

**Marcus Vinícius de Melo Rocha**

***Impactos da interatividade na escalabilidade de  
protocolos para mídia contínua***

Belo Horizonte – MG

Dezembro / 2007

**Marcus Vinícius de Melo Rocha**

***Impactos da interatividade na escalabilidade de  
protocolos para mídia contínua***

Tese de Doutorado apresentada ao Curso de Pós-graduação em Ciência da Computação da Universidade Federal de Minas Gerais, como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Orientador:

Prof. Dr. Sérgio Vale Aguiar Campos

Co-orientadora:

Prof.<sup>a</sup> Dr.<sup>a</sup> Jussara Marques Almeida

PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO  
UNIVERSIDADE FEDERAL DE MINAS GERAIS

Belo Horizonte – MG

Dezembro / 2007

Tese de Doutorado sob o título *Impactos da interatividade na escalabilidade de protocolos para mídia contínua* defendida por Marcus Vinícius de Melo Rocha e aprovada pela banca examinadora constituída por:

---

Prof. Dr. Sérgio Vale Aguiar Campos  
UFMG  
Orientador

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Jussara Marques de Almeida  
UFMG  
Co-orientadora

---

Prof. Dr. Antônio Alfredo Ferreira Loureiro  
UFMG

---

Prof. Dr. Edmundo de Souza e Silva  
UFRJ

---

Prof. Dr. Ivan Moura Campos

---

Prof. Dr. José Marcos Silva Nogueira  
UFMG

---

Prof<sup>a</sup>. Dr<sup>a</sup>. Rosa Maria Meri Leão  
UFRJ

*A meus pais,  
a minha esposa,  
a meus filhos,  
a minha avó Rosinha e a meus padrinhos Garrido e Alzirinha  
que, de longe, olham por mim.*

# *Agradecimentos*

Não, não tem jeito, eu tenho de falar. A lista é longa, que já faz tempo que não estou sozinho. Não estou, nunca estive.

Lília é minha esposa. Honrou-me quando escolheu dividir sua vida comigo. Vive comigo todas as dores e as alegrias da vida e, é claro, do curso. Como lhe agradecer isso? Aonde teria chegado sem tê-la a meu lado?

Bruno e Pedro são meus filhos amados, e seus nomes aparecem aqui na ordem em que vieram ao mundo. Dois momentos especiais na minha vida, pude assistir aos dois nascimentos e deles participar – e contar os dedinhos dos pezinhos e das mãozinhas para saber que estava tudo bem – e de tantos outros momentos alegres ou tristes em suas vidas de crianças, adolescentes, homens. Não há como descrever a alegria infinda de ser pai dessas duas pessoas.

Osíris e Vera são meus pais. Ele já se foi, faz alguns anos. Trouxeram-me ao mundo, nutriram-me, criaram-me. Apontaram-me o caminho. Não há como medir o que devo a eles, o quanto os admiro.

Luisa e Zina sempre foram muito mais que tias. Zina foi-se faz tempo e deixou-nos saudades. Luisa ora por nós todos a cada instante e usa sua memória prodigiosa para acalantar os laços da família.

Rose e Di, minhas irmãs. Seguimos todos os três, cada um da sua forma, pelas veredas acadêmicas nas quais nosso pai abriu caminhos e deixou marca.

Aos orientadores, Prof. Sérgio e Prof<sup>a</sup>. Jussara, agradeço a imensa paciência e a persistência frente às minhas limitações e o estoque constante de desafios que me propiciaram. Esses desafios é que me permitiram defrontar com minhas limitações e aprender a lidar com elas. E permitiram-me, por fim, entender o que é *pesquisa*. Agradeço-lhes, também, as palavras sábias e tranquilizadoras que me ajudaram a prosseguir quando a missão parecia pesada demais para ser levada adiante.

Agradeço aos Profs. Berthier, Loureiro e Rosa, que compuseram a banca de defesa do projeto de tese e avaliaram etapa anterior deste trabalho, e aos Profs. Edmundo, Ivan, José Marcos, Loureiro e Rosa por dedicarem seu tempo a avaliar o trabalho presente.

Sou devedor de meus colegas do Vod – Laboratório de Vídeo Sob Demanda –, que sempre proporcionaram um ambiente de trabalho leve e tranquilo, mesmo nos períodos em que este trabalho exigiu atenção, concentração e cuidado, sempre dispostos a responder minhas incontáveis perguntas. Alex, Claudiney, Cristiano, Daniel, Israel, Ítalo, Itamar e Vanessa são pessoas especiais. Agradeço especialmente ao Alex e ao Eric, que forneceram dados de topologias reais para alguns de nossos experimentos. E, muito especialmente, ao Marcelo e ao Ítalo, que, mais que colegas ou amigos, são companheiros de armas. Lutamos e vencemos juntos as batalhas mais ferozes e ensandecidas de todo o trabalho.

Agradeço a todo o corpo docente e administrativo do DCC e, em especial, à Renata e à Túlia, sempre prontas a nos receber e solucionar dificuldades.

Agradeço ao colega Bernardo, do LAND/UFRJ, a ajuda com as cargas do sistema MANIC e ao colega Kleber, também do LAND/UFRJ, por diversas discussões proveitosas.

Agradeço à Assembléia Legislativa de Minas Gerais, que viabilizou este trabalho, e, em especial, a Marcelo Migueletto, Flávio Hannas, Rodrigo Lucena, Alaor Messias, Sheyla Abreu e Leonardo Boechat, cujo apoio me foi imprescindível. E, é claro, aos colegas e amigos Daniele, Helson, Leonardo, Laertes, Marcelo, Maurício e Roberto, que conviveram pacificamente com minha ausência nos períodos mais críticos do curso.

Por fim, agradeço a todos que, de alguma forma, contribuíram com esta caminhada. Por vezes com contribuições diretas e, em outras tantas, apenas com olhares discretos de esperança e aprovação.

Àquele a quem agradeço a Vida,  
a Ele não ousou agradecer mais nada.

*Sempre que olho para trás, e olho e olho,  
eu olho e é sempre isto o que eu vejo.  
Uma sucessão infinda de sonhos em real transfeitos,  
e também uma sucessão de ilusões perdidas.  
De amores-ódios, encontros mais ou menos profundos.*

*E, ao longo de cada caminho que algum dia ousei percorrer,  
e também antes do início, e depois também do fim,  
um estoque interminável de lágrimas.  
Lágrimas doces, lágrimas amargas.  
Mas, sempre e sempre, lágrimas salgadas que me escorrem pelas faces.*

*Eu me pergunto se tudo se resume a isto, tão só e unicamente a isto.  
Viver, e sorrir e chorar, e nunca saber realmente o porquê.*

*Se somos apenas jagunços em guerra, jogando truco com o coisa-ruim,  
pondo um Zap na testa e dando gargalhada do destino,  
nos enfiando em mais uma vereda,  
errando rumo pelo sertão sem fim,  
matando pra não morrer,  
pra viver só mais um dia,  
só mais um dia pra matar o tempo, o tempo, o tempo...*

**Marcus Rocha**



# Resumo

Recentemente, vários protocolos escaláveis de compartilhamento de banda têm sido propostos com o objetivo de obter redução significativa nas bandas de servidor e de rede necessárias para a entrega de mídias populares. Embora a escalabilidade desses protocolos tenha sido avaliada para acesso seqüencial, alto grau de interatividade já foi observado em acesso a diversos servidores reais de mídia contínua. Além disso, alguns estudos indicaram que a interatividade pode penalizar seriamente a escalabilidade dos protocolos de mídia contínua com compartilhamento de banda e, assim, comprometer a popularidade das aplicações de mídia contínua na Internet. Dentro do contexto dos protocolos escaláveis de mídia contínua, são de particular interesse os protocolos com serviço imediato, nos quais os usuários são atendidos com uma latência muito baixa, e que se prestam a aplicações para as quais a interatividade e, portanto, a resposta rápida são requisitos.

O objetivo desta tese é investigar o impacto da interatividade na escalabilidade de protocolos de mídia contínua com compartilhamento de banda e serviço imediato e propor novos mecanismos que reduzam esse impacto. Esta tese identifica um conjunto reduzido de fatores de impacto da interatividade sobre a escalabilidade de protocolos de mídia contínua e usa esses fatores como base para criar um modelo de cargas interativas. Com base nesses fatores, constrói um gerador de cargas realistas de mídia contínua. Esse gerador, associado a um conjunto amplo de cargas reais (que cobre diversas aplicações de mídia contínua), é usado como base para construir um rico conjunto de cargas sintéticas interativas realistas. Essas cargas são, então, usadas para avaliar o impacto da interatividade na escalabilidade de dois protocolos correntes de mídia contínua, *Patching* e *Bandwidth Skimming*. Uma nova expressão para a escalabilidade do *Bandwidth Skimming* (válida para as cargas de mídia contínua aqui empregadas) é obtida a partir de dados de simulações e indica que o impacto da interatividade é ainda maior que o estimado por trabalhos anteriores.

