em uma rede. Algumas são bastante simples, como o super-dimensionamento de recursos, que procura oferecer recursos suficientes de acordo com a taxa de pico esperada. A alocação por taxa de pico é simples e eficaz, mas na prática é cara, pois faz com que os recursos sejam subutilizados. Quase sempre as soluções mais eficientes envolvem a combinação de duas ou mais técnicas. Neste trabalho, as técnicas de controle de tráfego e controle de admissão são combinadas para garantir um atraso máximo na transmissão dos dados.

2.4 Controle de Tráfego

O controle de tráfego provê mecanismos de controle para diminuir a latência e aumentar a taxa de utilização dos recursos de um enlace (TANENBAUM, 2002). Basicamente, o controle consiste em atrasar o tráfego em excesso, através de mecanismos de enfileiramento, retendo e liberando os pacotes de tal maneira que o fluxo de saída permaneça dentro dos parâmetros definidos. O controle de tráfego é útil, por exemplo, para limitar o tráfego de rajada de forma a não prejudicar o tráfego prioritário (reduzindo a latência) e também nas situações de congestionamento, limitando um determinado tipo de tráfego não sensível a retardo, eliminando assim possíveis gargalos. Entre os algoritmos de regulação de tráfego os mais utilizados são o *Leaky Bucket (LB)* e o *Token Bucket (TB)*, ambos têm propriedades e finalidades distintas (KUROSE; ROSS, 2009).

O mecanismo *LB* tem como objetivo transformar um tráfego com rajadas em um fluxo onde a taxa máxima de transmissão é controlada, realizando assim uma moldagem isócrona. O mecanismo tolera rajadas na entrada, mas impede que elas ocorram na saída, através da utilização de um *buffer*. Conceitualmente, o *LB* pode ser entendido como um contador que é incrementado sempre que um pacote chega para ser transmitido e é decrementado periodicamente (determinando assim a taxa média de transmissão). Se o contador exceder um limite fixado o pacote é descartado (TURNER, 1986). Ao produzir uma saída com taxa fixa e não permitir rajadas, o *LB* pode causar a subutilização dos recursos.

O *TB* tem como objetivo controlar a taxa média de transmissão de um tráfego com rajadas, permitindo a transmissão de rajadas com uma taxa de transmissão instantânea superior a média, mas limitando a sua duração, isto é, o *TB* permite a passagem de tráfego com rajadas, mas com intensidade limitada. O *TB* é baseado em um container de *tokens* com capacidade para b *tokens*, que são gerados de acordo com a taxa média de transmissão r . A transmissão de um pacote de n bytes só é possível depois da retirada de n *tokens* do container. Enquanto o container permanece vazio, nenhum pacote é transmitido, quando ele enche, a produção de *tokens* é suspensa. A presença de vários *tokens* no container permite a transmissão de rajadas com taxas de transmissão instantânea superior a r (TANENBAUM, 2002).

2.5 Modelagem de Tráfego

A modelagem de fontes de tráfego tem por objetivo a criação de modelos matemáticos que capturam o comportamento estatístico das informações geradas por essas fontes. A partir desses modelos é possível conhecer o impacto aproximado causado pela transmissão de dados da fonte modelada (WILLIAMSON, 1995).

Quanto mais realísticos forem os modelos, mais precisas são as estimativas sobre a utilização dos recursos da rede, assim maior precisão é garantida à qualidade de serviço. A eficácia de um modelo está relacionada a sua proximidade com as fontes reais, sua simplicidade (modelo descrito por um número pequeno de parâmetros) e facilidade de implementação.

Várias soluções foram propostas na literatura para determinar a banda efetiva de um fluxo de dados com base em modelos (KELLY, 1996). Entretanto, a eficácia dos modelos Markovianos foram questionadas após exposição do fenômeno da auto-similaridade do tráfego de redes (LELAND et al., 1994). Com isso surgiram os modelos auto-similares, como o descrito por Norros (1995). Nesse trabalho, o autor introduz um modelo baseado no conhecido Movimento Browniano Fractal (*FBM*) e deriva o cálculo da banda efetiva para esse modelo.

Mais tarde, pesquisas mostraram que o tráfego de redes pode ter propriedades estatísticas e comportamentos em escala mais complexos do que os representados nos modelos auto-similares, com a presença de comportamento não-trivial em pequenas escalas de tempo (PARK; WILLINGER, 2000).

Para tentar capturar essas singularidades foram introduzidos os modelos multifractais. O uso desses modelos é relevante na análise do tráfego de vídeo, pois a natureza do fluxo do protocolo de transporte utilizado pelo *RTP*, o *UDP*, é multifractal (OSTROWSKY, 2005). Segundo Melo (2004), caracterizar fluxos multifractais através de modelos fractais tem como consequência a superestimação dos recursos requeridos por esses fluxos.

A falta de consenso sobre modelos apropriados para caracterizar o tráfego de vídeo fez crescer o número de soluções baseadas em medições. O uso dessas soluções contorna os problemas criados pela adoção de modelos específicos através da extração das características estatísticas diretamente do tráfego.

2.6 Controle de Admissão de Conexão

O controle de admissão de conexão (CAC) é responsável por decidir, após avaliar o impacto, se deve ou não aceitar uma nova conexão, uma vez que, se for admitida, as características demandadas pelo tráfego específico devem ser respeitadas ao mesmo tempo que a qualidade de serviço das conexões existentes não pode ser afetada (PERROS; ELSAYED, 1996).

O controle de admissão deve ser capaz de fornecer uma resposta em tempo real, isso demanda métodos simples, aproximados ou baseados em tabelas, pois cálculos numéricos são inviáveis. Conhecendo os parâmetros de QoS desejado e o tráfego, deve-se verificar se há recursos suficientes para atender os requisitos solicitados.

Os protocolos utilizados na Internet se valem de mecanismos na camada de transporte para detectar e reagir ao congestionamento. Esses mecanismos são essenciais para assegurar uma divisão justa de recursos entre os fluxos de dados, mas são insuficientes para garantir os requisitos das aplicações de tempo real. Por isso, é necessário utilizar técnicas de admissão de conexão para fornecer as garantias necessárias para essas aplicações que são intolerantes às variações de atraso (inelásticas).

As técnicas utilizadas para alocação de capacidade em redes de dados podem ser estatísticas ou determinísticas (KROPOTOFF, 1999). As determinísticas alocam os recursos pela taxa de pico, sua implementação é bastante simples, mas há grande desperdício de capacidade. Por outro lado as alocações estatísticas são mais complexas e maximizam a utilização dos recursos da rede. O problema da alocação estatística pode ser esquematizado como um problema de filas, onde o objetivo é encontrar o número máximo de conexões que podem ser atendidas sem violar os requisitos definidos. Como as conexões tem comportamentos distintos, complexos e muitas vezes imprevisíveis a resolução desse problema é bastante árdua. Por isso as soluções mais plausíveis se baseiam em aproximações que não exigem a solução exata do problema (KROPOTOFF, 1999).

Existe uma grande variedade de mecanismos para controle de admissão na literatura (ABDALLA, 1996). Neste trabalho é utilizada a aproximação por banda efetiva. Nessa abordagem se deseja saber qual fração da capacidade do canal e do tamanho do *buffer* precisam ser reservados para se admitir um determinado número de fontes e garantir uma probabilidade máxima para perda de pacotes ou atraso.

O conceito de banda efetiva, simplifica consideravelmente os processos de dimensionamento de recursos e controle de admissão, provendo um método para quantificar os

requisitos mínimos de serviço em uma conexão. Dado um tamanho de *buffer*, a banda efetiva representa a taxa de serviço que é efetivamente necessária para servir um fluxo de tráfego de acordo com o nível de serviço desejado. Devido a importância da teoria de banda efetiva no desenvolvimento do sistema proposto, o Capítulo 3 é inteiramente dedicado ao seu estudo.

3 Teoria de Banda Efetiva

3.1 Introdução

As redes de banda larga normalmente integram vários fluxos de dados com diferentes características de tráfego. As diversas fontes transmitem em diferentes taxas, que podem variar entre 0 e algum valor de pico. Quando se deseja garantir uma determinada qualidade de serviço é intuitivo reservar a capacidade necessária utilizando como parâmetro o valor de pico da conexão. Considerando os tráfegos agregados e permitindo que as taxas de pico acumuladas dos diferentes fluxos excedam a capacidade total disponível do enlace, é possível conseguir um ganho estatístico, aumentando assim a utilização dos recursos de rede. Conforme pode ser visto na Figura 3.

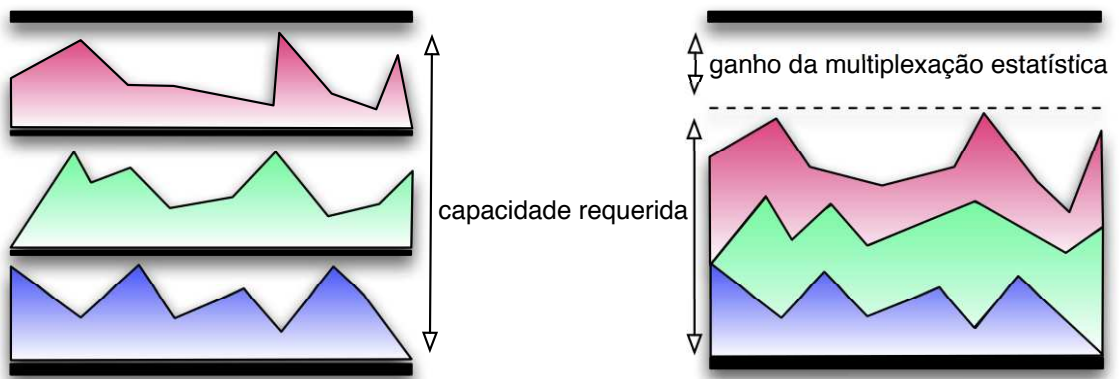


Figura 3: Multiplexação estatística dos fluxos de dados

A idéia por trás do conceito de multiplexação estatística é que nem todas as fontes transmitirão dados à taxa máxima ao mesmo tempo, dessa forma, utilizando uma abordagem estatística para calcular a taxa mínima de serviço, é possível permitir que mais fluxos de dados compartilhem o mesmo canal. O aumento da taxa de utilização do canal sem violar a qualidade do serviço pode ser alcançado através do conceito de banda efetiva (HUI, 1988).

O objetivo da teoria de banda efetiva é estimar a quantidade mínima de banda

necessária, dado as características da fonte e o tamanho do *buffer* B , para que os parâmetros de QoS (e.g. a probabilidade máxima de perda de pacotes) não sejam violados. O valor da banda efetiva é um escalar entre a taxa média e a máxima do tráfego agregado (KELLY, 1996), como pode ser visto na Figura 4.

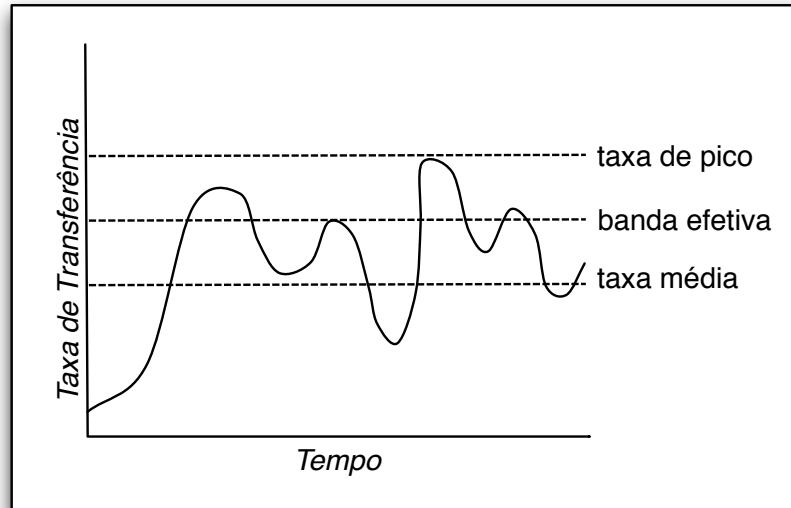


Figura 4: Banda efetiva de uma janela de tráfego

O conceito de banda efetiva foi inicialmente proposto por Hui (1988), mais tarde muitos pesquisadores desenvolveram suas próprias abordagens para estimar o valor para diferentes modelos de tráfego (GUERIN; AHMADI; NAGHSHINEH, 1991; NORROS, 1995; FONSECA; MAYOR; NETO, 2000). A banda efetiva é usualmente definida utilizando análise assintótica, que tem por objetivo definir como a probabilidade de perda de pacotes decresce a medida que alguma quantia aumenta (e.g. o tamanho do *buffer* ou o número de fluxos de dados).

3.2 Princípio dos Grandes Desvios

A Teoria dos Grandes Desvios é um conjunto de técnicas para estimar algumas propriedades de eventos raros, tais como suas frequências e a forma mais provável de ocorrerem (WEISS, 1995). Eventos raros são causados por um conjunto de coisas improváveis que ocorrem ao mesmo tempo. Um evento raro não deve ser confundido com um evento único de baixa probabilidade. Ser um vencedor em um concurso de prognóstico numérico, por exemplo, não é um evento raro, visto que não pode ser decomposto em outros eventos menores (WEISS, 1995). Um evento raro nas telecomunicações pode acontecer, por exemplo, quando um servidor de páginas divulga o resultado de um vestibular e várias pessoas

resolvem conferir a lista de aprovados ao mesmo tempo, levando a um congestionamento. Em redes de alto desempenho a perda de um pacote também pode ser considerado um evento raro.

Uma das finalidades da Teoria, que é utilizada na determinação da banda efetiva, é fornecer uma descrição assintótica da probabilidade de que o tamanho do *buffer* de dados ultrapasse sua capacidade máxima de armazenamento (transbordo de *buffer*). A probabilidade de transbordo do *buffer* é calculada através de um método sistemático de cálculo da função taxa. A função taxa controla o decaimento da cauda de uma distribuição de probabilidade e pode ser estimada por meio de funções geradoras de momentos.

De um modo geral, para se calcular a probabilidade de um evento raro, deve-se associar um custo a cada caminho de amostra que possa causar a ocorrência do evento e encontrar aquele que possui o menor custo. Essa probabilidade é estimada por:

$$P(\text{evento}) \approx e^{-n \cdot \text{custo}} \quad (3.1)$$

Onde n é um parâmetro assintótico, podendo ser por exemplo, o tamanho do *buffer* ou o número de fontes de dados.

A lei forte dos eventos raros diz que se existe um caminho único de menor custo que causa a ocorrência de um evento raro, com grande probabilidade este caminho é utilizado (WEISS, 1995). Em outras palavras os eventos raros ocorrem quase que de uma única maneira. Sendo assim a teoria não apenas pode estimar a probabilidade de um *buffer* atingir um determinado tamanho, mas pode também informar como o *buffer* alcançou este tamanho. Um passo essencial para prevenção de congestionamento em uma série de mecanismos (dimensionamento de *buffer*, alocação de recursos, controle de admissão) é determinar como ele ocorre, e neste sentido, o método de aproximação utilizando os grandes desvios é ideal (XIE; HAENGGI, 2002).

O Princípio dos Grandes Desvios é bastante abstrato para ser discutido extensivamente, por isso o escopo deste trabalho limita-se as principais equações e seus resultados. Maiores detalhes podem ser obtidos em Weiss (1995).