Esta tese faz uma exploração ampla do espaço de projeto de protocolos para mídia contínua interativa, a fim de propor mecanismos capazes de minimizar o impacto da interatividade. Essa exploração tem como base o protocolo *Bandwidth Skimming*, mas pode ser facilmente estendida a outros protocolos de mídia contínua. Inicialmente, são apresentadas diversas estratégias focadas em aspectos específicos das cargas de mídia contínua e em aspectos do protocolo base, o *Bandwidth Skimming*. O efeito da fragmentação de requisições, um fenômeno que afeta qualquer protocolo que use *buffer* local ao cliente como forma de reduzir a retransmissão, é identificado e avaliado. Esse efeito pode reduzir a economia obtida por uma estratégia e até mesmo fazer com que a banda média de servidor requerida pela estratégia supere a requerida pelo protocolo original. Como forma de avançar a exploração do espaço de projeto de protocolos, combinam-se diversas estratégias na construção de estratégias híbridas. O impacto dessas estratégias sobre a banda requerida no cliente e na rede é também investigado de forma preliminar. Nessa avaliação a *dispersão espacial* e a *dispersão temporal*, apresentadas por este trabalho como fatores das cargas interativas que capturam seu impacto primário sobre a escalabilidade de protocolos, são usadas como um arcabouço que facilita significativamente a avaliação dos

resultados do grande número de experimentos e simulações efetuados.

A dispersão, apresentada em duas dimensões (temporal e espacial), é ferramenta útil à avaliação de escalabilidade de protocolos. No entanto, é de difícil mensuração. Portanto, esta tese define dispersão como um parâmetro *único* das cargas de mídia contínua e usa-a para medir o impacto da interatividade na escalabilidade dos protocolos *Patching* e *Bandwidth Skimming*. A dispersão é, então, usada para estimar a escalabilidade do *Bandwidth Skimming*.

Por fim, como as estratégias propostas (e também outras estratégias que usem *buffer* local ao cliente) são sujeitas ao efeito degenerativo da fragmentação, a exploração do espaço de projeto de protocolos é concluída com a apresentação de um protocolo adaptativo que monitora a carga como forma de escolher, dinamicamente, qual a melhor estratégia a empregar para uma certa carga. Esse protocolo usa a dispersão para guiar suas decisões a fim de evitar o efeito degenerativo da fragmentação. Com isto, obtém economia de banda de servidor (de até 66%) para uma faixa de dispersões mais ampla que qualquer estratégia individual.

A melhor estratégia híbrida proposta reduz a banda média de servidor requerida pelo protocolo *Bandwidth Skimming* original para cargas altamente interativas, em até 53% para *buffer* de tamanho ilimitado e em até 46% para *buffer* restrito a apenas 50% do tamanho da mídia.

# *Abstract*

A number of scalable bandwidth sharing protocols have been proposed, with the goal of greatly reducing the server and network bandwidth required to deliver popular media. The scalability of these protocols have been assessed for sequential workloads. However, high degree of interactivity has been observed in real media workloads. Furthermore, a number of studies show that interactivity seriously affects the scalability of the protocols and, thus, can compromise the popularity of streaming media in the Internet. In this area, immediate service protocols are of particular interest, as they serve clients with no or little delay, and can be used where fast interactive response is required.

We investigate the impact of interactivity in the scalability of bandwidth sharing streaming media protocols with immediate service, and propose new strategies do reduce this impact. We identify a reduced set of interactive media workloads with impact in the scalability of such protocols, and use these factors as a basis to create a model for streaming media workloads. We use this model, together with a large set of real streaming media workloads, to build a rich set synthetic interactive workloads which covers a wide range of interactivity profiles. We use this set of workloads to evaluate the impact of interactivity in *Patching* and *Bandwidth Skimming*.

A number of different and complementary strategies is proposed to reduce the impact of the interactivity of *Bandwidth Skimming*, which scales better than *Patching* for sequential and interactive workloads.

We uncover the degenerative effect of *request fragmentation* which, in some degree, affects any protocol which uses client-side buffering to reduce data retransmission. In order to comprehensively cover the design space of streaming media protocols, we also propose hybrid strategies which combine efforts of different individual strategies.

*Dispersion* is defined as a factor of streaming media workloads which is key to the scalability of protocols. We start with a qualitative approach to dispersion, which is used as a framework for protocols analysis and evaluation. We follow with a quantitative approach to dispersion, which is used to estimate the server bandwidth required to the Bandwidth Skimming protocol.

As every strategy is affected by the degenerative effect of request fragmentation, we propose an *adaptive protocol*. This protocol monitors the workload, measuring it's dispersion, and uses this dispersion as a guide to decide which strategy, if any, to apply to a given workload. This protocol can be successfully used with any of our workloads (educational video, entertainment video and audio), and avoids the degenerative effect of request fragmentation. This protocol saves up to 66% of server bandwidth if compared to the original Bandwidth Skimming protocol.

adf

# *Sumário*

## **Lista de Figuras**

## **Lista de Tabelas**

<b>1</b>	<b>Introdução</b>	p. 18
<b>2</b>	<b>Trabalhos relacionados</b>	p. 24
2.1	Protocolos . . . . .	p. 24
2.1.1	Protocolos de mídia contínua . . . . .	p. 24
2.1.2	Mídia contínua para usuários interativos . . . . .	p. 31
2.1.3	Impacto da interatividade na escalabilidade de protocolos . . . . .	p. 33
2.2	Cargas interativas de mídia contínua . . . . .	p. 34
2.2.1	Caracterização de cargas de mídia contínua . . . . .	p. 34
2.2.2	Geração de cargas sintéticas de mídia contínua . . . . .	p. 37
<b>3</b>	<b>Cargas interativas de mídia contínua</b>	p. 40
3.1	Dispersão temporal e espacial . . . . .	p. 40
3.2	Definição formal para a dispersão . . . . .	p. 45
3.3	Impacto da dispersão na escalabilidade dos protocolos <i>Bandwidth Skimming</i> e <i>Patching</i> . . . . .	p. 46
3.3.1	Impacto da dispersão sobre a escalabilidade do protocolo <i>Bandwidth</i> <i>Skimming</i> com cargas interativas . . . . .	p. 47
3.3.2	Impacto da dispersão sobre a escalabilidade do protocolo <i>Bandwidth</i> <i>Skimming</i> com cargas sequenciais . . . . .	p. 51

3.3.3	Impacto da dispersão sobre a escalabilidade do protocolo <i>Patching</i> . . .	p. 54
3.3.4	Métricas de impacto da dispersão sobre a escalabilidade de outros protocolos . . . . .	p. 55
3.4	Cargas reais de mídia contínua interativa . . . . .	p. 57
3.5	Síntese de cargas realistas de mídia contínua interativa . . . . .	p. 60
<b>4</b>	<b>O impacto da interatividade nos protocolos correntes de mídia contínua</b>	p. 67
4.1	Comparando a escalabilidade dos protocolos <i>Patching</i> e <i>Bandwidth Skimming</i>	p. 67
4.2	Escalabilidade do <i>Bandwidth Skimming</i> com cargas interativas em função de <i>N</i>	p. 69
4.3	Escalabilidade do <i>Bandwidth Skimming</i> com cargas interativas em função da dispersão . . . . .	p. 72
<b>5</b>	<b>Estratégias de transmissão de mídia contínua para cargas interativas</b>	p. 76
5.1	Modelagem da rede e arquitetura de comunicação . . . . .	p. 76
5.2	Descrição das estratégias . . . . .	p. 77
5.3	Avaliação . . . . .	p. 82
5.4	Explorando o período inativo . . . . .	p. 86
5.4.1	Estratégias . . . . .	p. 87
5.4.2	Avaliação . . . . .	p. 89
5.4.3	Aspectos de implementação . . . . .	p. 93
5.5	Estratégias híbridas . . . . .	p. 94
5.5.1	Estratégias . . . . .	p. 94
5.5.2	Avaliação . . . . .	p. 94
5.5.3	Estratégias híbridas como envelopes das estratégias individuais que as integram . . . . .	p. 97
5.5.4	Aspectos de implementação . . . . .	p. 98
<b>6</b>	<b>Requisitos de banda de cliente e de rede</b>	p. 99
6.1	Banda de cliente para a estratégia SP . . . . .	p. 99

6.2	Banda de rede para a estratégia SP . . . . .	p. 100
6.3	Banda de rede para estratégias híbridas em topologia de rede real . . . . .	p. 103
6.3.1	Topologia da rede real . . . . .	p. 103
6.3.2	Banda de rede . . . . .	p. 103
<b>7</b>	<b>Estratégia adaptativa para transmissão de mídia contínua com cargas interativas</b>	<b>p. 106</b>
7.1	Protocolo adaptativo . . . . .	p. 106
7.2	Resultados . . . . .	p. 109
<b>8</b>	<b>Conclusões e trabalhos futuros</b>	<b>p. 112</b>
	<b>Referências Bibliográficas</b>	<b>p. 115</b>

# *Lista de Figuras*

2.1	Esquema de funcionamento do protocolo <i>Batching</i> . . . . .	p. 25
2.2	Esquema de funcionamento do protocolo <i>Piggybacking</i> . . . . .	p. 26
2.3	Esquema do uso de <i>buffer</i> para armazenar conteúdo a ser exibido no futuro . . . . .	p. 27
2.4	Esquema do protocolo <i>Patching</i> . . . . .	p. 28
2.5	Protocolo <i>Bandwidth Skimming</i> . . . . .	p. 29
2.6	Modelo hierárquico do comportamento de clientes. Fonte: [25] . . . . .	p. 35
3.1	Dispersão em cargas de mídia contínua . . . . .	p. 42
3.2	Extensão da chegada de requisições . . . . .	p. 47
3.3	Duas requisições com sobreposição . . . . .	p. 48
3.4	Blocos sobrepostos em duas requisições . . . . .	p. 49
3.5	Impacto de uma terceira requisição . . . . .	p. 50
3.6	Sobreposição com sete requisições . . . . .	p. 50
3.7	Sobreposição numa carga seqüencial . . . . .	p. 51
3.8	Primeiro e último blocos sobrepostos numa carga seqüencial . . . . .	p. 52
3.9	Sobreposição de requisições superestima a dispersão . . . . .	p. 54
3.10	Perfis de comportamento interativo para cargas reais . . . . .	p. 59
3.11	Modelo de histograma de uma carga simples . . . . .	p. 62
3.12	Perfis de comportamento interativo para cargas sintéticas . . . . .	p. 64
3.13	Número médio de fluxos simultâneos . . . . .	p. 65
4.1	Banda média de servidor para <i>Unicast</i> , <i>Patching</i> e <i>Bandwidth Skimming</i> . . . . .	p. 68
4.2	Banda média do <i>Bandwidth Skimming</i> e banda calculada pela equação (4.1) para três valores de $k$ . . . . .	p. 70



4.3	Regressão da banda média do <i>Bandwidth Skimming</i> à equação (4.2) . . . . .	p. 71
4.4	Transmissão com <i>Unicast</i> ( $\Delta_{BS} = 0,4$ ) . . . . .	p. 72
4.5	Transmissão com <i>Bandwidth Skimming</i> . . . . .	p. 73
4.6	Economia do <i>Bandwidth Skimming</i> sobre <i>Unicast</i> . . . . .	p. 74
5.1	Escolha de fluxos de período inativo . . . . .	p. 80
5.2	Economia de banda média de servidor com respeito ao <i>Bandwidth Skimming</i>	p. 83
5.3	Fragmentação de requisições sob KMB (carga IA, $N=66$ , <i>buffer</i> ilimitado) . .	p. 85
5.4	Economia de banda de servidor para diferentes níveis de interatividade . . . .	p. 90
5.5	Preenchimento do <i>buffer</i> (Carga IA) . . . . .	p. 91
5.6	Porcentagem do período inativo usada para escutar fluxos . . . . .	p. 92
5.7	Fragmentação do <i>buffer</i> de SP, carga IA . . . . .	p. 93
5.8	Impacto da restrição no <i>buffer</i> (Carga IA, $N=652$ ) . . . . .	p. 93
5.9	Economia de banda de servidor para estratégias híbridas . . . . .	p. 95
5.10	Impacto da restrição do <i>buffer</i> . . . . .	p. 96
5.11	Economia de banda com cargas de vídeo educacional para duas estratégias híbridas e para as estratégias individuais que elas empregam . . . . .	p. 97
6.1	SILENT PREFETCH - Economia de banda de servidor com respeito ao <i>Bandwidth Skimming</i> ( <i>buffer</i> ilimitado no cliente) . . . . .	p. 100
6.2	Modelo da topologia canônica . . . . .	p. 101
6.3	SILENT PREFETCH - Economia de banda média de rede com respeito ao <i>Bandwidth Skimming</i> com diferentes tamanhos de <i>buffer</i> ( $k=5$ ) . . . . .	p. 102
6.4	Economia de banda de rede para nível de interatividade alto . . . . .	p. 104
7.1	Erro do protocolo adaptativo para diferentes dispersões de referência . . . .	p. 110
7.2	Economia de banda para o protocolo adaptativo (limiar de dispersão 0.30) . .	p. 111

## *Lista de Tabelas*

3.1	Notação . . . . .	p. 46
3.2	Sumário das cargas de vídeo (CV=Coeficiente de Variação) . . . . .	p. 57
3.3	Sumário das cargas de áudio (CV=Coeficiente de Variação) . . . . .	p. 57

# 1 *Introdução*

Desde seu início, a distribuição de mídia contínua na Internet tem se defrontado com um grande desafio: a limitação severa de escalabilidade que se origina dos fortes requisitos, em termos de banda de servidor e de rede, que suas aplicações impõem. O sistema convencional de distribuição por *Unicast*, com um fluxo dedicado a cada usuário, claramente não apresenta boa escalabilidade, o que limita a entrega de mídia a um número grande de usuários. Os esforços voltados para a redução de banda têm levado a uma grande variedade de novos mecanismos, incluindo métodos de codificação eficientes para economia de banda [81], estratégias de *caching* para a redução da carga de rede e de servidor [72] e protocolos escaláveis que se apóiam no compartilhamento de fluxos para a redução dos requisitos de banda [11, 12, 14, 23, 30, 32, 33, 38, 43, 44, 55].

Dentre os protocolos de mídia contínua com escalabilidade existentes, *Patching* [14, 43] e *Bandwidth Skimming* [32, 33] são particularmente interessantes porque, além de permitir uma redução significativa dos requisitos de banda, também oferecem acesso imediato ao serviço para os clientes. Esses resultados, economia de banda e serviço imediato, baseiam-se no mesmo princípio, comum aos dois protocolos: permitir que os clientes recebam dois fluxos simultâneos. O primeiro fluxo é exibido de imediato pelo programa cliente (o *player*) para o usuário (e é, justamente, o que permite aos protocolos o serviço imediato). O segundo fluxo é recebido por antecipação e armazenado em *buffer* para exibição futura. Esse fluxo permite que esse cliente *alcance* o cliente que já esteja recebendo o segundo fluxo. Quando o primeiro fluxo atinge o ponto em que o segundo começou a ser armazenado, diz-se que ocorreu a união dos fluxos. O primeiro fluxo pode ser desconectado, a exibição passa a ser feita a partir do *buffer*, e este segundo fluxo é compartilhado pelos clientes até o fim da exibição do arquivo. Uma diferença fundamental entre *Bandwidth Skimming* e *Patching* é que, enquanto o primeiro permite que as uniões de fluxos se organizem de forma hierárquica, criando uma árvore de uniões de profundidade arbitrária, o segundo restringe as uniões a hierarquias de dois níveis apenas. Além disso, diferentemente do *Patching*, variantes do *Bandwidth Skimming* podem ser usadas por clientes com banda de recepção inferior ao dobro da taxa de transmissão requerida para sustentar a

transmissão da mídia [32].

Ambos os protocolos têm sido extensivamente analisados, principalmente considerando-se cargas sintéticas *seqüenciais*, cargas nas quais os clientes emitem requisições para a exibição da mídia completa sem qualquer interrupção na exibição. Para tais cargas, ambos os protocolos têm se mostrado capazes de permitir economia significativa de banda para taxas variadas de requisições. A escalabilidade do *Bandwidth Skimming* é, porém, superior à do *Patching* [33], já que o primeiro permite hierarquias de uniões de múltiplos níveis.

Um alto grau de interatividade, por parte dos clientes, tem sido observado em muitas cargas reais de mídia contínua [2, 5, 7, 19–21, 25, 41, 63, 76, 78]. Em outras palavras, usuários de serviços reais de mídia contínua tipicamente *não* a acessam seqüencialmente. De fato, um usuário freqüentemente interrompe a execução, efetuando pausas, saltos (para frente ou para trás) e exibição acelerada para frente ou para trás. Esse comportamento é típico de usuários que requisitam vídeos educacionais, conforme foi observado em trabalho recente de caracterização de cargas de diferentes servidores de mídia contínua [25].