3.3 Definição Formal de Banda Efetiva

Considerando que \mathbb{E} é a esperança matemática (valor médio esperado de uma variável aleatória), $X[0, t]$ é a quantidade de tráfego gerado por uma determinada fonte

no intervalo de tempo $[0, t]$, e ainda que, X possui incrementos estacionários. A banda efetiva dessa fonte depende dos parâmetros s (espaço) e t (tempo) e é definida pela expressão (KELLY, 1996):

$$\alpha(s, t) = \frac{1}{st} \log \mathbb{E} (e^{sX[0, t]}) \quad (0 < s, t < \infty) \quad (3.2)$$

3.3.1 Propriedades da Definição

- i. Se $X[0, t]$ tem incrementos independentes, então $\alpha(s, t)$ não depende de t .
- ii. Se existe uma variável aleatória X tal que $X[0, t] = Xt$ para $t > 0$, então $\alpha(s, t) = \alpha(st, 1)$, dessa forma $\alpha(s, t)$ depende de s, t somente através do produto st .
- iii. Se $X[0, t] = \sum_i X_i[0, t]$ e os componentes da soma, $(X_i[0, t])_i$, são independentes, então

$$\alpha(s, t) = \sum_i \alpha_i(s, t). \quad (3.3)$$

- iv. Para qualquer valor fixo de t , $\alpha(s, t)$ é crescente em s , com valor entre a taxa média e a taxa de pico, ou seja

$$\frac{\mathbb{E}X[0, t]}{t} \leq \alpha(s, t) \leq \frac{\bar{X}[0, t]}{t} \quad (3.4)$$

3.3.2 Os Parâmetros s (espaço) e t (tempo)

Os parâmetros s (espaço) e t (tempo) não dependem apenas da fonte, mas de todo o contexto da conexão, que inclui, a taxa de serviço, o tamanho do *buffer*, o esquema de escalonamento, os parâmetros de *QoS* e as características individuais das fontes (COURCOUBETIS; SIRIS; STAMOULIS, 1999).

O parâmetro tempo representa a duração mais provável em que o *buffer* permanece cheio antes que ocorra uma perda de pacotes, além disso, o parâmetro t indica a granularidade mínima necessária para a observação das propriedades relevantes do tráfego (i.e. as propriedades estatísticas que mais contribuem para a perda de pacotes). O parâmetro espaço indica o grau de multiplexação estatística, conforme o valor de s aumenta o grau de multiplexação diminui. Um valor grande de s ocorre tipicamente quando se multiplexam fluxos com taxas de pico não muito menores que a capacidade do enlace.

Quando $s \rightarrow \infty$ o valor da banda efetiva se aproxima da taxa de pico medida no

intervalo t , se $s = \infty$ a multiplexação deve ser determinística e a probabilidade de perda de pacotes nula. Por outro lado, valores pequenos indicam um alto grau de multiplexação. Conforme $s \rightarrow 0$ o valor da banda efetiva se aproxima da taxa média. O parâmetro tempo pode ser medido em, por exemplo, milisegundos, enquanto que uma unidade de medida válida para o parâmetro espaço é $kbit^{-1}$.

3.4 Estimadores de Banda Efetiva

Na prática, a estimativa de banda usando a equação (3.2) não é uma tarefa trivial, para se determinar os parâmetros s e t com precisão é necessária a caracterização total do processo. A banda efetiva pode ser determinada de forma paramétrica ou através de medições. Na paramétrica é necessária a análise completa do processo, o que é feito com auxílios de modelos de tráfego. Através de amostras da fonte, é possível determinar a banda efetiva sem a necessidade de assumir um modelo de tráfego específico, o que é ideal no contexto das redes multiserviços (VIEIRA, 2006).

No caso da estimação através de medições é necessário definir a unidade de tempo em que o tráfego é agregado. Uma janela de observação contendo um ou mais intervalos de tempo é utilizada para o cálculo dos parâmetros necessários aos estimadores de banda efetiva. A escolha do intervalo de tempo e da janela de observação podem afetar drasticamente a estimação da banda efetiva, por isso, quando se estima a banda através de medições é necessário utilizar escalas de tempo variáveis, que dependem diretamente do tráfego analisado (HACIOMEROGLU, 2003).

Alguns algoritmos de banda efetiva são bastante complexos e podem não ser capazes de fornecer respostas em tempo real. Esse é o caso da aproximação assintótica por muitas fontes, porém, devido a sua eficácia e precisão na estimação de banda em tráfegos com dependência de curta ou de longa duração, esforços foram realizados na tentativa de reduzir a sua complexidade com o objetivo de tornar viável o seu emprego no controle de admissão (SERES et al., 2002). Também foram desenvolvidas técnicas para otimizar o cálculo do parâmetro de tempo, através de uma busca de valores em torno das últimas escalas de tempo críticas calculadas, diminuindo assim o tempo necessário para o cálculo da banda efetiva (FERNANDEZ-VEIGA et al., 2003).

Na seqüência, alguns dos principais estimadores disponíveis na literatura, são discutidos e comparados brevemente. Maiores informações podem ser encontradas nas referências citadas.

3.4.1 Estimador Imediato e em Bloco

A banda efetiva pode ser estimada diretamente pelo uso da equação (3.2), substituindo a expectativa do processo pela sua média temporal (GIBBENS, 1996):

$$\hat{\alpha}(s, t) = \frac{1}{st} \log \frac{1}{t_N - t} \int_0^{t_N - t} e^{s \sum_{i=1}^N x_i I(\tau \leq t_i \leq \tau + t)} d\tau \quad (0 \leq \tau \leq t_N - t) \quad (3.5)$$

sendo que N é o número de pacotes que chegaram durante o tempo de análise, t_i é o tempo de chegada do pacote i , x_i o tamanho do pacote i e o termo $\sum_{i=1}^N x_i I(\tau \leq t_i \leq \tau + t)$ representa a quantidade de dados que chegaram durante o intervalo $[\tau, \tau + t]$. A vantagem desse estimador é que ele não é restrito por qualquer suposição, porém a integral da equação (3.5) pode ser de difícil obtenção numérica.

O estimador em bloco também se baseia na equação (3.2), porém em contraste com o estimador imediato, considera blocos não-sobrepostos de chegadas de dados em um intervalo de tempo t (DUFFIELD et al., 1995):

$$\hat{\alpha}(s, t) = \frac{1}{st} \log \left[\frac{1}{T/t} \sum_{i=1}^{T/t} e^{s X_j[(i-1)t, it]} \right] \quad (3.6)$$

sendo que T é o tamanho do *trace* e $X_j[(i-1)t, it]$ é a quantidade de dados produzida no intervalo $[(i-1)t, it]$. O estimador supõe que os tempos de chegada dos blocos são variáveis aleatórias independentes e identicamente distribuídas (I.I.D) e por isso não é adequado para tráfego com dependência de longa duração (*LRD*). Sua precisão é comparável ao estimador imediato com um menor tempo de processamento (FALKNER et al., 2000).

3.4.2 Estimador de Courcoubetis

Courcoubetis e Weber (1995b) propuseram um método para estimação da banda efetiva que se baseia na Teoria dos Grandes Desvios. O método supõe que o *buffer* é muito grande, dessa forma, o parâmetro t , que está relacionado com o tempo em que o *buffer* permanece cheio antes que ocorra um transbordo, tende ao infinito. Para uma fonte estacionária, a banda efetiva pode ser calculada utilizando a equação:

$$\alpha = \mu + \frac{\gamma s}{2B} \quad (3.7)$$

sendo que

$$\gamma = \lim_{x \rightarrow \infty} \frac{1}{N} \text{var} \left(\sum_{n=1}^N X_n \right)$$

Os parâmetros μ , B , s e γ são, respectivamente, a taxa média, o tamanho do *buffer*, o parâmetro espaço e o índice de dispersão. O índice de dispersão mostra a variabilidade do processo em diferentes escalas de tempo (*burstiness*) e pode ser estimado através dos dados de tráfego, utilizando técnicas de estimação de espectro, como as descritas por Chatfield (2004).

Quando a taxa de serviço é $\alpha(s)$ e o *buffer* é grande, a cauda da distribuição de probabilidade do tamanho da fila Q_N pode ser aproximada por:

$$P(B < Q_N) \approx e^{-sB} \quad (3.8)$$

sendo que o parâmetro espaço (s) pode ser interpretado como a taxa de decaimento exponencial assintótico da distribuição do tamanho da fila e depende das restrições impostas pela qualidade de serviço desejada.

A equação (3.7) pode fornecer uma estimativa precisa da utilização dos recursos quando o *buffer* é grande, porém quando o *buffer* é pequeno a probabilidade de transbordo pode ser subestimada. Além disso o estimador também não é adequado para tráfego (*LRD*).

3.4.3 Estimador de Norros

As formulas de banda efetiva apresentadas até agora são baseadas na hipótese de que a probabilidade do transbordo do *buffer* decresce exponencialmente conforme o tamanho do *buffer* cresce. Todavia, estudos mostraram um comportamento auto-similar no tráfego de dados na Internet (LELAND et al., 1994). Devido as características do tráfego auto-similar, a diminuição da probabilidade do transbordo do *buffer* com o crescimento do *buffer* é mais lenta (apresenta decaimento hiperbólico) (GARRETT; WILLINGER, 1994).

Norros (1995) introduziu um modelo gaussiano auto-similar para modelagem de tráfego em redes reais, e definiu a banda efetiva para esse modelo:

$$\alpha = \mu + K(H) \sqrt{-2 \ln(\epsilon)}^{1/H} a^{\frac{1}{2H}} B^{-(1-H)/H} \mu^{\frac{1}{2H}} \quad (3.9)$$

sendo que $K(H) = H^H(1-H)^{1-H}$ e os parâmetros μ , H , ϵ , B e a são, respectivamente, a taxa média, o parâmetro de Hurst, a probabilidade de transbordo, o tamanho do *buffer* e o coeficiente de variação. Quando o tráfego possui dependência de curta duração o parâmetro a pode ser aproximado pelo índice de dispersão (γ), no caso de tráfego com dependência de longa duração, o parâmetro a é aproximado por:

$$a = r^{2H-1} x_0^{-2H} (2x_0 \mathbb{E}(U \wedge x_0) - \mathbb{E}(U \wedge x_0)^2) \quad (3.10)$$

sendo r a taxa de transmissão de rajadas, x_0 uma constante e U uma variável aleatória relacionada à distribuição do tamanho das rajadas.

O parâmetro de Hurst, denotado por H , é uma métrica que caracteriza o grau de auto-similaridade de um processo. Este parâmetro varia no intervalo $[0, 1]$, sendo que os valores acima de 0.5 representam as correlações positivas (processos com dependências de longa duração) e os valores abaixo as correlações negativas. O parâmetro de Hurst pode ser estimado de forma eficiente, utilizando *wavelets*, em particular através do método AV (VEITCH; ABRY, 1999). Quanto maior o valor do parâmetro de Hurst, maior é o impacto das auto-correlações no desempenho do sistema.

Esse estimador é ideal para o tráfego com longa-duração, porém a derivação da fórmula é feita utilizando a distribuição de Weibull para aproximar o decaimento da cauda da distribuição do tamanho do *buffer*, o que a torna precisa para *buffer* grandes, mas produz estimativas menos precisas quando o *buffer* é pequeno (DRUMMOND; FONSECA; DEVETSIKIOTIS, 2005).

3.4.4 Aproximação Assintótica por Muitas Fontes

Essa técnica envolve a resolução de um problema numérico para estimar o ponto de operação do enlace, ou seja os valores de tempo (t) e espaço (s) com os quais a banda efetiva se relaciona com uma determinada probabilidade assintótica de transbordo.

Seja $Q(Nc, Nb, Nn) = P(Q_N > B)$ a probabilidade de que em um *buffer* infinito que recebe dados de $Nn = (Nn_1, \dots, Nn_j)$ fontes e é servido a uma taxa $C = Nc$, o tamanho da fila (Q_N) cresça acima do limite $B = Nb$. De acordo com os resultados demonstrados por Courcoubetis e Weber (1995a) é possível assumir que:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \log Q(Nc, Nb, Nn) = \sup_t \inf_s \left[st \sum_{j=1}^J n_j \alpha_j(s, t) - s(b + ct) \right] = -I \quad (3.11)$$

sendo que I é conhecido como função taxa assintótica e a equação (3.11) é chamada de aproximação assintótica por muitas fontes.

Uma aproximação assintótica similar pode ser definida com relação a proporção de pacotes perdidos devido ao transbordo de um *buffer* finito de tamanho Nb . Utilizando o teorema de Chernoff, a última equação pode ser escrita como (WEISS, 1995):

$$P(\text{transbordo}) = e^{-NI + o(N)} \quad (3.12)$$

onde $o(N)$ denota um termo que quando dividido por N tende a zero conforme $N \rightarrow \infty$ (KUMAR; MANJUNATH; KURI, 2004). Para N elevado, a equação (3.12) pode ser aproximada por:

$$P(\text{transbordo}) \approx e^{-NI} \quad (3.13)$$

Utilizando Bahadur-Rao, um refinamento do clássico teorema de Cramer, é possível fornecer uma aproximação menos conservativa (MONTGOMERY; VECIANA, 1996):

$$P(\text{transbordo}) \approx e^{-NI - \frac{1}{2} \log(4\pi NI)} \quad (3.14)$$

Seja N , o número de fontes multiplexadas em um servidor com um *buffer* de tamanho B e ρ_j a porcentagem de fluxos do tipo j , então os parâmetros s, t podem ser calculados da seguinte maneira:

$$NI = \inf_t \sup_s F(s, t) \quad (3.15)$$

sendo

$$F(s, t) = s(B + Ct) - stN \sum_j \rho_j \alpha_j(s, t)$$

sendo que α_j é calculado empiricamente, utilizando o estimador em bloco (3.6) e a

solução da equação (3.15) envolve dois procedimentos de otimização. O primeiro consiste em procurar, para um valor fixo de t , o máximo $F^*(t) = \max_s F(s, t)$ e o segundo em achar o mínimo $NI = \min_t F^*(t)$.

A maximização pode ser resolvida numericamente de uma maneira eficiente se for levado em conta que o logaritmo da função geradora de momentos ($\log \mathbb{E}(e^{sX_j[0,t]})$) que é igual a $st\alpha_j(s, t)$ é convexo em s , enquanto que $s(B + Ct)$ é linear. Devido a isso, $F_t(s) = F(s, t)$ é uma função unimodal de s e o ponto de máximo é único (COURCOUBETIS; SIRIS, 2002). Então, para encontrar $F^*(t) = \max_s F_t(s)$ deve-se definir um incerto intervalo inicial $[s_a, s_b]$ que contenha o máximo (para isso é suficiente que para algum $x \in [s_a, s_b]$, $F_t(x) > F_t(s_a)$ e $F_t(x) > F_t(s_b)$) e diminuir sucessivamente esse intervalo utilizando o método de busca por seção áurea, descrito a seguir e ilustrado na Figura 5.

1. Dado um intervalo $[s_a, s_b]$, que contém o máximo, dois pontos s_e e s_d são selecionados de modo que: $s_d - s_a = s_b - s_e = a(s_b - s_a)$ onde a é a chamada proporção áurea, que é igual a $\frac{\sqrt{5}-1}{2} \approx 0.618$.
2. $F_t(s_e)$ e $F_t(s_d)$ são calculados:
 - (a) se $F_t(s_e) > F_t(s_d)$ o intervalo passa a ser $[s_a, s_d]$
 - (b) senão, se $F_t(s_e) < F_t(s_d)$ o intervalo passa a ser $[s_e, s_b]$
 - (c) senão, se $F_t(s_e) = F_t(s_d)$ o novo intervalo é $[s_e, s_d]$.
3. Os passos 1 e 2 são repetidos até que o intervalo seja suficientemente pequeno.

Diferentemente do procedimento de maximização $\max_s F_t(s)$, não existe uma propriedade de $F^*(t)$ que se possa tirar vantagem para encontrar o valor que minimiza $\min_t F^*(t)$. Na verdade, $F^*(t)$ pode ter mais de um mínimo e por essa razão a busca pelo valor mínimo é linear no intervalo $[0, \kappa\tau]$. A granularidade da busca é igual ao período de tempo de análise dos dados τ . Já o valor de κ é determinado empiricamente e depende do tamanho do *buffer*, com efeito, quanto maior for o tamanho do *buffer* maior é o valor de κ .

Esse estimador de banda efetiva também faz uso da Teoria dos Grandes Desvios, utilizando a teoria assintótica de muitas fontes ao invés de considerar o regime assintótico de buffers grandes, obtendo assim, uma estimativa menos conservadora (banda menor que atende à probabilidade de transbordo desejada). A aproximação assintótica de muitas fontes pode ser aplicada igualmente em tráfegos com dependências de curta ou longa

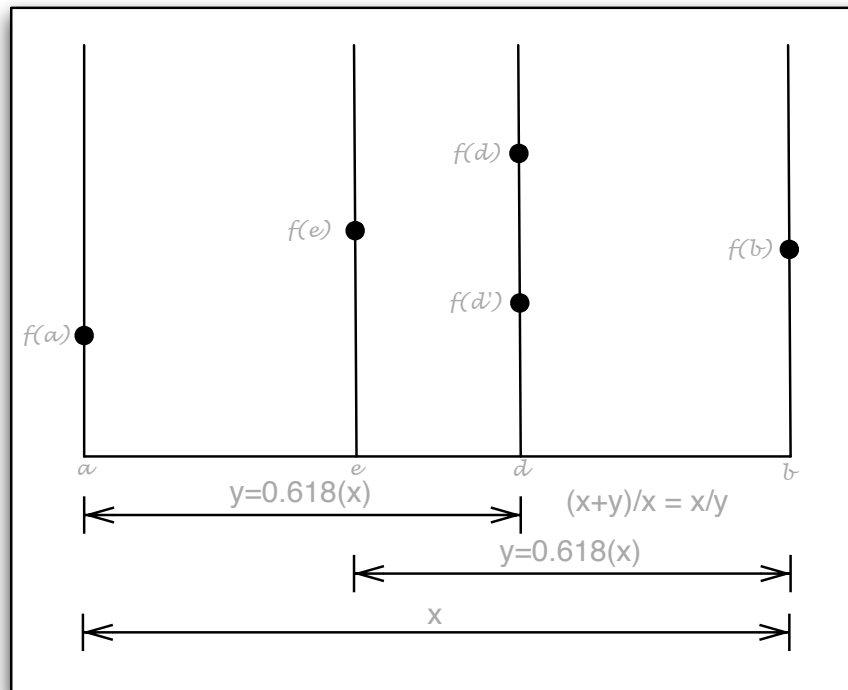


Figura 5: Método de Busca por Seção Áurea

duração (COURCOUBETIS; SIRIS; STAMOULIS, 1999; GROSSGLAUSER; BOLOT, 1996). O problema desse estimador reside na sua complexidade computacional, que o torna inviável para estimações em tempo real.

3.4.5 Aproximação Gaussiana

Esse método assume que a distribuição de chegada é Gaussiana e possui a menor complexidade computacional entre os algoritmos analisados. Porém ele frequentemente superestima a banda requerida, porque não leva em consideração a existência de um *buffer* no enlace. Além disso para assumir que a distribuição é Gaussiana é necessário que o efeito da multiplexação estatística seja relevante, ou seja, o método é inadequado para os casos em que uma ou poucas fontes estão transmitindo. O valor aproximado da taxa mínima de serviço (C) tal que a probabilidade de que a taxa de chegada seja maior que a taxa de serviço seja menor ou igual a ϵ pode ser estimado por:

$$C = \mu + \sigma \sqrt{-2\ln(\epsilon) - \ln(2\pi)} \quad (3.16)$$

sendo que μ é a taxa média e σ é o desvio padrão da taxa. Normalmente esse

estimador fornece um limitante superior para o valor da banda efetiva, por isso, é comum ele ser combinado com outros estimadores a fim de fornecer um valor mais próximo da capacidade efetiva (GUERIN; AHMADI; NAGHSHINEH, 1991).

4 Controle de Admissão para Servidores de Vídeo sob Demanda

4.1 Introdução

Para suprir as necessidades das aplicações de vídeo sob demanda, foi desenvolvido um mecanismo preventivo de controle de congestionamento no nível de conexão, que visa garantir uma alta taxa de utilização dos recursos do servidor e níveis constantes de atraso. O sistema de admissão de conexão utiliza a teoria assintótica de muitas fontes para calcular a banda efetiva de um agregado de fluxos de vídeo e uma base de dados para contabilizar as conexões ativas e os recursos disponíveis. Com o auxílio de um mecanismo de controle de tráfego, o sistema garante a reserva de uma fração mínima da capacidade do servidor, de acordo com os requisitos de *QoS* desejados (STAHLSCHMIDT; GODOY JR.; FORONDA, 2010).

A maioria dos modelos de admissão de conexão para servidores de vídeo sob demanda existentes na literatura, utilizam medições periódicas ou modelos de tráfego (KROPOTOFF, 1999). O problema com as abordagens baseadas em medições reside principalmente na dificuldade da definição de um intervalo de tempo para efetuar as medições, com efeito, se for muito pequeno consome muitos recursos e nem sempre é possível fornecer a resposta no tempo requerido, se for grande demais, é possível que características estatísticas relevantes do tráfego não sejam capturadas. Além disso, fluxos de vídeo apresentam uma estrutura bastante complexa e não existe um consenso sobre um modelo ideal para caracterizar esse tipo de tráfego.

Quando o conteúdo de um vídeo *VBR* é conhecido a priori, pode-se reorganizar o envio dos dados para suavizar a transmissão. Ao antecipar o envio dos maiores *frames* é possível evitar rajadas e reduzir a variabilidade do tráfego, facilitando o gerenciamento de recursos e o processo de admissão. Esforços significativos foram realizados nesse sentido (FENG; SECHREST, 1995; DUFFIELD; RAMAKRISHNAN; REIBMAN, 1998). Porém nesses trabalhos, os algoritmos dependem do programa cliente, que reproduz o vídeo, limitando

assim seu escopo de aplicação. Mesmo que a suavização seja ótima, o tráfego resultante ainda é *VBR*, ou seja, ainda há possibilidade de ganho com a multiplexação estatística (SALEHI et al., 1998).

O sistema proposto nessa dissertação utiliza arquivos de *trace*, que são automaticamente gerados de modo a simular o envio dos vídeos sob uma rede de comutação de pacotes. A análise baseada em *traces* permite a caracterização total dos processos e permite uma estimativa mais precisa da banda efetiva. Porém como o sistema não captura o comportamento dinâmico dos fluxos no tempo e é muito difícil prever a quantidade demandada de cada vídeo em um dado instante, a multiplexação estatística é considerada somente entre os fluxos idênticos (multiplexação por grupo).

Além de viabilizar a predição de tráfego, a multiplexação por grupo permite associar diferentes níveis de serviço a vídeos distintos. A perda representada pela ausência da multiplexação entre grupos distintos deve ser menor nos servidores de vídeo sob demanda, pois, geralmente, nesse tipo de aplicação alguns poucos vídeos respondem por grande parte do consumo dos recursos do enlace. Em outras palavras, o ganho estatístico resultante da multiplexação entre os fluxos idênticos dos vídeos mais requisitados deve ser relevante em relação ao montante total dos recursos consumidos.

4.2 Arquitetura

No instante em que um novo vídeo é adicionado ao acervo de vídeos do servidor, seu *trace* é criado de modo a simular o transporte do seu conteúdo sobre uma rede de comutação de pacotes. Após a criação, o *trace* é enviado para um servidor de cálculo que, por sua vez, calcula o valor da banda efetiva, que garante um nível constante de atraso, para um certo número de conexões iniciais. Terminado os cálculos, os resultados são inseridos em uma base de dados que provê os subsídios necessários para o controle de admissão.

À medida que o número de conexões que transmitem um mesmo vídeo simultaneamente se aproxima do número máximo de conexões, cuja a banda efetiva foi inicialmente calculada, um módulo de estatísticas de conexões se encarrega de solicitar novos valores de banda efetiva ao servidor de cálculo. Para introduzir a qualidade de serviço, além do controle de admissão, é empregada a técnica de controle de tráfego, que visa assegurar uma fração mínima da capacidade (banda efetiva) à cada agregado de fontes que transmitem o mesmo conteúdo.

A Figura 6 ilustra a arquitetura do sistema. O servidor de controle de admissão é o ponto de entrada do sistema, onde uma interface web permite aos clientes selecionar os vídeos que desejam assistir. Os servidores se comunicam entre si utilizando um protocolo de transferência de arquivos, uma API de acesso a base de dados e chamadas remotas de procedimento. O objetivo da arquitetura é garantir um atraso máximo e uma alta utilização do enlace entre o servidor de admissão e o roteador.

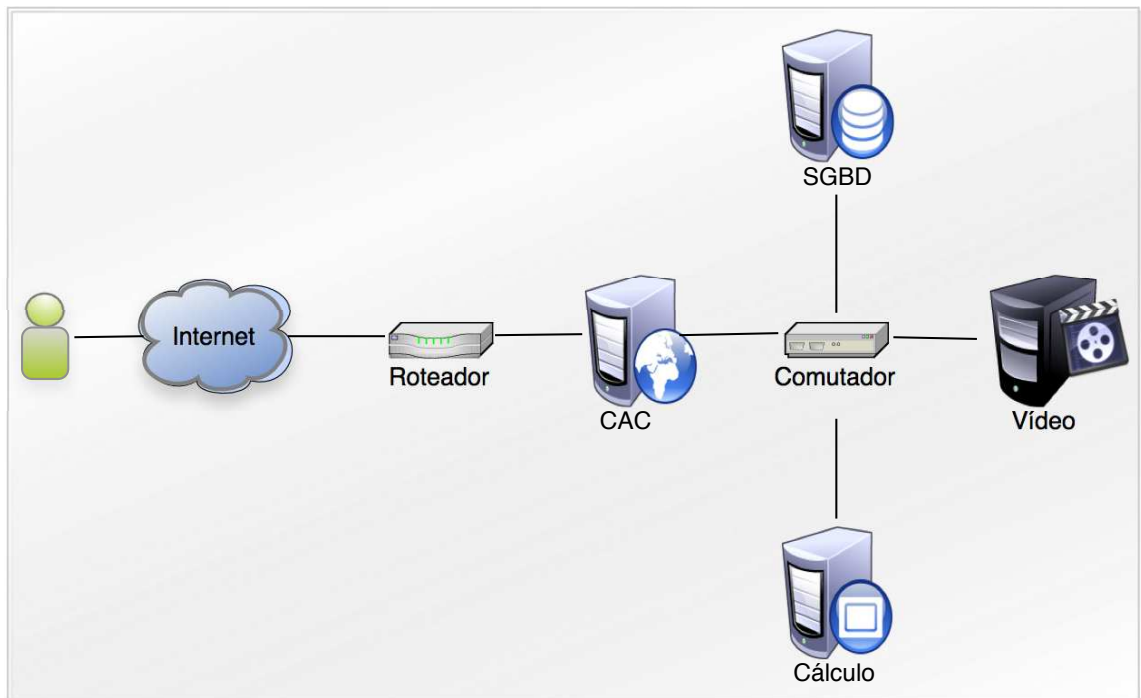


Figura 6: Arquitetura do sistema de Controle de Admissão de Conexão

4.3 Processo de Chegada de Dados

O processo de chegada de dados (*streams*) no enlace que conecta o servidor de controle de admissão à Internet pode ser modelado utilizando uma única fila, onde os pacotes são desenfileirados a uma taxa constante (C), na ordem de chegada (*FIFO*) e a capacidade do *buffer* é dimensionada para que a perda de pacotes seja mínima, ou seja, a probabilidade de um pacote chegar e não haver um lugar livre no *buffer* é suficientemente pequena.

Para simplificar, o modelo descrito anteriormente e representado na Figura 7 é adotado daqui pra frente. Isso significa que a causa da perda de um pacote é a inexistência de posições livres na fila e o atraso é considerado o tempo que um pacote permanece na

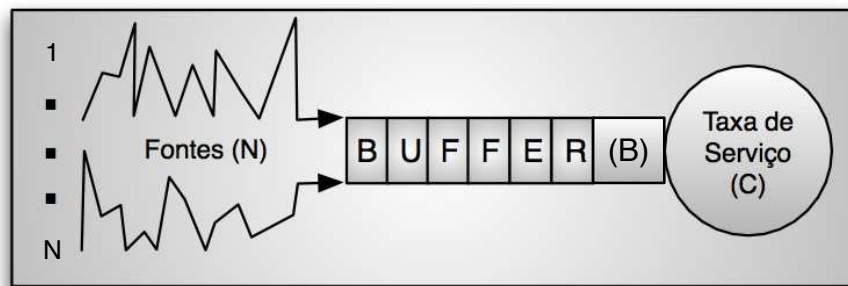


Figura 7: Processo de chegada de dados

fila antes de ser removido. Nesse modelo, para garantir um atraso máximo, basta garantir que todo elemento que entra na fila seja desenfileirado respeitando um tempo máximo. Esse tempo máximo é calculado utilizando a capacidade do enlace e o tamanho máximo atingido pela fila em um determinado instante. Assim, se o tamanho da fila crescer além do tamanho do *buffer* haverá violação das restrições de atraso. Portanto para satisfazer os requisitos de atraso é suficiente analisar a probabilidade do tamanho da fila exceder o tamanho do *buffer*. Dessa forma é possível uma certa previsibilidade, com garantia de níveis de atraso constantes.

4.4 Construção e Conversão do Trace de Vídeo

A estimação da banda efetiva de um fluxo de vídeo é feita através de um arquivo de texto que detalha o processo de chegada de dados no tempo (*trace*). Para automatizar a geração do *trace*, um conhecido player multimídia de código aberto chamado *MPlayer* (GEREOFFY et al., 2007) foi alterado de modo a detalhar as informações sobre o tamanho de cada *frame* do vídeo.

Em um vídeo cada imagem individual é chamada de *frame*, e a medição da cadência (*frame rate*) é feita em *frames* por segundo (*fps*). O tamanho dos arquivos de *trace* dependem da duração, da resolução e da cadência dos vídeos. Na análise da banda efetiva de um vídeo o período de análise coincide com a sua cadência de modo a capturar suas propriedades estatísticas relevantes e fornecer uma estimativa precisa de seus requisitos de serviço.