Uma questão fundamental que se coloca é: *como a escalabilidade dos protocolos de mídia contínua existentes é afetada pela presença de usuários interativos?* Esforços prévios no sentido de tratar essa questão são relativamente limitados e incluem a avaliação de protocolos com apenas poucas cargas reais [5, 21] e a derivação analítica de limites para os requisitos de banda, considerando-se apenas padrões de interatividade gerais [49, 74] e, portanto, não necessariamente realistas. Não obstante, esses estudos chegaram a uma mesma conclusão: a escalabilidade de protocolos é fortemente degradada pela presença de usuários interativos. As oportunidades de compartilhamento de fluxos são reduzidas já que as requisições que chegam para um mesmo vídeo próximas no tempo são, freqüentemente, para segmentos não sobrepostos. Com isso, a banda requerida pelos protocolos para cargas altamente interativas pode se aproximar muito da banda requerida pelo *Unicast*, transmitindo o mesmo dado múltiplas vezes, uma para cada requisição de cada cliente.

Alguns trabalhos prévios foram desenvolvidos no sentido de projetar protocolos especificamente otimizados para usuários interativos de mídia contínua. Em [55], os autores propõem um protocolo de *Patching* estendido. Sua escalabilidade, avaliada para padrões simplistas de interatividade, ainda não é conhecida para um número amplo de cenários realistas. Dois outros trabalhos [59, 68] propõem variações para o *Patching* orientadas para a interatividade. No primeiro caso, a avaliação do protocolo foi feita para cenários limitados. No segundo caso foi empregado um número limitado de cenários realistas. Assim, a escalabilidade dos protocolos de mídia contínua para usuários altamente interativos e com serviço imediato ainda é uma questão

em aberto e que demanda atenção.

Este trabalho investiga, através de abordagens complementares, o impacto da interatividade sobre a escalabilidade de protocolos de mídia contínua. Em primeiro lugar, identifica e faz a modelagem de parâmetros de cargas de mídia contínua que são chaves para a escalabilidade de protocolos. Essa modelagem é usada como base para construir um gerador de cargas sintéticas realistas. Esse gerador, associado a cargas de cinco servidores de mídia contínua reais, é usado para construir um rico conjunto de cargas sintéticas realistas, as quais cobrem diferentes domínios de aplicação (entretenimento, educacional, áudio e vídeo) e contêm variados graus de interatividade. Essas cargas são usadas em todas as simulações efetuadas neste trabalho. Como uma primeira abordagem para simplificar a análise de resultados, as cargas interativas são classificadas, usando os aspectos identificados como fundamentais para a escalabilidade de protocolos, como de interatividade alta, média e baixa.

Em segundo lugar, é feita uma avaliação experimental, por meio de simulação, do impacto da interatividade na escalabilidade de protocolos correntes de mídia contínua e dos resultados prévios para essa escalabilidade. São mostradas avaliações quantitativas para os protocolos *Bandwidth Skimming* e *Patching*. Essas avaliações, que cobrem uma região do espaço de projeto significativamente mais ampla que trabalhos anteriores, mostram que *Patching* degrada muito mais rapidamente que *Bandwidth Skimming* à medida em que a interatividade cresce. Os resultados experimentais são, ainda, usados para propor uma fórmula para a escalabilidade do *Bandwidth Skimming* para cargas de nível alto ou médio de interatividade a qual evidencia impacto de interatividade ainda mais pronunciado que o encontrado em [74].

Em terceiro lugar, esta tese define um novo conceito, o da *dispersão*, como um parâmetro único de cargas interativas que permite avaliar o impacto da interatividade na escalabilidade de protocolos de mídia contínua. O objetivo é simplificar a avaliação do impacto da interatividade sobre a escalabilidade de protocolos por meio de uma análise quantitativa, em lugar da análise qualitativa feita com a organização de cargas em diferentes níveis de interatividade. É mostrada uma forma geral para a dispersão. Também se mostra que a dispersão afeta diferentemente a escalabilidade de diferentes protocolos. Para avaliar o impacto da dispersão na escalabilidade de protocolos específicos, define-se o coeficiente de dispersão de um protocolo. Inicialmente é definido o coeficiente de dispersão para o *Bandwidth Skimming*, que é usado para estimar a escalabilidade desse protocolo. Também é mostrada uma definição de coeficiente de dispersão para o protocolo *Patching*.

Em quarto lugar, sendo a escalabilidade do *Bandwidth Skimming* também seriamente afetada pela interatividade alta, este trabalho propõe e avalia uma série de estratégias para reduzir

tal impacto. Embora sejam apresentadas como extensões ao *Bandwidth Skimming*, essas estratégias podem ser facilmente adaptadas a outros protocolos. As novas estratégias exploram características das cargas reais, tais como a alta localidade de referência [25], assim como características inerentes ao protocolo *Bandwidth Skimming* básico. Essas estratégias usam *buffers* localizados no cliente, a fim de reduzir o número de vezes que o servidor tem de transmitir o mesmo dado. Esta tese identifica um impacto significativo do uso de *buffers* localizados no cliente sobre a escalabilidade de protocolos de mídia contínua. Esse efeito, aqui denominado *fragmentação*, leva a um impacto degenerativo, tal que pode comprometer seriamente a escalabilidade de uma estratégia que use esse *buffer* local ao cliente para reduzir a retransmissão. Como resultado da fragmentação, o desempenho de uma estratégia varia de acordo com a forma de obtenção de conteúdo para armazenamento em *buffer*, e cada estratégia obtém maior economia de banda média de servidor para uma faixa diferente de dispersão. Como algumas das estratégias propostas requerem que o cliente escute fluxos em períodos nos quais ele estaria, de outra forma, inativo é feita uma avaliação do impacto dessas estratégias sobre a banda disponível para o cliente e também sobre a banda de rede.

Por fim, em quinto lugar, este trabalho apresenta um protocolo adaptativo que combina as estratégias previamente avaliadas, selecionando dinamicamente qual a técnica a empregar para uma certa carga. Esse protocolo adaptativo usa a dispersão para optar por uma das estratégias individuais e consegue, assim, evitar o efeito degenerativo da fragmentação. Como resultado, o protocolo adaptativo obtém economia de banda em uma faixa mais ampla de dispersões que qualquer estratégia individual.

Este trabalho centra seu foco na redução de banda média de servidor requerida para prover serviço imediato para usuários interativos. A análise da redução da banda média é de grande utilidade, uma vez que é aplicada às situações em que o servidor opera num nível de serviço estável (com a taxa de chegada de requisições estável). O foco na banda *média* apóia-se em dois fatos. Em primeiro lugar, os resultados podem ser comparados diretamente com a maior parte dos estudos anteriores, que também têm seu foco na banda média. Em segundo lugar, Tan *et al.* [75] mostraram que, para uma certa carga, devido a flutuações estatísticas nas taxas de requisição para múltiplos arquivos, um servidor provisionado com banda ligeiramente acima da banda média será capaz de atender essa carga, assumindo chegadas de requisições regidas por processo Poisson, como em [5, 25].

Este trabalho volta-se para *mídia contínua* e não para o *download* de arquivos de mídia. Em primeiro lugar, *mídia contínua* é mais eficiente para a entrega de objetos multimídia, uma vez que permite que os usuários examinem rapidamente se a mídia em transmissão é ou não

de interesse, abandonando-a, se for o caso, mais depressa e passando a examinar uma outra [40]. Segundo caracterização de carga efetivada por [40] com dados de diversos provedores de conteúdo, esse exame rápido, feito pelos usuários, ainda faz com que *download* e *pseudo-streaming* (uma modalidade de *download* na qual o cliente começa a exibir a mídia de imediato, mas sem as possibilidades de controle da exibição permitidas pela mídia contínua) levem a desperdício significativo de banda de rede. Nas medições efetivadas em [40], a banda de servidor requerida por *download* e a requerida por *pseudo-streaming* são, respectivamente, 2,25 e 1,75 superiores à requerida por mídia contínua. Em segundo lugar, protocolos de mídia contínua podem permitir compartilhamento de banda, com significativa economia em relação ao *download* de mídia. Essa economia é mais expressiva na mídia contínua, mesmo nos casos em que os usuários toleram latência de início de exibição relativamente longa [74]. Se, por um lado, é baixa a disponibilidade de *Multicast* ao nível de IP como base para protocolos de compartilhamento de fluxo, por outro lado uma série de estudos propõem ou avaliam o seu uso no nível de aplicação como forma de suprir sua baixa disponibilidade na Internet aberta, ou mesmo em redes corporativas [8, 9, 22, 35, 45, 47]. Por fim, embora sejam descritas como extensões ao *Bandwidth Skimming*, as estratégias aqui propostas podem ser aplicadas a qualquer protocolo de união hierárquica [11, 23, 59, 69].