Inicialmente, a geração automática de *trace* foi desenvolvida para os *streams* de vídeo encapsulados no formato Audio Video Interleave, um formato de arquivo recipiente extremamente popular, desenvolvido pela Microsoft® cujo a extensão oficial é *AVI*. A construção do *trace* é feita através da extração de dados de um cabeçalho opcional

presente nos arquivos recipientes chamado *IDX1*. Se o cabeçalho não estiver presente o *MPlayer* pode facilmente reconstruí-lo, além disso vários outros formatos de arquivos recipientes podem ser convertidos em *AVI* tornando possível a construção do *trace* para outros formatos de arquivos.

Para simular a passagem de um vídeo por uma rede de comutação de pacotes, de modo a calcular com precisão a sua banda efetiva, foi utilizado o software *Encapsulate*, que converte o *trace* de vídeo em *trace* de rede (MCALARNEY; HADDAD; MCGARRY, 2009). O software é capaz de empacotar e fragmentar os *frames* de vídeo de acordo com os protocolos de rede e o *Maximum Transmission Unit (MTU)* utilizados.

Neste trabalho é utilizado o protocolo *RTP* e um *MTU* de 1400 bytes para a entrega dos dados de vídeo ponto-a-ponto. A Tabela 1 mostra a transformação de parte de um *trace* de vídeo em um *trace* de rede. Os cabeçalhos introduzidos para o transporte do vídeo utilizando os protocolos *IP*, *UDP* e *RTP* somam 40 bytes por pacote.

Tabela 1: Conversão do *trace* de vídeo em *trace* de rede

Frame (Bytes)	Pacote(s) (Bytes)
2102	2182
824	864
846	886
1578	1658
768	808

4.5 Seleção do Estimador Adequado

Dentre os estimadores de banda efetiva apresentados, os mais promissores para a estimação da banda efetiva para servidores de vídeo sob demanda são o de Norros (*NEB*), a Aproximação Gaussiana (*GA*) e a Aproximação Assintótica por Muitas Fontes (*MSAEB*). A Tabela 2 compara o desempenho dos estimadores selecionados em diferentes cenários.

Para as aplicações de vídeo sob demanda, além da vazão, um dos parâmetros de qualidade mais importantes é o atraso. Para que o atraso seja pequeno, muitas vezes o *buffer* ajustado (de acordo com a taxa de serviço) também deve ser pequeno e por isso o estimador *NEB*, que faz suposições sobre o tamanho do *buffer*, não é eficaz nesse contexto. O estimador baseado na Aproximação Gaussiana é inadequado para processos com decaimentos lentos (dependência de longa duração), comportamento observado nos

Tabela 2: Comportamento dos estimadores selecionados

Cenário	<i>NEB</i>	<i>MSAEB</i>	<i>GA</i>
Buffer Pequeno	Ruim	Muito Bom	Ótimo
Buffer Grande	Muito Bom	Muito Bom	Ruim
Tráfego <i>SRD</i>	Bom	Bom	Bom
Tráfego <i>LRD</i>	Ótimo	Bom	Ruim
Fonte Única	Bom	Ruim	Ruim
Várias Fontes	Muito Bom	Ótimo	Ótimo
Complexidade	Bom	Ruim	Ótimo

Fonte: Adaptado de Haciomeroglu (2003).

fluxos de tráfego de vídeo, e além disso, freqüentemente superestima o valor da banda efetiva. Como a banda efetiva é pré-calculada (off-line), a complexidade computacional do algoritmo não é um fator determinante na escolha do estimador. Portanto, entre os algoritmos estudados, o mais adequado as necessidades do sistema proposto é o *MSAEB*.

4.6 Cálculo do Número Máximo de Conexões Admitidas

Para garantir um atraso máximo, o tamanho do *buffer* deve ser ajustado de acordo com a taxa de serviço do enlace, de modo que o tempo máximo que um pacote permanece na fila antes de ser desenfileirado não exceda o atraso desejado. Para calcular o tamanho do *buffer* em bytes basta multiplicar a capacidade do enlace em bytes por segundo pelo atraso desejado em segundos.

A utilização da aproximação assintótica por muitas fontes permite calcular diretamente o número de fontes que podem transmitir em um enlace, fornecendo os dados necessários para o controle de admissão. A equação (4.1) estima o número máximo de conexões que podem ser aceitas (N), dada uma capacidade (C) e um *buffer* previamente ajustado (B), com uma probabilidade de exceder o atraso desejado de aproximadamente $e^{-\gamma}$.

$$N = \inf_t \sup_s \left[\frac{s(B + Ct) - \gamma}{st\alpha(s, t)} \right] \quad (4.1)$$

O cálculo de $\alpha(s, t)$ é feito utilizando o *trace* de rede e o estimador em bloco (equação (3.6)). Assim como na equação (3.15), também é possível tirar proveito da convexidade em s e resolver \sup_s utilizando uma busca por seção áurea. Utilizando o

teorema de Bahadur-Rao, pode-se aumentar a precisão da equação (4.1), sem aumentar a complexidade computacional, substituindo γ por $\gamma - \frac{1/2 \log(4\pi\gamma)}{1+1/(2\gamma)}$ (MONTGOMERY; VECIANA, 1996).

A computação do número máximo de conexões admitidas é executada repetidas vezes para vários pares (C, B) , onde o tamanho do *buffer* (B) em bytes varia de acordo com a capacidade alocada (C) mas é fixo em função do tempo. A taxa de serviço é incrementada sucessivamente até a obtenção das capacidades mínimas necessárias para suprir a demanda inicial do vídeo, isto é, o número de conexões que transmitem o vídeo simultaneamente.

A Tabela 3 ilustra os parâmetros utilizados no cálculo do número de conexões que podem ser admitidas, utilizando a equação (4.1) e o aperfeiçoamento de Bahadur-Rao, para que a probabilidade de que o tamanho da fila exceda o tamanho do *buffer* seja aproximadamente 10^{-6} . O tamanho do *buffer* foi ajustado, de acordo com cada capacidade, em 40 milisegundos. O cálculo foi realizado utilizando o *trace* de um vídeo com cadência de 25 *fps* em formato *MPEG-4*, contendo um fragmento do filme "O Silêncio dos Inocentes" (FITZEK; REISSLEIN, 2000), que foi encapsulado e segmentado em pacotes de 1400 bytes de modo a simular sua passagem por uma rede *IP* utilizando o protocolo *RTP*.

Tabela 3: Cálculo do número máximo de fluxos em função da *QoS* desejada

Capacidade (Mbps)	Fluxos (N)	s ($kbit^{-1}$)	t (ms)	$\alpha(s, t)$ (Mbps)
10	21	0.001243	2160	0.2785
20	61	0.000926	2160	0.2383
30	105	0.000512	3360	0.2250
40	151	0.000455	3360	0.2184
50	198	0.000379	3680	0.2140
100	440	0.000217	4840	0.2043
1000	5103	0.000035	11080	0.1909

Ao observar o comportamento dos parâmetros de espaço (s) e tempo (t) na Tabela 3, nota-se que a medida que a capacidade cresce, a multiplexação estatística aumenta e o valor do parâmetro espaço decresce. Com o crescimento da capacidade, cresce também o tamanho do *buffer* ajustado, e com isso o *buffer* passa a "absorver" as menores escalas de tempo responsáveis pelo transbordo, o que implica no crescimento do valor do parâmetro tempo. Conforme s decresce e t cresce o valor de $\alpha(s, t)$ se aproxima da taxa média do *trace* de rede (0.1853 Mbps).

4.7 Controle de Admissão Auxiliado por SGBD

O processo de admissão de conexão é auxiliado por um SGBD, onde um banco de dados armazena as informações sobre os recursos necessários (taxa de serviço e *buffer*) para admitir um determinado número de fontes de cada vídeo armazenado no servidor. O banco contém ainda, relações que mantêm informações sobre os vídeos armazenados e os recursos do enlace, além de uma tabela específica, que é responsável por contabilizar as conexões ativas, detalhando quantas fontes estão transmitindo cada vídeo.

Para facilitar o processo de admissão, visões informam o total de recursos disponíveis e utilizados. O ganho decorrente da utilização do sistema proposto quando comparado a alocação determinística pela taxa de pico pode ser obtido em tempo real, através de uma simples consulta a uma visão que detalha a economia de recursos. O modelo do banco de dados que fornece os subsídios para o controle de admissão está representado na Figura 8 .

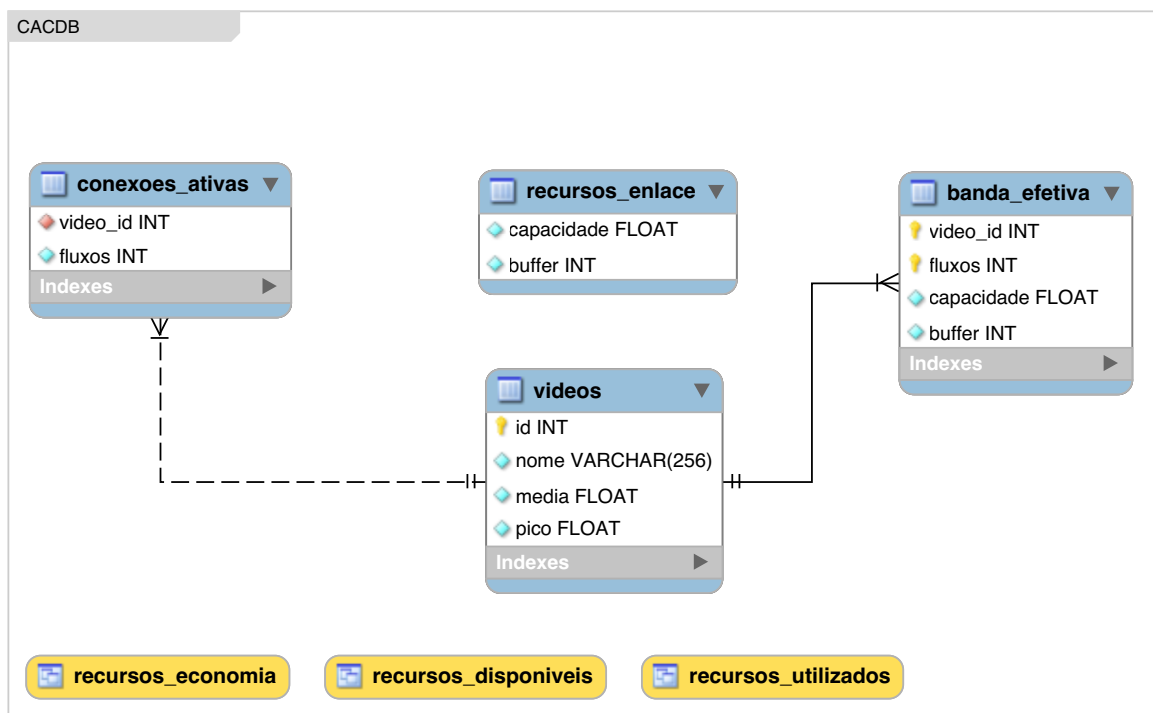


Figura 8: Diagrama ER do controle de admissão

Quando um cliente faz uma requisição de vídeo, o servidor de admissão invoca um procedimento armazenado no servidor de base de dados usando o nome do vídeo solicitado como parâmetro. O procedimento de admissão, verifica quantas fontes estão transmitindo o vídeo requisitado e calcula, através da tabela de banda efetiva, quanto recurso adicional é necessário se mais uma fonte vier a transmitir o mesmo vídeo. Se há

disponibilidade de recursos, a conexão é aceita e a tabela de conexões ativas atualizada. Sempre que uma transmissão é iniciada ou encerrada, um procedimento é executado para atualizar o número de conexões ativas.

O Apêndice B contém o código necessário para a construção da base de dados de admissão, com as tabelas, visões e funções descritas nessa seção. O código é compatível com o MySQL, um popular sistema gestor de base de dados de código aberto (WIDENIUS; AXMARK; MYSQL, 2002).

4.8 Controle da Demanda

Em um determinado momento, o número de fluxos ativos excederá o número de conexões para as quais a banda efetiva foi pré-calculada. Para evitar esse problema, sempre que uma conexão é admitida, a relação que contém os dados das conexões ativas é atualizada e um gatilho (*trigger*) associado a essa atualização dispara um procedimento. Esse procedimento verifica o quão próximo o número de conexões ativas do vídeo requisitado está do número máximo de conexões, cuja banda efetiva foi previamente calculada.

Para que uma solicitação de um cliente não tenha que esperar o tempo necessário para o cálculo da banda efetiva, sempre que o procedimento verificar que o número de conexões ativas de um vídeo se aproxima do teto contido na base de dados, novos valores de banda efetiva são solicitados ao servidor de cálculo com o objetivo de suprir a demanda futura.

Caso a demanda cresça de maneira imprevisível, pode-se admitir a conexão utilizando um cálculo conservativo, com base na diferença entre as duas maiores capacidades previamente calculadas. Essa admissão pode ser realizada porque conforme a capacidade cresce a banda efetiva dos fluxos diminui, como pode ser visto na Tabela 3.

O cálculo antecipado é necessário porque a computação da banda efetiva sobre o regime assintótico de muitas fontes não fornece uma resposta em tempo real e depende do tamanho do vídeo e dos valores do parâmetro de tempo pesquisados.

4.9 Classificação e Controle de Tráfego

Uma vez aceita pelo servidor de admissão, a conexão é classificada dentro de uma classe apropriada, de acordo com o conteúdo que está transmitindo. Na seqüência, um

mecanismo de controle de tráfego, garante a cada grupo de vídeos (classe) a capacidade mínima necessária para fornecer os serviços de acordo com as garantias desejadas.

O controle de tráfego permite utilizar o enlace físico para simular vários enlaces mais lentos e enviar diferentes tipos de tráfego em diferentes enlaces virtuais. Para implementar o controle de tráfego é utilizado o *Hierarchical Token Bucket (HTB)* (DEVERA, 2004), ele garante que o total de serviço provido para cada classe (enlace virtual onde todas as fontes transmitem o mesmo conteúdo) é no mínimo o total requerido por esta. Quando uma classe requisita menos que o total a ela reservado, a largura de banda remanescente é distribuída para outras classes que demandam serviço.

O *HTB* controla os pacotes que saem de uma interface de rede de forma hierárquica. A classe raiz é definida através de dois parâmetros, taxa e teto, esses valores representam a banda total disponível no enlace. O parâmetro taxa indica a banda garantida para a classe e o teto indica o máximo de banda que a classe pode consumir.

O enlace físico é dividido com base no número de vídeos que estão sendo transmitidos e as suas bandas efetivas. Cada grupo de fluxos idênticos é transmitido através de um enlace virtual (classe filha) e tem garantias de receber, no mínimo, a banda necessária para garantir a *QoS* desejada. O parâmetro teto é o mesmo para todas as classes filhas e é igual ao total de banda disponível no enlace. Os detalhes da implementação constam no Apêndice C. O esquema é ilustrado na Figura 9.

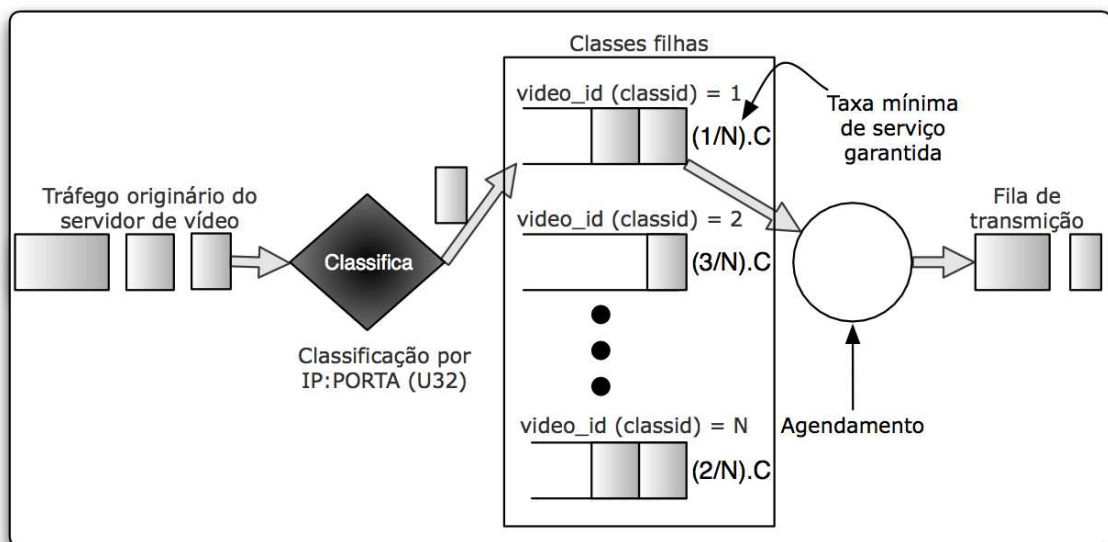


Figura 9: Esquema de controle de tráfego utilizando *HTB*

Sob o regime assintótico de muitas fontes, o ganho alcançado com a multiplexação estatística depende do tamanho do sistema (número de fontes e recursos), com a divisão

do enlace e a divisão do sistema em subsistemas existe uma perda (ganho não explorado) que é analisada com mais detalhes na próxima seção.

4.10 Estratégia de Multiplexação por Grupos

Quando um pequeno número de vídeos é responsável por uma porção significativa do tráfego ou o número de fontes de cada vídeo é suficientemente grande, a perda representada pela multiplexação em grupos é mínima. Mas quando há pouca demanda e principalmente quando essa demanda é igualmente distribuída entre os vídeos a economia de recursos é pequena.

Freqüentemente o comportamento da demanda pelos vídeos em um servidor de vídeo sob demanda é semelhante ao de uma vídeo locadora, onde os filmes mais recentes e as grandes produções respondem por grande parte das locações. Por esse motivo a multiplexação por grupos pode ser aplicada satisfatoriamente em muitos servidores de vídeo sob demanda.

A Tabela 4 detalha quantos fluxos do vídeo "Guerra nas Estrelas" (FITZEK; REIS-SLEIN, 2000), encapsulado utilizando *IP/UDP/RTP* e fragmentado em pacotes de 1400 bytes, podem ser admitidos de acordo com a capacidade do enlace. O cálculo foi feito utilizando a equação (4.1) com probabilidade de aproximadamente 10^{-6} de que o atraso não exceda 40 milissegundos .

Tabela 4: Capacidade mínima vs número de fluxos admitidos

Capacidade (Mbps)	Fluxos (N)
1	3
2	12
4	31
8	73
16	158
32	334
64	693

Com o objetivo de ilustrar a perda introduzida pela adoção da multiplexação por grupos, dois arquivos contendo o mesmo fragmento do vídeo anterior são tratados como se fossem diferentes, ou seja, multiplexados em enlaces virtuais diferentes, a capacidade requerida e a perda percentual são ilustradas na Tabela 5.

Conforme o número de fontes multiplexadas aumenta, a banda efetiva tende a

Tabela 5: Desperdício de recursos com o modelo de multiplexação em grupo

Capacidade (Mbps)	Fluxos (N)	Perda (%)
2	6	50.00
4	24	22.58
8	62	15.07
16	146	7.60
32	316	5.39
64	668	3.61

taxa média e a perda introduzida pela análise em grupos diminui. Com isso é possível concluir que a arquitetura proposta é ideal para servidores com grande volume de tráfego e não serve para pequenos servidores de vídeo sob demanda.

4.11 Funcionamento do Sistema

Para ilustrar o funcionamento do sistema de admissão são utilizados diagramas de seqüência. Os diagramas de seqüência descrevem a maneira como os servidores interagem ao longo do tempo para a realização de uma operação. As linhas verticais representam a passagem do tempo e quando preenchidas indicam também a atividade do servidor. As linhas horizontais representam as trocas de informações e chamadas de procedimentos. Uma condição é representada através de colchetes e a resposta a uma ação é representada por linhas horizontais tracejadas.

A Figura 10 representa a seqüência de passos executados para a população da base de dados com as capacidades mínimas necessárias ao nível de serviço exigido para a transmissão dos fluxos de um vídeo. O processo começa com o *upload* do vídeo pelo usuário. Em seguida, o *trace* de vídeo é gerado e imediatamente convertido em *trace* de pacotes. Na seqüência as informações sobre o *stream* são registradas na base de dados. Então, a partir do *trace* de pacotes, são calculadas as bandas efetivas necessárias para admitir um número variável de conexões enquanto um atraso máximo é garantido. Por fim os valores calculados são inseridos na tabela apropriada com o objetivo de prover os dados necessários para o controle de admissão. O segmento de reta com meia seta na extremidade representa a solicitação assíncrona que ocorre de acordo com a demanda pelo vídeo e é responsável pelo cálculo antecipado das bandas efetivas.

Sempre que um cliente solicita um vídeo, o servidor de admissão invoca o procedimento de admissão no sistema gestor de base de dados. O procedimento implementado

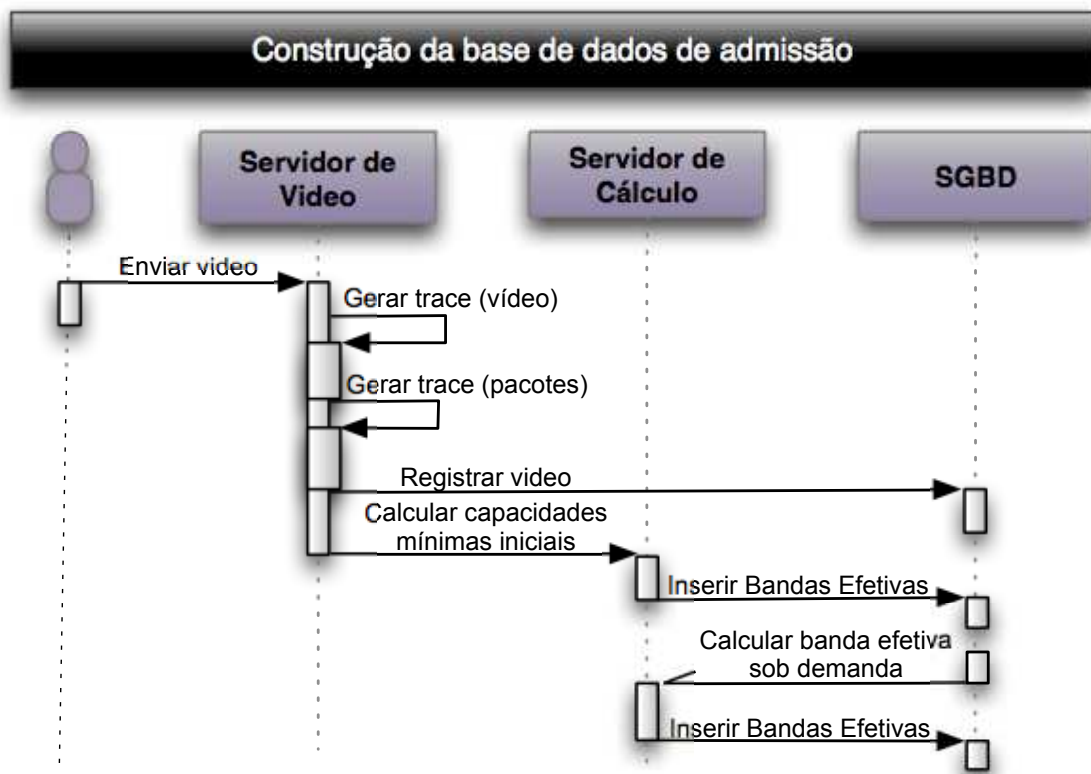


Figura 10: Construção da base de dados que auxilia o controle de admissão

no SGBD verifica se há recursos suficientes para admitir a conexão de acordo com a banda efetiva pré-calculada. Caso haja recursos suficientes, eles são alocados (a tabela de recursos é atualizada) e o servidor de admissão é informado. O servidor de admissão então ajusta a capacidade do enlace virtual (classe do vídeo) para suportar mais uma fonte de vídeo e cumprir os requisitos de QoS desejados. Após a capacidade ser reservada, o tráfego vindo do servidor de vídeo com direção a Internet é marcado e direcionado para o enlace virtual que contém outros fluxos idênticos. Esse processo é ilustrado na Figura 11.

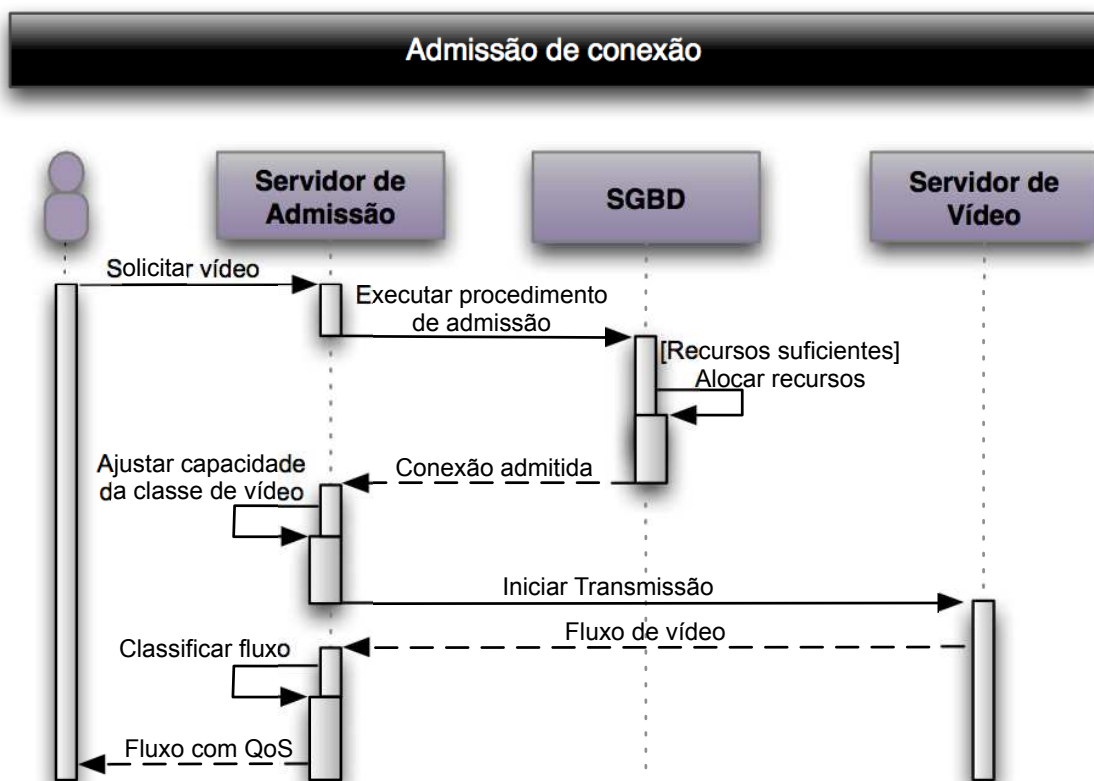


Figura 11: Processo de admissão de conexão

5 Simulações

5.1 Introdução

Com o objetivo de validar a aplicação do estimador selecionado no transporte de vídeos *MPEG-4* sobre redes *IP*, foi desenvolvido um módulo de software, utilizando o simulador de eventos discretos *NS-2* (MCCANNE et al., 1997), que permite verificar o comportamento da multiplexação estatística quando vários fluxos de vídeo com bitrate variável são transportados sobre o mesmo enlace.

O enlace que conecta o servidor de admissão à Internet (ver Figura 6) é o ponto onde os *streams* de vídeo são agregados. O objetivo dos experimentos é simular a passagem dos fluxos por esse enlace, conforme representado na Figura 12. O enlace foi modelado de acordo com o processo descrito na Seção 4.3, exceto pelo fato de que a capacidade do *buffer* é pré-definida e é tão grande que no contexto dos experimentos pode ser considerada infinita.

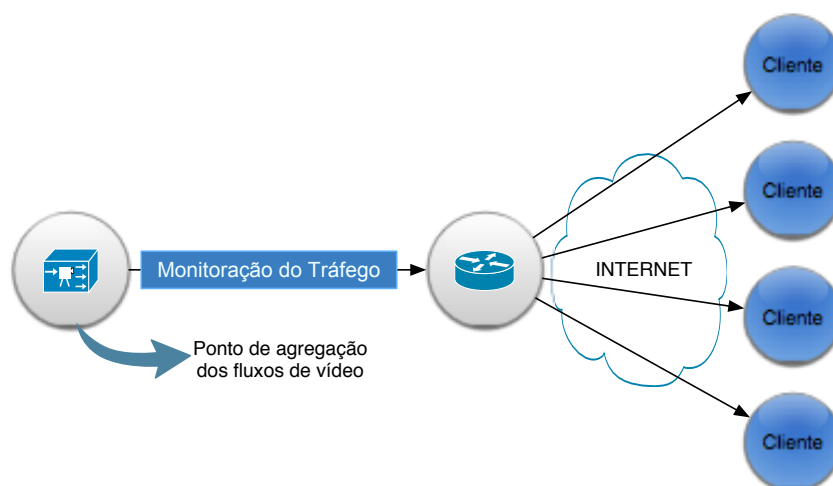


Figura 12: Enlace alvo dos experimentos

Em cada experimento, a alocação por banda efetiva é comparada a alocação determinística, onde é verificado que a teoria assintótica de muitas fontes permite uma

maior utilização dos recursos sem violar os requisitos de *QoS* desejados. O código do simulador está disponível no Apêndice B.

5.2 Cenário

O software de simulação recebe como parâmetros: um arquivo de *trace* de rede, uma taxa de serviço (C) e um número de fluxos (N). O arquivo de *trace* é lido para a memória principal e arranjado na forma de uma lista circular encadeada, onde cada *frame* ocupa uma posição, dessa forma, enquanto durar a simulação as fontes transmitirão dados. Para cada fluxo de vídeo, o primeiro *frame* a ser transmitido é selecionado de maneira aleatória e o restante é enviado de forma seqüencial, seguindo a seqüência da lista. Os tempos iniciais de transmissão são aleatórios, com distribuição uniforme de probabilidade sobre o tempo de duração de um frame.

Para simular os *streams*, um *trace* de rede foi gerado a partir de um segmento de aproximadamente 60 minutos do filme "O Silêncio dos Inocentes" . As informações sobre o vídeo, o *trace* e a codificação utilizada estão disponíveis publicamente em Fitzek e Reisslein (2000). Na conversão, os *frames* foram encapsulados utilizando os protocolos *IP* (Internet Protocol, para roteamento), *UDP* (User Datagram Protocol, para o transporte sem conexão) e *RTP* (Real-time Transport Protocol, para o controle de tempo-real) e segmentados em pacotes de 1400 bytes. A Figura 13 detalha o *trace* de rede, sua taxa de transmissão média é de 0.185 Mbps e a taxa de pico de 2.455 Mbps.