As principais contribuições apresentadas por este trabalho são:

- a identificação de aspectos das cargas interativas que são fundamentais para a escalabilidade de diversas classes de protocolos de mídia contínua;
- a identificação da dispersão como o parâmetro chave de cargas de mídia contínua para a escalabilidade de protocolos de mídia contínua. A dispersão é definida como  $D = 1 - \left(\frac{P}{M}\right)$ , onde  $P$  é o potencial de compartilhamento de banda da carga (ou seja, a quantidade de mídia sobreposta existente entre requisições) e  $M$  é a quantidade de mídia entregue;
- um gerador de cargas realistas interativas;
- uma avaliação quantitativa mais abrangente do *Patching* e do *Bandwidth Skimming* para um número amplo de cargas sintéticas realistas interativas;
- uma quantificação mais precisa da escalabilidade do *Bandwidth Skimming*, por meio da obtenção experimental de nova expressão da escalabilidade de cargas de nível alto ou médio de interatividade;
- o uso da dispersão para avaliar a escalabilidade do *Bandwidth Skimming* com cargas interativas;
- a proposição de seis estratégias para reduzir ainda mais a banda requerida por usuários interativos, com economia de até 66% de banda de servidor e de até 40% de banda de

rede;

- a proposição de três estratégias híbridas, que exploram a combinação de estratégias individuais como forma de complementar a exploração do espaço de projeto de protocolos;
- uma avaliação quantitativa bastante completa das otimizações propostas com um número amplo de cargas realistas;
- um protocolo adaptativo que usa a dispersão para decidir qual estratégia deve ser aplicada a uma dada carga. Esse protocolo economiza até 66% da banda de servidor requerida para cargas interativas com respeito ao protocolo *Bandwidth Skimming* original, para uma faixa de taxa de chegada superior a qualquer estratégia individual ou híbrida aqui proposta.

O restante deste trabalho está assim organizado. O capítulo 2 apresenta trabalhos prévios nas áreas de protocolos de mídia contínua para cargas sequenciais e interativas, resultados concernentes à escalabilidade de protocolos de mídia contínua com cargas realistas, e, por fim, alguns resultados relevantes nas áreas de caracterização e síntese de cargas de mídia contínua. Os fatores das cargas de mídia contínua interativa com impacto na escalabilidade de protocolos de compartilhamento de banda são identificados no capítulo 3, no qual se definem a *dispersão*, fator de impacto primário sobre a escalabilidade de protocolos de mídia contínua com compartilhamento de banda, e seus coeficientes para *Patching* e *Bandwidth Skimming*. O coeficiente de dispersão para o *Bandwidth Skimming* é avaliado experimentalmente para 959 cargas de tipos variados. Nesse capítulo também se apresentam as cargas reais usadas neste trabalho, e discute-se o gerador de cargas sintéticas que, em conjunto com essas cargas, é usado para a construção de um rico conjunto de cargas sintéticas realistas. O impacto da interatividade sobre a escalabilidade de dois protocolos correntes de compartilhamento de banda é avaliado no capítulo 4. Novas técnicas para a redução do impacto da interatividade na escalabilidade do *Bandwidth Skimming*, e que podem ser aplicadas também a outros protocolos de compartilhamento de banda, são apresentadas e avaliadas no capítulo 5. Com a finalidade de explorar mais amplamente o espaço de projeto de protocolos, são também apresentadas estratégias híbridas que combinam as estratégias propostas previamente nesse mesmo capítulo. Algumas das estratégias propostas neste trabalho têm impacto sobre a banda de cliente e de rede. O capítulo 6 faz uma avaliação preliminar desse impacto, com o objetivo de explorar os principais compromissos envolvidos. O capítulo 7 usa os resultados dos capítulos 3 e 5 como base para o projeto de um protocolo adaptativo que monitora a carga e escolhe dinamicamente qual estratégia melhor se aplica a essa carga. Por fim, o capítulo 8 apresenta conclusões e sugere trabalhos futuros.



## 2 *Trabalhos relacionados*

Dada a importância da mídia contínua, diversos protocolos para sua transmissão com escalabilidade foram projetados ao longo dos anos. Uma revisão dos protocolos e conceitos de maior interesse para esta tese é apresentada na seção 2.1. Na seção 2.1.1 são analisados protocolos orientados para cargas seqüenciais. Concentra-se a atenção em protocolos orientados a requisições, por serem eles de especial aplicabilidade para a presente tese. Na seqüência, a seção 2.1.2 relata esforços prévios despendidos no sentido de desenvolver protocolos adaptados às cargas de mídia interativas. Por fim, a seção 2.1.3 registra resultados prévios para a escalabilidade de protocolos de mídia contínua com cargas interativas.

Na seção 2.2 faz-se uma revisão breve de trabalhos nas áreas de caracterização de cargas. A seção 2.2.1 apresenta alguns trabalhos na área de caracterização de cargas cujos resultados são relevantes para esta tese, e a seção 2.2.2 apresenta alguns trabalhos relevantes na área de modelagem e síntese de cargas. Embora o foco desta tese não se coloque sobre estes aspectos (caracterização, modelagem e síntese de cargas), a síntese de um amplo conjunto de cargas realistas de mídia contínua que cubra uma variedade de domínios de aplicação é crucial para a avaliação quantitativa a que esta tese se propõe, e que é uma contribuição da mesma.

### 2.1 Protocolos

Esta seção apresenta uma revisão de esforços prévios feitos no sentido de projetar protocolos de mídia contínua com escalabilidade.

#### 2.1.1 Protocolos de mídia contínua

Na transmissão em *Unicast*, cada usuário recebe a mídia num fluxo independente. A escalabilidade linear do *Unicast* pode, então, se tornar um sério limitador às aplicações de mídia contínua, especialmente se considerados os requisitos de banda cada vez mais crescentes das aplicações multimídia. Uma transmissão de televisão digital em alta definição (HD) no padrão

norte-americano, por exemplo, requer uma banda de 19,28 Mbps para transmissão terrestre e de 38,57 Mbps para transmissão por meio de cabo [58].

Dada a escalabilidade linear da transmissão *Unicast*, diversos protocolos de mídia contínua com escalabilidade que se apóiem no compartilhamento de fluxos têm sido propostos com o objetivo de obter grande redução de banda de servidor e de rede (comparados ao *Unicast*) [11, 12, 14, 15, 23, 30, 32, 33, 38, 43, 44, 55]. Uma visão geral de protocolos para mídia contínua e uma taxonomia dos sistemas de mídia contínua partindo dos sistemas nos quais o usuário é uma entidade passiva, sem controle sobre o conteúdo visualizado, até os sistemas efetivos de vídeo sob demanda, nos quais o usuário tem total controle sobre o andamento da exibição, podem ser vistos em [42, 56]. Dentre essa variedade de estratégias de transmissão de mídia, duas classes de protocolos são especialmente interessantes: *Bandwidth Skimming* e *Patching*.

Um primeiro esforço que resultou no projeto de um protocolo escalável foi o *Batching*, apresentado em [28]. A idéia fundamental desse protocolo é simples: agrupar os pedidos de exibição de mídia, de forma que diversos deles possam ser atendidos por um mesmo fluxo *Multicast*. Embora permita grande redução na banda requerida do servidor, o protocolo *Batching* apresenta uma restrição significativa: os pedidos não são atendidos de imediato e devem aguardar por algum tempo, até a chegada de um certo número de pedidos, sendo então todos eles atendidos pelo mesmo fluxo *Multicast*. Essa diferença de tempo entre o pedido do usuário para exibir a mídia e o tempo em que a exibição efetivamente se inicia é chamada *latência*.

A figura 2.1 exibe um esquema do funcionamento do *Batching*. No eixo  $x$  é mostrado o tempo e no eixo  $y$  a posição na mídia. Na figura, três requisições,  $r_a$ ,  $r_b$  e  $r_c$  (que não são mostradas na figura), chegam nos instantes  $t_a$ ,  $t_b$  e  $t_c$ , respectivamente. As requisições chegam

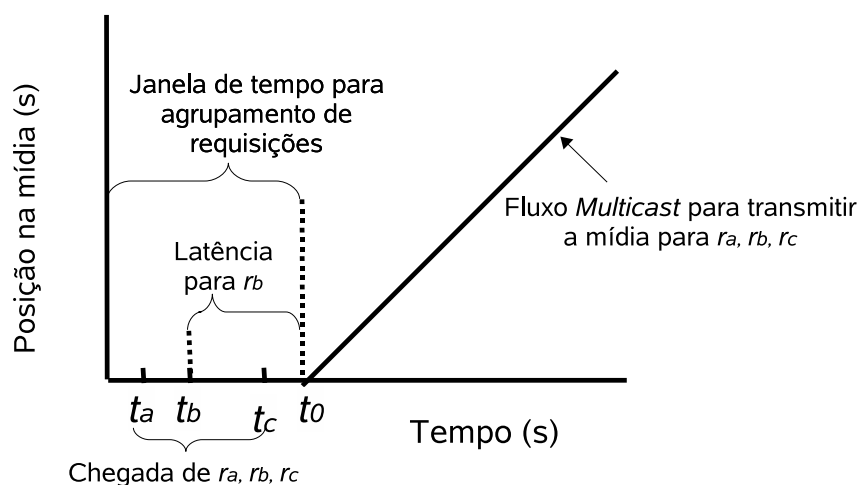


Figura 2.1: Esquema de funcionamento do protocolo *Batching*

durante a janela de tempo para agrupamento de requisições e não são atendidas de imediato. No instante  $t_0$  inicia-se a transmissão do fluxo *Multicast*, que atente todas as três requisições. Esse esquema geral é usado, com modificações, por diversos dos protocolos que são examinados a seguir.