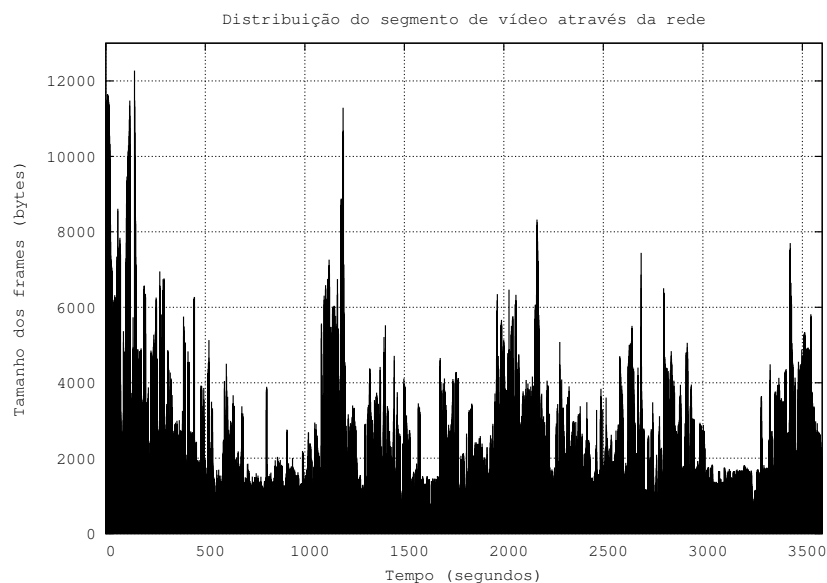


Figura 13: Trace de rede utilizado nos experimentos

As capacidades mínimas necessárias para se admitir até 100 fluxos do vídeo (N), com probabilidade de aproximadamente 10^{-6} de que o atraso exceda 20 milissegundos, foram calculadas utilizando a equação (4.1) e o teorema de Bahadur-Rao.

5.3 Ganho Obtido com a Multiplexação Estatística

O objetivo desse experimento é analisar a economia de recursos alcançada com a utilização do controle de admissão baseado na aproximação assintótica de muitas fontes. Na Figura 14, os valores das bandas efetivas calculadas, utilizando a equação (4.1) e o aperfeiçoamento de Bahadur-Rao, para admitir até 100 fontes com um *buffer* de 20 milissegundos são comparados com as alocações determinísticas por taxa média e por taxa de pico.

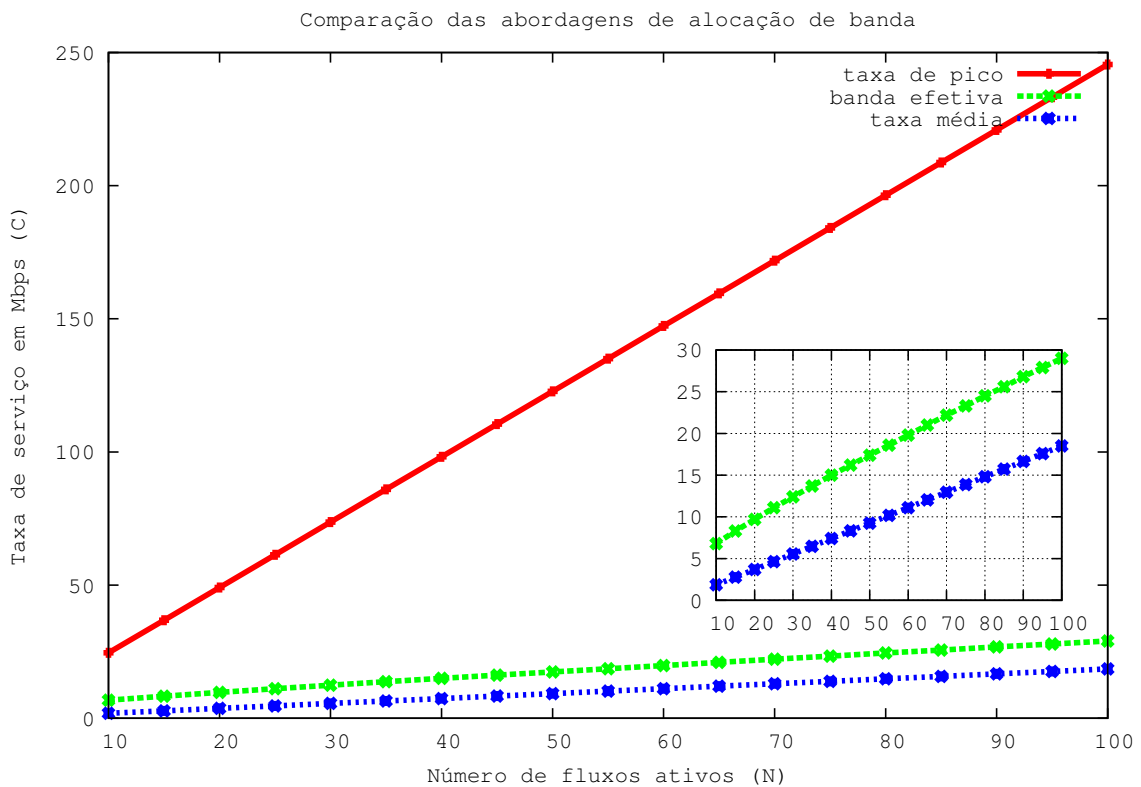


Figura 14: Comparação das alocações por taxa de pico, banda efetiva e taxa média

Na alocação por taxa de pico há um grande desperdício de recursos, enquanto que na alocação por taxa média não há desperdício, mas a qualidade do serviço é degradada. Como pode ser visto nos experimentos seguintes, a alocação por banda efetiva garante os parâmetros requeridos de QoS e uma alta taxa de utilização dos recursos.

A alocação por banda efetiva acompanha a forma da alocação por taxa média.

Pela teoria assintótica de muitas fontes, para um número de fluxos suficientemente grande, conforme o número de fontes de vídeo cresce, a taxa agregada se aproxima da taxa média multiplicada pelo número de fontes.

5.4 Verificação da Qualidade de Serviço

A finalidade dessa simulação é verificar se as restrições de *QoS* impostas no cálculo da banda efetiva estão sendo cumpridas. Ao todo foram executadas 10 simulações, cada uma durante um intervalo de tempo de 200 minutos (5 vezes o tamanho do *trace*) com coeficiente de variação inferior a 5%. Ao fim de cada simulação o tamanho máximo que o *buffer* atingiu durante o experimento é convertido em uma unidade de medida de tempo de acordo com a capacidade do enlace. O resultado das alocações por banda efetiva e taxa de pico podem ser vistos na Figura 15.

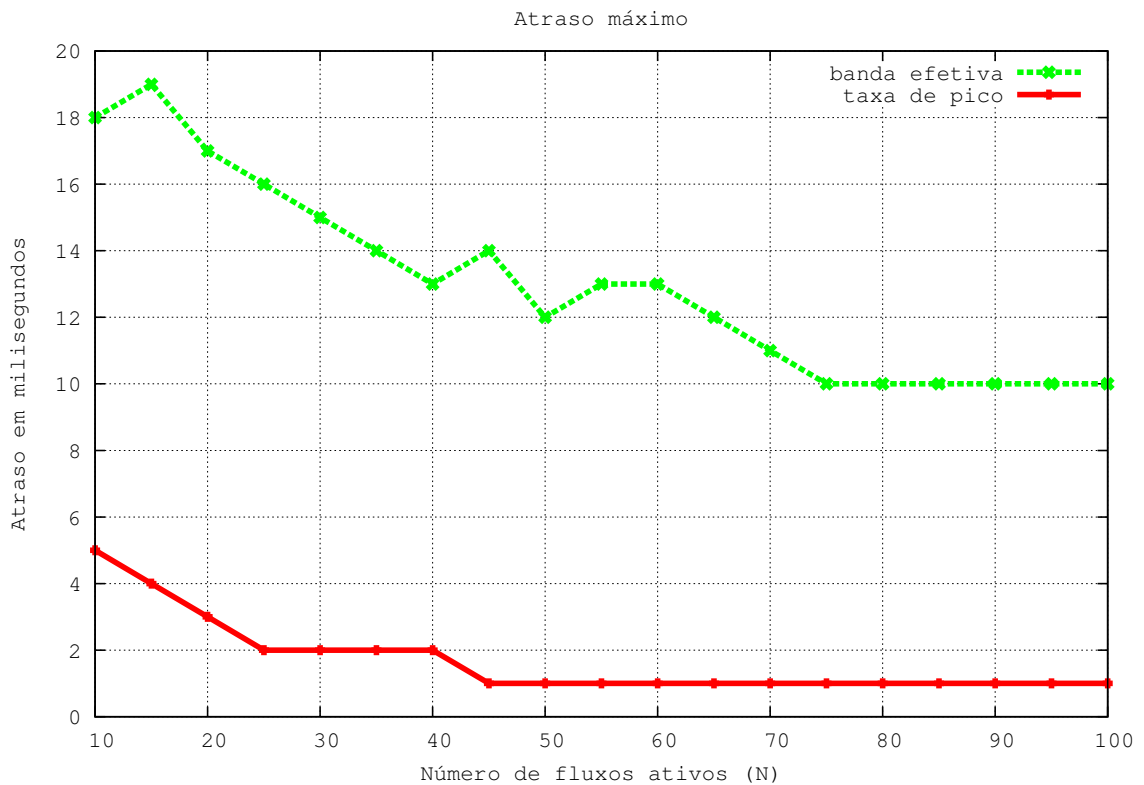


Figura 15: Atraso máximo das alocações por taxa de pico e banda efetiva

É possível verificar que o atraso decresce conforme o número de fontes aumenta. Na verdade o atraso decresce porque com o crescimento do número de fontes aumentam também a taxa de serviço e o *buffer* correspondente. Os maiores valores do atraso, observados no canto superior esquerdo do gráfico, se devem ao fato de que o tamanho dos

maiores *frames* do vídeo são pouco menores que o tamanho em bytes dos buffers ajustados para as menores capacidades alocadas.

Quando se utiliza a alocação por taxa média, as restrições não só são violadas, como os atrasos chegam a ser até quatro ordens de grandeza maiores que os atingidos quando se emprega as abordagens anteriores. A Figura 16 ilustra o atraso máximo quando se utiliza a taxa média como base para a alocação de recursos.

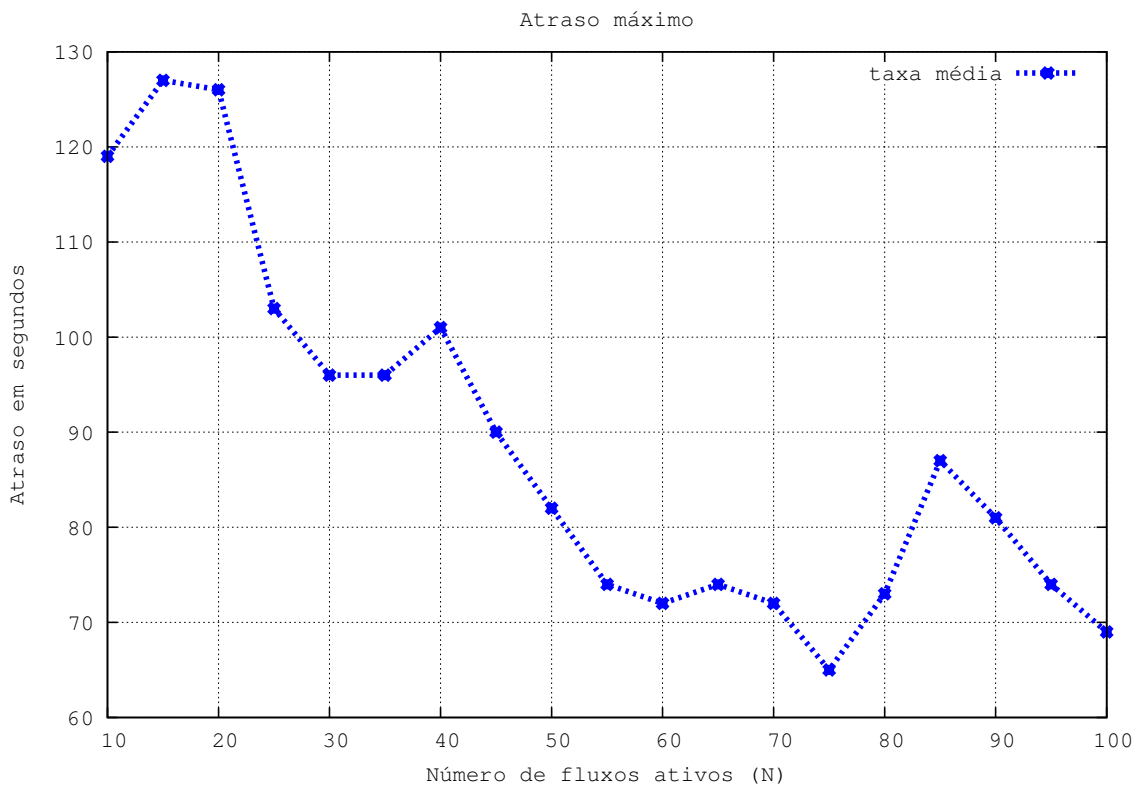


Figura 16: Atraso máximo da alocação por taxa média

5.5 Taxa de Utilização do Enlace

Com o objetivo de analisar a taxa de utilização do canal quando se emprega o conceito de banda efetiva na estimação da capacidade, as taxas de utilização alcançadas pela alocação estatística baseada na aproximação assintótica por muitas fontes e pela alocação determinística utilizando a taxa média e a taxa de pico foram comparadas. A taxa de utilização do enlace é calculada no fim de cada simulação como se segue:

$$utilização = \frac{total\ de\ bytes\ desenfileirados\ em\ i\ segs}{capacidade \times i}$$

O numerador é igual ao número total de bytes que são desenfileirados do *buffer* durante o intervalo de i segundos e o denominador é igual a taxa total de bytes que poderiam ter sido desenfileirados durante o mesmo intervalo.

Na Figura 17 é possível verificar que a alocação baseada na taxa de pico, apesar de garantir a qualidade de serviço, causa um grande desperdício de recursos, enquanto que na alocação por taxa média os recursos são utilizados de forma ótima, mas os atrasos gerados são inaceitáveis.

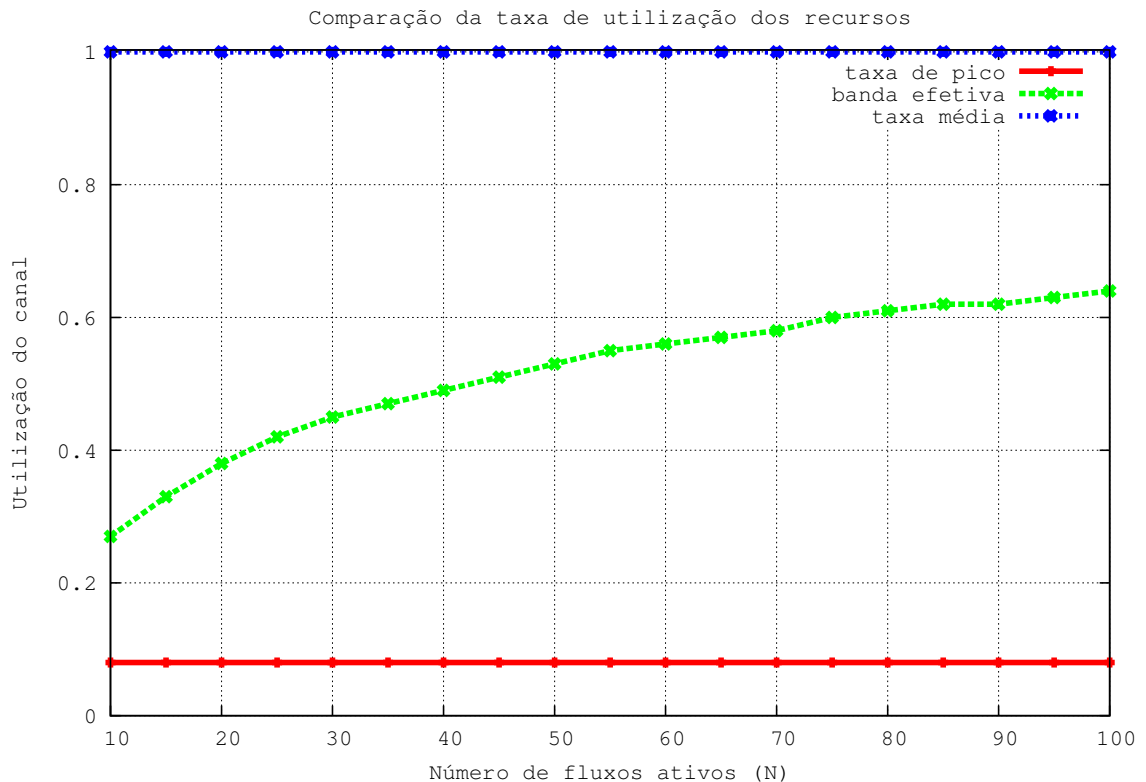


Figura 17: Taxa de utilização das abordagens analisadas

Na abordagem por banda efetiva, conforme o número de fontes e a taxa de serviço cresce a taxa de utilização dos recursos aumenta. A medida em que a razão entre a capacidade do enlace e a taxa de pico dos fluxos de vídeo cresce o parâmetro de espaço s tende a zero, a banda efetiva tende a taxa média e a taxa de utilização se aproxima do máximo.

6 Conclusões

O modelo proposto neste trabalho tira proveito das características dos servidores de vídeo sob demanda, onde os arquivos de vídeo permanecem armazenados, para antecipar a definição do ponto de operação do enlace (parâmetros tempo e espaço), com o objetivo de estimar a banda efetiva sob o regime assintótico de muitas fontes.

O cálculo da taxa mínima de serviço que garante a qualidade de serviço desejada é baseado em arquivos de *trace*, que são automaticamente gerados para simular a passagem dos arquivos de vídeo sobre a rede. Essa abordagem, além de não sofrer dos problemas existentes nas soluções baseadas em medições e em modelos de tráfego, possibilita a utilização do estimador baseado na teoria assintótica de muitas fontes que é bastante preciso, mas computacionalmente complexo.

Todo processo de admissão é auxiliado por um banco de dados, que simplifica a implementação e por um mecanismo de controle de tráfego, que garante a taxa de serviço necessária a cada agregado de *streams* ao mesmo tempo que aumenta a taxa de utilização do enlace.

A eficácia do mecanismo foi validada através de um software que utiliza os próprios arquivos de *trace* para simular o transporte e a multiplexação de vídeos *MPEG-4*. A análise dos resultados mostrou que a introdução do mecanismo minimiza o atraso e aumenta a taxa de utilização do enlace, e que a eficiência do sistema cresce com o aumento do número de fontes.

6.1 Trabalhos Futuros

O sistema proposto utiliza diversos enlaces virtuais, permitindo atribuir diferentes parâmetros de serviço à diferentes *streams*, porém essa abordagem é conservativa, pois o ganho decorrente da multiplexação entre os enlaces virtuais não é explorado. Adotando um esquema de alocação adaptativa de banda efetiva como o descrito por Li et al. (2005),

seria possível explorar a multiplexação entre os grupos de *streams*, definindo uma margem de alocação acima da capacidade real do enlace, através da diferença entre a banda efetiva medida no enlace real e o somatório da banda alocada para os enlaces virtuais. As respostas do protocolo *RTCP* podem ser usadas para verificar se os parâmetros de *QoS* estão sendo respeitados bem como para definir a prioridade do compartilhamento da banda excedente.

Para que a arquitetura possa ser aplicada também em sistemas menores, pode-se adotar uma abordagem dual, onde dois estimadores de banda efetiva são combinados. Nos casos em que há poucas fontes transmitindo, onde a eficácia do *MSAEB* é limitada, pode ser empregado um estimador que seja preciso com poucas fontes e *buffer* pequeno. Esse outro estimador poderia ser, por exemplo, o de Norros (*NEB*) ou o estimador baseado no Processo Envelope do Movimento Browniano Fractal (FONSECA; MAYOR; NETO, 2000), que deriva uma expressão para o cálculo da banda efetiva que obtém os mesmos resultados alcançados por Norros, mas sem fazer suposições quanto o tamanho do *buffer*.

Referências

- ABDALLA, M. *Análise de mecanismos de controle de admissão de conexão para redes ATM*. Dissertação (Mestrado) — Universidade Federal do Rio de Janeiro, PEE / COPPE - UFRJ, 1996.
- AVRAMOVA, Z. et al. Dimensioning of a Multi-rate Network Transporting Variable Bit Rate TV Channels. In: *International Conference on Communications (ICC09)*. Dresden, Germany: IEEE Computer Society, 2009.
- CALLADO, A.; KAMIENSKI, C. A.; KELNER, J. Uma Nova Abordagem para Recepção de vídeo vbr em camadas. *Simpósio Brasileiro em Sistemas Multitmídia e Hipermtídia (SBMIDIA2002)*, Fortaleza, CE, 2002.
- CHATFIELD, C. *The analysis of time series: an introduction*. sixth. Boca Raton, FL: Chapman & Hall/CRC, 2004.
- CHENG, X. Understanding the Characteristics of Internet Short Video Sharing. In: *Procs of the 7th ACM SIGCOMM Conference on Internet Measurement*. San Diego (CA, USA): ACM, 2007. p. 28.
- COURCOUBETIS, C.; SIRIS, V. Measurement and analysis of real network traffic. In: *Proceedings of the 7th Hellenic Conference on Informatics (HCI99)*. Ioannina, Greece: [s.n.], 1999.
- COURCOUBETIS, C.; SIRIS, V. Procedures and tools for analysis of network traffic measurements. *Performance Evaluation*, Elsevier, v. 48, n. 1-4, p. 5–23, 2002.
- COURCOUBETIS, C.; SIRIS, V.; STAMOULIS, G. Application of the many sources asymptotic and effective bandwidths to traffic engineering. *Telecommunication Systems*, v. 12, p. 167–191, 1999.
- COURCOUBETIS, C.; WEBER, R. Buffer Overflow Asymptotics For A Buffer Handling Many Traffic Sources. *Journal of Applied Probability*, v. 33, p. 886–903, 1995.
- COURCOUBETIS, C.; WEBER, R. Effective bandwidths for stationary sources. *Probability in the Engineering and Informational Sciences*, Cambridge University Press, v. 9, n. 02, p. 285–296, 1995.
- DEVERA, M. *Hierarchical Token Bucket*. 2004. Acesso em: 20 jun 2010. Disponível em: <<http://luxik.cdi.cz/~devik/qos/htb/>>.
- DRUMMOND, A. C.; FONSECA, N. L. S.; DEVETSIKIOTIS, M. Uma Avaliação de Estimadores de Banda Passante Baseados em Medições. In: *Congresso SBC, São Leopoldo, RS, Brasil*. [S.l.: s.n.], 2005.

- DUFFIELD, N. et al. Entropy of ATM traffic streams: a tool for estimating QoS parameters. *IEEE Journal on Selected Areas in Communications*, v. 13, n. 6, p. 981–990, 1995.
- DUFFIELD, N.; RAMAKRISHNAN, K.; REIBMAN, A. SAVE: An algorithm for smoothed adaptive video over explicit rate networks. *IEEE/ACM Transactions on Networking (TON)*, IEEE Press, v. 6, n. 6, p. 728, 1998.
- FALKNER, M. et al. Empirical effective bandwidths. In: *Proc. Global Telecommunications Conference (GLOBECOM 2000)*. San Francisco, CA: IEEE Computer Society, 2000. p. 672–678.
- FENG, W.; SECHREST, S. Smoothing and buffering for delivery of prerecorded compressed video. In: *IS&T/SPIE Multimedia Computing and Networking*. [S.l.: s.n.], 1995. p. 234–244.
- FERNANDEZ-VEIGA, M. et al. On the effectiveness of the many-sources asymptotic for admission control. *Computer Communications*, Elsevier, v. 26, n. 12, p. 1376–1391, 2003.
- FITZEK, F.; REISSLEIN, M. *MPEG-4 and H.263 Video Traces for Network Performance Evaluation*. Telecommunication Networks Group, Technische Universität Berlin, oct 2000. Acesso em: 25 jun 2010. Disponível em: <<http://www.tkn.tu-berlin.de/research/trace/trace.html>>.
- FONSECA, N.; MAYOR, G.; NETO, C. On the equivalent bandwidth of self-similar sources. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, ACM, v. 10, n. 2, p. 104–124, 2000.
- GARRETT, M.; WILLINGER, W. Analysis, modeling and generation of self-similar VBR video traffic. *ACM SIGCOMM Computer Communication Review*, ACM, v. 24, n. 4, p. 269–280, 1994.
- GEREOFFY, A. et al. *MPlayer - The Movie Player*. 2007. Acesso em: 23 jun 2010. Disponível em: <<http://www.mplayerhq.hu/design7/>>.
- GIBBENS, R. Traffic characterisation and effective bandwidths for broadband network traces. *Stochastic Networks, Theory and Applications*, p. 169–179, 1996.
- GIBBENS, R. J.; KELLY, F. P. Measurement-Based Connection Admission Control. In: *15th International Teletraffic Congress*. Statistical Laboratory, Wilberforce Road, Cambridge: Cambridge University Press, 1997.
- GROSSGLAUSER, M.; BOLOT, J. chrysostome. On the Relevance of Long-Range Dependence in Network Traffic. *IEEE/ACM Transactions on Networking*, v. 7, p. 15–24, 1996.
- GUERIN, R.; AHMADI, H.; NAGHSHINEH, M. Equivalent capacity and its application to bandwidth allocation in high-speed networks. *IEEE Journal on selected areas in communications*, v. 9, n. 7, p. 968–981, 1991.
- HACIOMEROGLU, F. *On-line Measurement-based Capacity Allocation Schemes*. Dissertação (Mestrado) — North Carolina State University, Department of Electrical and Computer Engineering, 2003.

- HUI, J. I. Resource Allocation for Broadband Networks. *IEEE Journal on Selected Areas in Communications*, v. 6, p. 1598–1608, 1988.
- IZQUIERDO, M. R.; REEVES, D. S. A Survey of Statistical Source Models for Variable-Bit-Rate Compressed Video. *Multimedia Systems*, v. 7, p. 199–213, 1999.
- KELLY, F. P. Notes on effective bandwidths. In: KELLY, F. P.; ZACHARY, S.; ZIEDINS, I. B. (Ed.). *Stochastic Networks: Theory and Applications*. Wellington Square, Oxford: Oxford University Press, 1996, (Royal Statistical Society Lecture Notes Series, 4). p. 141–168.
- KROPOTOFF, A. B. *Controle de Congestionamento em Redes ATM*. Dissertação (Mestrado) — UFRJ, COPPE-PESC, 1999.
- KUMAR, A.; MANJUNATH, D.; KURI, J. *Communication networking: an analytical approach*. [S.l.]: Morgan Kaufmann Publishers, 2004. 206-212 p.
- KUROSE, J. F.; ROSS, K. W. *Computer Networking: A Top-Down Approach*. 5th. ed. USA: Addison-Wesley Publishing Company, 2009.
- LELAND, W. et al. On the self-similar nature of Ethernet traffic. *IEEE/ACM Transactions on networking*, v. 2, n. 1, p. 1–15, 1994.
- LI, H. et al. A Robust Adaptive Effective Bandwidth Allocation Scheme. In: *IEEE International Conference on Communications*. [S.l.: s.n.], 2005. v. 1, p. 115.
- MCALARNEY, J.; HADDAD, R.; MCGARRY, M. *Turning Video Traces into Packet Traces for Network Simulation*. Department of Electrical and Computer Engineering, University of Akron, 2009.
- MCCANNE, S. et al. *Network Simulator NS-2.34*. 1997. Acesso em: 26 jun 2010. Disponível em: <<http://www.isi.edu/nsnam/ns/>>.
- MELO, C. *Modelagem e computação de banda passante equivalente de fluxos multifractais*. Tese (Doutorado) — Unicamp, Instituto de Matemática, Estatística e Computação Científica, 2004.
- MONTGOMERY, M.; VECIANA, G. D. On the Relevance of Time Scales in Performance Oriented Traffic Characterizations. In: *In Proc. IEEE INFOCOM*. San Francisco, CA: IEEE Computer Society, 1996. p. 513–520.
- NORROS, I. On the use of fractional Brownian motion in the theory of connectionless networks. *IEEE Journal on selected areas in communications*, v. 13, n. 6, p. 953–962, 1995.
- OSTROWSKY, L. *Modelos de trafego para fluxos gerados pelo protocolo UDP*. Dissertação (Mestrado) — Unicamp, Instituto de Matemática, Estatística e Computação Científica, 2005.
- PARK, K.; WILLINGER, W. *Self-similar network traffic and performance evaluation*. New York, NY, USA: John Wiley & Sons, Inc., 2000.