Essa latência no atendimento às requisições para exibição de mídia, presente no *Batching*, deu ensejo ao desenvolvimento de técnicas que buscam o que se convencionou chamar *serviço imediato* [42]. Essas técnicas visam a atender com pouco ou nenhum atraso as requisições para exibição de mídia. Uma primeira técnica desenvolvida para permitir o serviço imediato foi a de *Piggybacking* [3, 39, 50]. Assim como no *Batching*, o *Piggybacking* busca agrupar requisições e atendê-las com um único fluxo *Multicast*. No caso do *Batching*, esse agrupamento é feito de forma estática. Uma vez emitido o pedido de exibição de mídia, o cliente deve aguardar algum tempo (pela possível chegada de mais clientes), até que se possa iniciar a recepção do fluxo *Multicast* que será compartilhado com os outros clientes do mesmo grupo (*batch*). Já no *Piggybacking*, o agrupamento é dinâmico. Cada novo cliente recebe um fluxo que tem uma taxa de exibição ligeiramente aumentada (a taxa deve ser aumentada, mas mantida abaixo de um limite dentro do qual o usuário não é capaz de perceber essa alteração). Após algum tempo de exibição, esse cliente terá, então, alcançado um fluxo já existente. Ocorrerá a união dos fluxos [39], e o fluxo acelerado poderá ser desconectado. A figura 2.2 exhibe um esquema de funcionamento desse protocolo. Nela pode-se ver que, quando da chegada de  $r_b$ , o servidor começa a retardar o fluxo  $f_a$  que atende  $r_a$ . O fluxo  $f_b$ , que atende  $r_b$ , é acelerado de forma a alcançar, após algum tempo,  $f_b$  o ponto da mídia em que  $f_a$  se encontra. Ocorre a união dos dois fluxos, e  $f_b$  pode ser desconectado.

Embora elimine a latência, essa técnica tem a desvantagem de exigir a alteração das taxas

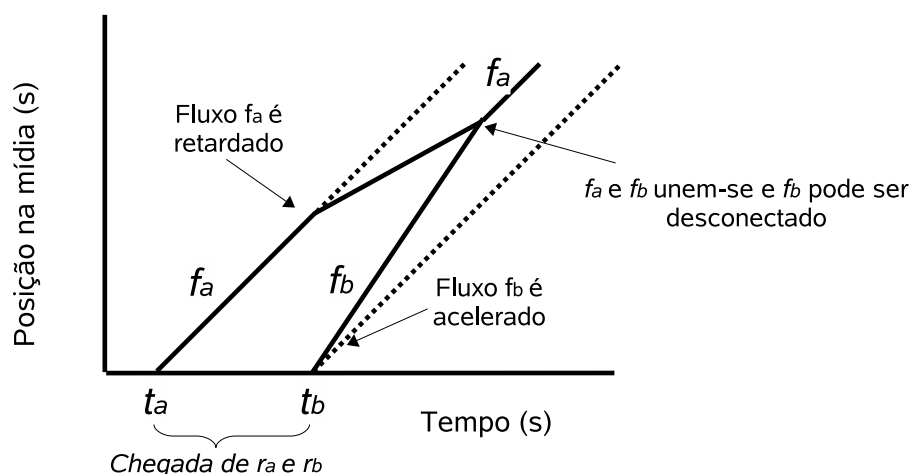


Figura 2.2: Esquema de funcionamento do protocolo *Piggybacking*

de exibição. A alteração não pode exceder a um limite acima do qual se torne perceptível, o que limita as oportunidades para união de fluxos. Contudo, embora a alteração de taxas de exibição imponha limites, a idéia de união de fluxos foi aproveitada em outras técnicas (descritas adiante) que não envolvem tal alteração.

*Stream Tapping* [16,17] é uma técnica que não requer alteração de taxas de exibição. Assim como *Piggybacking*, baseia-se na união de fluxos. No *Stream Tapping*, cada cliente recebe dois fluxos: um para exibição imediata, e outro para ser armazenado num *buffer* para exibição futura. Opera com *buffer* de tamanho variável e tem seu desempenho afetado por esse tamanho. A figura 2.3 mostra um esquema desse uso de *buffer*, e nela se vêem os dois fluxos recebidos pelo cliente (um para exibição imediata, outro para armazenar no *buffer*). A exibição alterna entre o fluxo para exibição imediata e o conteúdo do *buffer*. A viabilização da união de fluxos através de um *buffer* é também usada em outros protocolos.

*Patching* [14, 15, 43] é uma técnica também voltada para serviço imediato. No *Patching*, a chegada de uma requisição de um cliente dispara a criação de um fluxo *Multicast* para a entrega do objeto requisitado. Um cliente que chegue mais tarde junta-se ao fluxo *Multicast* que está em andamento e recebe, num fluxo *unicast* separado, o prefixo perdido. A figura 2.4 mostra um esquema deste protocolo, composto de três requisições  $r_a$ ,  $r_b$  e  $r_c$ . Com a chegada da primeira requisição,  $r_a$ , inicia-se a transmissão de um fluxo *Multicast*  $f_a$ , que transmite o objeto de mídia requisitado. Ao chegar a segunda requisição,  $r_b$ , inicia-se a transmissão de um fluxo *Unicast*  $f_b$ , que transmite o prefixo de mídia já transmitido por  $f_a$ . Enquanto a mídia recebida de  $f_b$  é exibida, a mídia recebida de  $f_a$  é armazenada no *buffer*. Quando  $f_b$  acaba de transmitir todo o prefixo de mídia que havia sido perdido pelo cliente, ocorre a união dos fluxos  $f_a$  e  $f_b$ . O fluxo

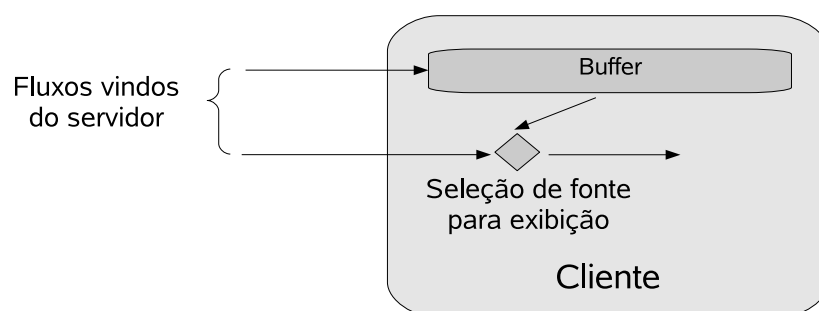


Figura 2.3: Esquema do uso de *buffer* para armazenar conteúdo a ser exibido no futuro

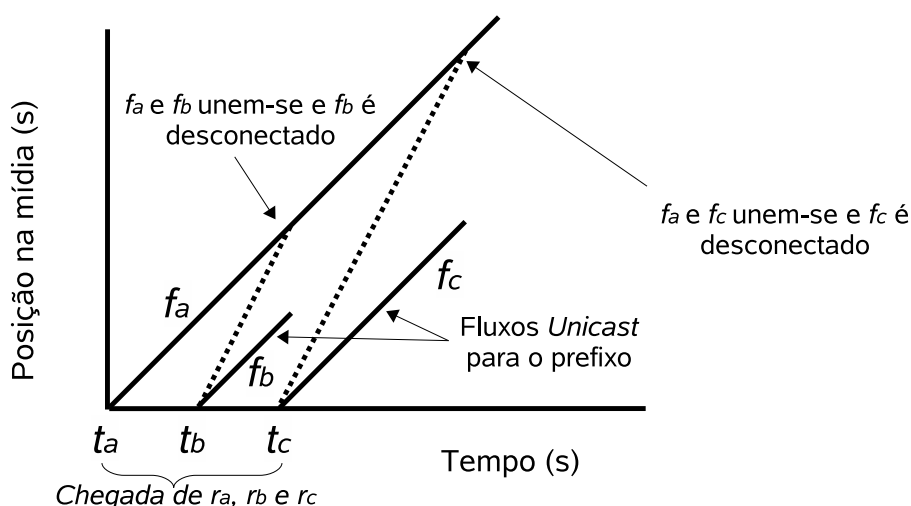


Figura 2.4: Esquema do protocolo *Patching*

$f_b$  pode ser desconectado, e o cliente segue recebendo  $f_a$ , que continua sendo armazenado no *buffer*. A exibição passa a ser feita a partir do conteúdo do *buffer*. Com a chegada da requisição  $r_c$ , processo similar é efetuado. As linhas tracejadas indicam o progresso acelerado do volume de dados recebidos por um cliente. O progresso é acelerado porque o cliente está recebendo dois fluxos. O progresso deixa de ser acelerado quando ocorre a união de fluxos e o cliente passa a receber um fluxo apenas.