- PEREIRA, F.; EBRAHIMI, T. *The MPEG-4 book*. [S.l.]: Prentice Hall PTR Upper Saddle River, NJ, USA, 2002.
- PERROS, H.; ELSAYED, K. Call admission control schemes: a review. *IEEE Communications Magazine*, v. 34, n. 11, p. 82–91, 1996.
- PINHO, L. D.; ISHIKAWA, E.; AMORIM, C. D. Glove: a distributed environment for scalable video-on-demand systems. *International Journal of High Performance Computing Applications*, v. 17, n. 2, p. 147–161, 2003.
- RICHARDSON, I. H. *264 and MPEG-4 video compression: video coding for next-generation multimedia*. Chichwster, England: John Wiley & Sons Inc, 2003.
- SALEHI, J. et al. Supporting stored video: Reducing rate variability and end-to-end resource requirements through optimal smoothing. *IEEE/ACM Transactions on Networking (TON)*, IEEE Press, v. 6, n. 4, p. 397–410, 1998.
- SCHULZRINNE, H. et al. *RTP: A Transport Protocol for Real-Time Applications*. [S.l.], jul 2003. 104 p.
- SERES, G. et al. Alternative admission rules based on the many sources asymptotics. IEEE Computer Society, 2002.
- SILVA, D. J. da. *Análise de Qualidade de Serviço em Redes Corporativas*. Dissertação (Mestrado) — Unicamp, Instituto de Computação, 2004.
- STAHLSCHMIDT, T.; GODOY JR., W.; FORONDA, A. Effective bandwidth based connection admission control for stored variable bit rate video streams. In: *LATINCOM 2010*. Bogota, Colombia: IEEE Computer Society, 2010.
- TANENBAUM, A. *Computer Networks*. 4th. ed. USA: Prentice Hall, 2002.
- TURNER, J. New directions in communications (or which way to the information age?). *IEEE communications Magazine*, v. 24, n. 10, p. 8–15, 1986.
- VEITCH, D.; ABRY, P. A wavelet based joint estimator for the parameters of LRD, Special issue on Multiscale Statistical Signal Analysis and its Applications. *IEEE Trans. Info. Th*, v. 45, n. 3, p. 878–897, 1999.
- VIEIRA, F. *Contribuições ao Cálculo de Banda e de Probabilidade de Perda para Tráfego Multifractal de Redes*. Tese (Doutorado) — Unicamp, Faculdade de Engenharia Elétrica e Computação, 2006.
- WEISS, A. An Introduction to Large Deviations for Communications. *IEEE Journal of Selected Areas in Communications*, v. 13, n. 6, p. 938–952, 1995.
- WIDENIUS, M.; AXMARK, D.; MYSQL, A. *MySQL reference manual: documentation from the source*. Sebastopol, Ca: O’Reilly Media, 2002.
- WILLIAMSON, C. Network traffic measurement and modeling. In: ACM. *Proceedings of the 1995 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*. [S.l.], 1995. p. 56–57.

WYDROWSKI, B.; ZUKERMAN, M. QoS in best-effort networks. *IEEE Communications Magazine*, v. 40, n. 12, p. 44–49, 2002.

XIE, M.; HAENGGI, M. Statistical Performance Analysis of a Generalized Processor Sharing System by Using Large Deviations. In: *International Symposium on Mathematical Theory of Networks and Systems (MTNS'02)*. Notre Dame, IN: [s.n.], 2002.

APÊNDICE A – Simulador de Streams (NS-2)

```

# msasim.tcl
# Simula vários fluxos de datagramas baseados em arquivos de traces
# Thiago Stahlschmidt (CPGEI – UTFPR)
#

# Checa os parâmetros da simulação
if {$argc < 3} {
    puts "Use: ns msasim.tcl trace.tr N (fontes) C (capacidade em Mbps)"
    exit -1
}

# Nome do arquivo de trace
set ascii_trace_file [lindex $argv 0]
# Número de fluxos (N)
set nclients [lindex $argv 1]
# Capacidade (C) em Mbps
set bandwidth [lindex $argv 2]

# Nova Simulação
set ns [new Simulator]

# Suprime advertência de debug
Tracefile set debug_ 0

# Define o formato do trace
Agent/UDP set tracevar_ true

# Define o tamanho máximo do datagrama
Agent/UDP set packetSize_ 1400

# Servidor
set server_node [$ns node]

```

```

# Janela de tempo em segundos (1/fps)
set epoch 0.040000

# Buffer em KB(s)
set backlog 1000000

# Roteador IP
set router_node [$ns node]
$router_node shape "box"

# Define o enlace entre os nós
$ns duplex-link $server_node $router_node [expr $bandwidth]Mb 10ms DropTail

# Para usar o pacote RPI, apos sua instalação , deve-se definir a variável
# de ambiente NSRPIPATH com o local da instalação do NS.
# (export NSRPIPATH=/usr/local/ns-allinone-x.x/ns-x.x)
# e definir USERPI como true.
set USERPI true
if {$USERPI} {
    source $env(NSRPIPATH)/tcl/rpi/link-stats.tcl
    set stats [new LinkStats $server_node $router_node]
}

# Define o tamanho das filas em KBytes (qlim_ * mean_pktsize_)
Queue/DropTail set queue_in_bytes_ true
Queue/DropTail set mean_pktsize_ 1000
$ns queue-limit $server_node $router_node $backlog

# Define as fontes e os destinos
for {set i 0} {$i < $nclients} {incr i} {
    set udp($i) [new Agent/UDP]
    $ns attach-agent $server_node $udp($i)
    set snk($i) [new Agent/Null]
    $ns attach-agent $router_node $snk($i)
    $ns connect $udp($i) $snk($i)
    $udp($i) set fid_ [expr $i + 1]
}

# Converte o arquivo de tráfego no formato binário do ns-2
# (tempo entre pacotes em milisegundos) (tamanho do pacote em bytes)
set ascii_file_id [open $ascii_trace_file r]
set trace_file_name video.dat
set trace_file_id [open $trace_file_name w]

```

```

fconfigure $trace_file_id -encoding binary
set last_time 0
while {[eof $ascii_file_id] == 0} {
    gets $ascii_file_id line
    scan $line "%d%d" next_time length
    set time [expr 1000 * ($next_time - $last_time)]
    set last_time $next_time
    puts -nonewline $trace_file_id [binary format "II" $time $length]
}
close $ascii_file_id
close $trace_file_id

# Carrega o arquivo de trace criado anteriormente
set trace_file [new Tracefile]
$trace_file filename $trace_file_name

# Define os agentes utilizando os traces de tráfego
for {set i 0} {$i < $nclients} {incr i} {
    set video($i) [new Application/Traffic/Trace]
    $video($i) attach-agent $udp($i)
    $video($i) attach-tracefile $trace_file
}

# Define o ponto onde se deseja analisar os eventos
set nd [open out.tr w]
$ns trace-queue $server_node $router_node $nd

# Inicia a simulação
# cada fluxo inicializará em um momento e uma posição aleatória
set rng [new RNG]
$rng seed 0
for {set i 0} {$i < $nclients} {incr i} {
    set starttime [format "%8.6f" [expr [$rng uniform $min $max]]]
    $ns at $starttime "$video($i) start"
}

# Para a simulação
$ns at [expr 5.0 * $last_time / 1000 + 0.001] {
    finish
}

# Procedimento final
proc finish {} {

```



```

global ns nd
$ns flush-trace
close $nd
if {$USERPI} {
    global stats bandwidth
    set maxqueuebits [expr [$stats get-max-byte-queue-length] * 8]
    set bandwidthbps [expr $bandwidth * 1000000]
    set maxqueuedelay [expr 1.0 * $maxqueuebits / $bandwidthbps]
    set maxqueuelength [$stats get-max-byte-queue-length]
    set meandelay [format "%8.6f" [$stats get-mean-queue-delay]]
    set maxdelay [format "%8.6f" $maxqueuedelay]
    set npackets [$stats get-packet-arrivals]
    puts "Estatísticas da Fila: "
    puts "Número de pacotes: $npackets"
    puts "Atraso médio (segundos): $meandelay"
    puts "Atraso máximo (segundos): $maxdelay"
    puts "Tamanho maximo atingido (bytes): $maxqueuelength"
}
exit 0
}

```

```
$ns run
```

```

# ns2stats.awk
# Analisa a saída do NS-2 caso o RPI não esteja disponível.
# Para evitar problemas de ponto flutuante com virgula,
# usar "LC_ALL=C" antes de invocar esse script.
BEGIN {
    packets=0;
    maxsize=0;
    maxdelay=0;
    current=0;
}
/^[+]/{
    time = $2
    pid = $12
    current += $6
    packets++
    packetenq[pid] = time
    if (current > maxsize) {
        maxsize = current;
    }
}

```

```

    }
}
/^[^-]/{
    time = $2
    pid = $12
    current -= $6
    packetdelay[pid] = time - packetenq[pid]
    delete packetenq[pid]
    if (packetdelay[pid] > maxdelay) {
        maxdelay = packetdelay[pid]
    }
}
/^[d]/{
    current -= $6
    maxsize -= $6;
}
END {
    totaldelay = 0
    for (i=0;i<packets;i++)
        totaldelay += packetdelay[i]
    printf("Pacotes: %d BufferMaximo: %d bytes", packets, maxsize);
    printf("AtrasoMedio: %8.6f s AtrasoMaximo: %8.6f s\n",
        totaldelay/packets, maxdelay)
}

```

APÊNDICE B – Base de Dados do Controle de Admissão (MySQL)

```
# cacdb.sql
```

```
CREATE TABLE recursos_enlace(
  capacidade FLOAT UNSIGNED NOT NULL COMMENT 'Capacidade em mbps',
  buffer INT UNSIGNED NOT NULL COMMENT 'Tamanho da fila em bytes')
COMMENT = 'Descreve o enlace através de seus recursos';
```

```
CREATE TABLE videos(
  id INT UNSIGNED NOT NULL AUTO_INCREMENT COMMENT 'Identificador do video',
  nome VARCHAR(256) NOT NULL COMMENT 'Nome do arquivo',
  media FLOAT UNSIGNED NOT NULL COMMENT 'Taxa media (Mbps)',
  pico FLOAT UNSIGNED NOT NULL COMMENT 'Taxa de pico (Mbps)',
  PRIMARY KEY (id),
  INDEX IDXNOME USING HASH (nome))
COMMENT = 'Informações sobre os arquivos de video';
```

```
CREATE TABLE banda_efetiva(
  video_id INT UNSIGNED NOT NULL COMMENT 'Identificador do video',
  fluxos INT UNSIGNED NOT NULL COMMENT 'Numero de fluxos',
  capacidade FLOAT UNSIGNED NOT NULL COMMENT 'Capacidade utiliza pelos
  fluxos (Mbps)',
  buffer INT UNSIGNED NOT NULL COMMENT 'Buffer utilizado pelos fluxos
  (Bytes)',
  PRIMARY KEY (video_id, fluxos),
  INDEX idx_fk (video_id),
  INDEX idx_pk USING HASH (video_id, fluxos),
  FOREIGN KEY (video_id) REFERENCES videos(id) ON DELETE RESTRICT ON
  UPDATE CASCADE)
COMMENT = 'Capacidade mínima necessária';
```

```
CREATE TABLE conexoes_ativas(
  video_id INT UNSIGNED NOT NULL COMMENT 'Identificador do video',
```

```

    fluxos INT UNSIGNED NOT NULL COMMENT 'Numero de fluxos ',
    INDEX idx_fk (video_id),
    FOREIGN KEY (video_id) REFERENCES videos(id) ON DELETE RESTRICT ON
        UPDATE CASCADE)
COMMENT = 'Mantém o registro das conexões ativas ';

CREATE VIEW recursos_utilizados(capacidade, buffer) AS
    SELECT ROUND(SUM(be.capacidade), 2), SUM(be.buffer) FROM conexoes_ativas
        ca, banda_efetiva be
    WHERE ca.video_id = be.video_id AND be.fluxos = ca.fluxos;

CREATE VIEW recursos_disponiveis(capacidade, buffer) AS
    SELECT (re.capacidade - ru.capacidade), (re.buffer - ru.buffer) FROM
        recursos_utilizados ru, recursos_enlace re;

CREATE VIEW recursos_economia(capacidade) AS
    SELECT ROUND(SUM(pico * fluxos) - ru.capacidade, 2) FROM conexoes_ativas
        ca, videos v, recursos_utilizados ru WHERE ca.video_id = v.id;

CREATE TRIGGER inicializa_fluxos AFTER INSERT ON videos FOR EACH ROW
    INSERT conexoes_ativas VALUES (NEW.id, 0);

DELIMITER //
CREATE FUNCTION admite_fluxo(nome VARCHAR(256)) RETURNS INT UNSIGNED
COMMENT 'Controle de admissão (retorna o id do fluxo admitido ou zero caso
    não haja recursos suficientes)'
BEGIN
    DECLARE video_id, fluxos, buso, breq, bdisp INT UNSIGNED;
    DECLARE cuso, creq, cdisp FLOAT UNSIGNED;
    SELECT v.id, ca.fluxos INTO video_id, fluxos FROM videos v,
        conexoes_ativas ca WHERE ca.video_id = v.id AND v.nome = nome;
    SELECT rd.capacidade, rd.buffer INTO cdisp, bdisp FROM
        recursos_disponiveis rd;
    SELECT be.buffer, be.capacidade INTO buso, cuso FROM banda_efetiva be
        WHERE be.video_id = video_id AND be.fluxos = fluxos;
    SELECT be.buffer, be.capacidade INTO breq, creq FROM banda_efetiva be
        WHERE be.video_id = video_id AND be.fluxos = fluxos + 1;
    SET breq = breq - buso;
    SET creq = creq - cuso;
    IF (breq <= bdisp) && (creq <= cdisp) THEN
        UPDATE conexoes_ativas ca SET ca.fluxos = fluxos + 1 WHERE ca.video_id
            = video_id;
    ELSE

```

```

        SET video_id = 0;
    END IF;
    RETURN video_id;
END
//

CREATE FUNCTION libera_fluxo(nome VARCHAR(256)) RETURNS INT UNSIGNED
COMMENT 'Libera um fluxo de dados'
BEGIN
    DECLARE video_id, fluxos INT UNSIGNED;
    SELECT v.id, ca.fluxos INTO video_id, fluxos FROM videos v,
        conexoes_ativas ca WHERE ca.video_id = v.id AND v.nome = nome;
    IF (fluxos > 0) THEN
        UPDATE conexoes_ativas ca SET ca.fluxos = fluxos - 1 WHERE
            ca.video_id = video_id;
        RETURN 1;
    END IF;
    RETURN 0;
END
//

CREATE PROCEDURE inicializa()
COMMENT 'Deve ser executado toda vez que o serviço for iniciado'
BEGIN
    UPDATE conexoes_ativas ca SET ca.fluxos = 0;
END
//
DELIMITER ;

```

APÊNDICE C – Classificação e Controle de Tráfego (IPTABLES/TC)

```
# Uso dos mecanismos de classificação e controle de tráfego
# IPTABLES/TC/HTB
# Define a nova disciplina de controle de tráfego PFIFO (DEFAULT) -> HTB
tc qdisc del dev eth0 root
tc qdisc add dev eth0 root handle 1:0 htb default 9999
# Classe raiz (capacidade do enlace 100Mbps)
tc class add dev eth0 parent 1:0 classid 1:0 htb rate 100mbit ceil 100mbit
# Classe padrão para pacotes não classificados (9999)
tc class add dev eth0 parent 1:0 classid 1:9999 htb rate 1mbit ceil 100mbit
# Define as classes de vídeo (classes = SELECT MAX(id) FROM videos)
for (vid = 1; vid <= classes; vid++)
    tc class add dev eth0 parent 1:0 classid 1:${vid} htb rate 500kbit ceil
    100mbit
# Redefine a capacidade do enlace virtual do vídeo
# identificado pela variável VIDEOID (início/fim de uma transmissão)
tc -s -d class change dev eth0 parent 1:0 classid 1:${VIDEOID} htb rate
    1mbit ceil 100mbit
# Marca os pacotes do stream de vídeo de acordo
# com a porta e o IP de destino (início de transmissão)
iptables -t nat -A FORWARD -o eth0 -p udp -d $CLIENTIP --dport $CLIENTPORT
    -j CLASSIFY --set-class 1:$VIDEOID
# Remove a marcação (fim de transmissão)
iptables -t nat -D FORWARD -o eth0 -p udp -d $CLIENTIP --dport $CLIENTPORT
    -j CLASSIFY --set-class 1:$VIDEOID
```