Uma questão que se coloca é decidir quando se deve iniciar um novo fluxo *Multicast*. Em [43] são apresentadas duas políticas distintas. A primeira é o *Greedy Patching*, uma estratégia agressiva que somente cria um novo fluxo *Multicast* se não existir nenhum outro fluxo *Multicast* ativo. Essa estratégia pode levar à criação de um grande número de fluxos *Unicast* e, portanto, a um baixo compartilhamento de banda. A segunda política apresentada em [43] é o *Grace Patching*, que cria um novo fluxo *Multicast* quando o tamanho do prefixo perdido excede o tamanho do *buffer* do cliente. Um parâmetro denominado *janela ótima*, definido em [38], limita a banda requerida para fluxos *Unicast*: se o prefixo perdido é maior que o tamanho da *janela ótima*, o servidor cria um novo fluxo *Multicast* para transmitir todo o arquivo.

O protocolo *Bandwidth Skimming* [31–33] usa um princípio similar ao empregado pelo protocolo *Patching*, que é o de permitir que os clientes recebam até dois fluxos simultâneos, um para exibição imediata (e portanto serviço imediato), e outro para ser armazenado num *buffer* para exibição futura. No entanto, diferentemente do que ocorre com o protocolo *Patching*, tanto *Bandwidth Skimming* quanto *Piggybacking* utilizam a *união hierárquica de fluxos*. A figura 2.5(a) ilustra o processo de união hierárquica de fluxos, mostrando quatro fluxos,  $f_a$ ,  $f_b$ ,  $f_c$  e  $f_d$ , criados para atender as requisições  $r_a$ ,  $r_b$ ,  $r_c$  e  $r_d$ , respectivamente. Assim como na

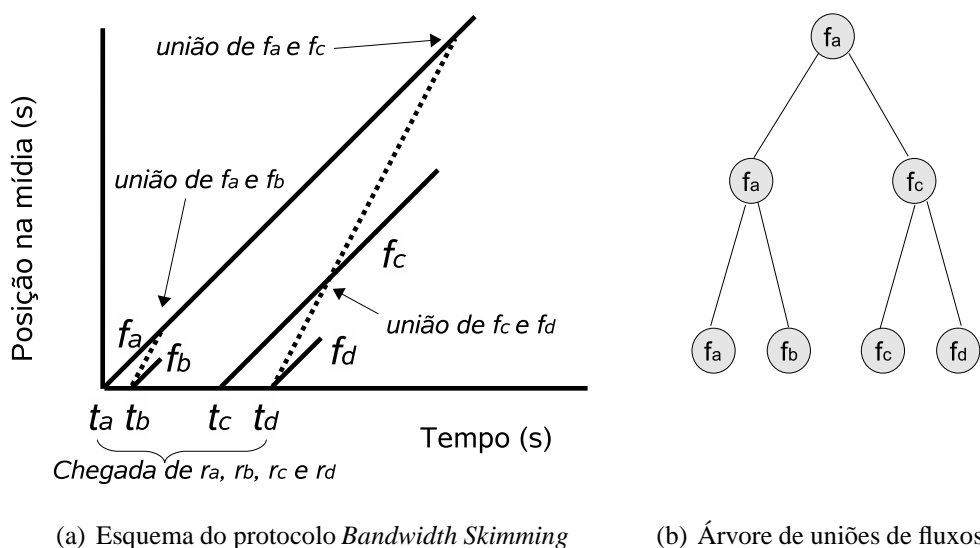


Figura 2.5: Protocolo *Bandwidth Skimming*

figura 2.4, as linhas tracejadas indicam o progresso acelerado de um cliente, o qual ocorre pela recepção simultânea de dois fluxos. A requisição  $r_a$  chega no instante  $t_a$ , e o fluxo *Multicast*  $f_a$  é criado para atendê-la. No instante  $t_b$  chega a requisição  $r_b$ . É criado o fluxo *Multicast*  $r_b$  para transmitir o prefixo perdido pelo cliente, que recebe também o fluxo  $f_a$ , armazenando-o em *buffer* para exibição futura (o fluxo  $f_a$  é dito *alvo* do fluxo  $f_b$ ). Quando esse prefixo é totalmente recebido, ocorre a união dos fluxos  $f_a$  e  $f_b$ , o fluxo  $f_b$  é desconectado, e o cliente passa a exibir dados do *buffer*. Esse *buffer* é conhecido como *merge buffer*. Procedimento similar ocorre com os fluxos  $f_c$  e  $f_d$  e, por fim, com os fluxos  $f_c$  e  $f_a$ , resultando num fluxo único  $f_a$ , o qual passa a ser compartilhado por todos os clientes. A árvore de uniões resultante é mostrada na figura 2.5(b).

Uma questão que se impõe é como criar árvores de uniões eficientes, que minimizem os requisitos de banda de servidor. Diversas técnicas são apresentadas em [31]. Em síntese, as principais características dessas técnicas são as seguintes:

**União Ótima de Fluxos:** Pode ser calculada para uma carga já conhecida, requerendo para isso métodos de programação dinâmica. Pode ser aproximada por uma política que, a cada chegada de requisição, calcule a árvore de custo mínimo necessária para os fluxos ativos até a chegada dessa requisição. Envolve custo cúbico em termos de execução e quadrático em termos de espaço. Detalhes acerca do custo de algoritmos de programação dinâmica podem ser encontrados em [24].

**Earliest Reachable Merge Target (ERMT):** Como o cálculo da árvore ótima não é prático, a política ERMT tenta unir os fluxos mais próximos, ou seja, os que levarão menos tempo

para se unir, considerando que não ocorrerá mais nenhuma chegada de requisição. Exige que se simule o andamento das uniões.

***Simple Reachable Merge Target (SRMT)***: Assim como a política ERMT, a política SRMT tenta unir os fluxos mais próximos, considerando que cada fluxo terminará após sua união com outro fluxo. É mais simples que ERMT, mas é menos efetiva.

***Closest Target (CT)***: Simplesmente escolhe o fluxo anterior mais próximo como alvo. É a política de implementação mais simples mas, ainda assim, tem desempenho muito próximo ao da ERMT e ao ótimo.

Embora outras variantes do *Bandwidth Skimming* [31–33] e da união hierárquica de fluxos, tais como o algoritmo diádico [23] e o algoritmo de árvore dinâmica [11], apliquem diferentes estratégias para a construção da árvore hierárquica de uniões, o algoritmo mais simples, *Closest Target*, mostrou-se capaz de obter banda média de servidor e de rede muito próximas ao ótimo [84]. Além disso, diferentemente do que ocorre com outros protocolos, *Bandwidth Skimming* pode ser aplicado mesmo quando a banda disponível no cliente é inferior a duas vezes a banda necessária para a transmissão de fluxos [32]. Nesta tese, usa-se a expressão *Bandwidth Skimming* para referência à variante *Closest Target* com banda igual ao dobro da necessária para a transmissão de um fluxo para o cliente. O funcionamento mais detalhado da política *Closest Target* é descrito a seguir.

Com a política *Closest Target*, o cliente escuta o novo fluxo *Multicast* e também o fluxo (alvo) *Multicast* mais próximo ainda ativo (o qual é chamado *closest target*). Quando o novo fluxo termina de transmitir todos os dados que o cliente perdeu (com relação ao fluxo alvo), o novo fluxo é encerrado e todos os clientes que estejam escutando o fluxo alvo se unem na escuta de um único fluxo. Os clientes do fluxo unido também iniciam a escuta do fluxo mais próximo ainda ativo, criando assim uma árvore hierárquica de união de fluxos. Contudo, se durante o andamento de uma tentativa de fusão, o fluxo alvo termina antes de ser alcançado, os clientes que o tinham como alvo simplesmente selecionam o alvo mais próximo disponível (o *closest target*). Todo o conteúdo recebido do alvo perdido, e que foi armazenado no *merge buffer* é descartado.

Tanto *Patching* quanto *Bandwidth Skimming* mostram-se capazes de obter grandes reduções nos requisitos de banda para cargas seqüenciais [12, 14, 15, 32, 33, 38, 43]. Para tais cargas, e com chegadas de requisições regidas por processo Poisson, a banda média de servidor requerida pelo *Patching* cresce com a raiz quadrada do número de requisições que chegam num período de duração igual ao da duração da mídia [33, 38]. Por outro lado, como o *Bandwidth Skimming*

cria árvores hierárquicas de uniões de fluxos, a banda média requerida pelo protocolo cresce logaritmicamente, com respeito ao número de chegadas acima descrito, mesmo para chegadas não Poisson [33].

### 2.1.2 Mídia contínua para usuários interativos

A preocupação com a interatividade, em muitos casos identificada como operações de VCR (Video Cassete Recorder ou, simplesmente, aparelho de vídeo cassete). faz-se presente no projeto de servidores para mídia contínua já há muitos anos, como se pode observar em trabalhos como [51, 61, 62]. Uma forma limitada de interatividade foi proposta por [10], em trabalho que previa o *broadcast* de diversas cópias do mesmo fluxo, de forma que a interatividade poderia ser oferecida, de forma limitada, pela simples troca de fluxos. Já em [27–29], usa-se a reserva de banda para canais de contingência destinados a atender as operações de avanço e retrocesso acelerados, com garantias estatísticas de que se possa atender a maioria dos usuários que requisitem essas operações. Em [18] são propostos esquemas de organização de disco voltados para atender avanço e retrocesso acelerados. Um esquema de alocação de banda, associado ao uso restrito de *buffers*, foi proposto em [37] para permitir operações interativas numa janela reduzida de tempo.

Um esquema usando transmissão *broadcast* é proposto em [6]. O sistema proposto agrupa pedidos de exibição de mídia e não fornece serviço imediato. Usa *buffer* de dimensões restritas como forma de fornecer operações interativas de duração prefixada. Uma operação de pausa, por exemplo, termina automaticamente ao fim de um período de tempo preestabelecido e ajustado conforme a configuração do sistema para a duração de pausas. Assim como outras propostas anteriores, usa canais adicionais para facilitar as operações interativas.

A técnica *Split-and-Merge*, proposta em [54], usa canais adicionais destinados a tratar as operações interativas e conta com uma estrutura de *buffers* compartilhados por diversos clientes para permitir que, após uma operação interativa, o cliente se desligue do fluxo adicional criado e volte a receber dados de um fluxo *Multicast* compartilhado.

Em oposição ao *Split-and-Merge*, que coloca o *buffer* num nó da rede para que possa ser compartilhado por vários clientes, no esquema proposto em [36] o *buffer* é colocado no cliente. Nessa proposta, um mecanismo de *broadcast* faz a distribuição dos vídeos. Um ponto interessante desse esquema é que ele carrega conteúdo no *buffer* antecipadamente, como forma de facilitar as operações interativas. Em [1] o *buffer* também é deslocado para o cliente. O protocolo *BEP - Best-Effort Patching* [55] retoma e estende o *Split-and-Merge*, buscando contornar deficiências, tais como *buffer* compartilhado colocado na rede e criação de número excessivo



de fluxos dedicados a atender as operações interativas.

A codificação de vídeos em uma forma que permite a sua recuperação em diferentes resoluções é usada em [52]. Esse armazenamento é empregado como base para a criação de um mecanismo capaz de oferecer as operações interativas de avanço e retrocesso acelerados.

O uso de *buffers* associado a fluxos de contingência é retomado em [17, 64]. O primeiro faz extensões ao *Stream Tapping* [16], e o segundo propõe um protocolo de *broadcast* próprio, denominado *Pagoda Broadcasting Protocol*.

A técnica *BIT - Broadcast-based Interaction Technique* é proposta em [77]. Essa técnica baseia-se no *broadcast* de uma chamada versão interativa da mídia, que nada mais é que uma versão comprimida da dessa mídia. Essa versão comprimida é usada como base para o atendimento às ações interativas.

*Chaining* [73] é um protocolo de *Multicast* que modifica o *Batching* de forma a reduzir a latência nos atendimentos de novos usuários. Esse protocolo foi modificado em [71] para permitir operações de interatividade.

Uma arquitetura que opera em dois níveis é proposta por [53]. Num primeiro nível, usa técnicas de *broadcast* para transmitir a mídia até um conjunto de *proxies*. Num segundo nível, usa conexões ponto-a-ponto (*Unicast*) entre *proxy* e cliente para transmitir a mídia. Tira partido do *buffer* existente no *proxy* para implementar operações de interatividade.

Em [45] é proposta a arquitetura *RM - Range Multicast*, uma modificação na arquitetura tradicional de protocolos *Multicast* (onde todos os clientes ligados a um canal recebem exatamente o mesmo fluxo). Na arquitetura *Range Multicast*, cada cliente pode receber um ponto diferente, ainda que próximo, do fluxo *Multicast*. Para tanto, o trabalho propõe a criação de uma estrutura *overlay* que faça o roteamento dessa nova forma de *Multicast*. Os nós de roteamento aos quais os clientes se conectam dispõem de *buffer* suficiente para permitir que cada cliente receba um ponto diferente do fluxo. Esse mesmo *buffer* é usado para permitir operações interativas.

Protocolos orientados a requisições, com serviço imediato, têm sido alvo de trabalhos recentes com foco na redução do impacto da interatividade. Otimização para o protocolo *Patching*, chamada *Patching Interativo - PI*, foi apresentada em [59]. Novas otimizações para o *Patching*, denominadas *Patching Interativo Eficiente - PIE* e *Patching Interativo Completo - PIC*, foram descritas em [68]. Uma otimização para o *Bandwidth Skimming*, denominada *Merge Interativo - MI*, foi apresentada em [69] e comparada às técnicas anteriormente propostas – *PI* [59], *PIE* e *PIC* [68] – e também ao *Bandwidth Skimming*. Além de comparações de banda de rede, fez-se nesse trabalho [69] uma avaliação da complexidade de tempo para a

execução dessas técnicas. Propostas para melhor uso de *buffer* são encontradas em [13].

Em conclusão, várias abordagens têm sido adotadas no sentido de melhorar a escalabilidade de protocolos de compartilhamento de banda para cargas interativas. Diversas técnicas empregam *buffer* para aprimorar o compartilhamento e reduzir a banda de servidor. No entanto, uma avaliação ampla do impacto na escalabilidade de protocolos, considerando uma variada gama de cenários e cargas, ainda não foi efetivada. Este trabalho tem como alvo fazer tal avaliação, como forma de subsidiar o projeto de estratégias para redução de banda de servidor.

### 2.1.3 Impacto da interatividade na escalabilidade de protocolos

Diversos estudos recentes identificam um alto grau de interatividade (pausas, saltos etc.), em várias cargas interativas reais [2, 5, 7, 13, 19–21, 25, 41, 63, 76, 78]. Alguns desses trabalhos mostram, por meio de simulação com algumas poucas cargas reais específicas, que a escalabilidade de protocolos de mídia contínua com compartilhamento de banda se reduz significativamente para essas cargas [5, 21]. Em outra linha de trabalho, estudos também recentes [49, 74] ocupam-se da derivação analítica da banda mínima de servidor requerida para usuários interativos com acesso imediato mostrando um grande acréscimo nos requisitos de banda de servidor, ocasionado pela interatividade, com relação à banda média previamente derivada para acesso sequencial [33]. Em [49], mostra-se que a escalabilidade da banda média requerida para protocolos de compartilhamento com cargas interativas é  $O(\sqrt{N})$ , onde  $N$ , a taxa normalizada de chegadas, é expressa como o número de requisições recebidas num período de tempo de duração igual ao da duração da mídia ( $N = \lambda \times T$ , onde  $\lambda$  é a taxa de chegada de requisições e  $T$  é a duração da mídia,  $N$  é a taxa normalizada com respeito à duração da mídia, de forma a estabelecer uma medida que possa ser usada para comparar mídias de durações diferentes). No outro trabalho [74], numa análise mais detalhada, conclui-se que a banda média requerida:

$$B_{\text{média}}^{\text{interatividade}}(N) = k\sqrt{\frac{2}{\pi}N} \quad (2.1)$$

Nesse resultado,  $k$  é uma constante de proporcionalidade de valor não especificado. Esse resultado confirma o forte impacto da interatividade sobre os protocolos de compartilhamento de banda com acesso imediato, já que a banda média requerida deixa de crescer logaritmicamente para crescer com a raiz quadrada. No entanto, fica em aberto o valor da constante  $k$ , o que limita a aplicabilidade desse resultado para trabalhos de planejamento de capacidade. Nesta tese é proposto um valor de  $k$ , aplicável ao protocolo *Bandwidth Skimming*, para cargas com baixa interatividade. Para esse mesmo protocolo, é proposta uma nova expressão que traduz

com mais precisão o crescimento da banda média requerida para cargas com nível médio a alto de interatividade.

Esses dois estudos baseiam-se em modelos um tanto arbitrários de interatividade e *não* em cargas reais (ainda que [74] tenha observações derivadas de cargas reais para motivar seus modelos de interatividade). Assim, a precisão das expressões de banda mínima ainda está por ser avaliada de forma mais efetiva. De forma geral, apesar da forte evidência do impacto da interatividade sobre a escalabilidade de protocolos de mídia contínua, uma avaliação ampla, com um grande número de cargas realistas, ainda está por fazer.

## 2.2 Cargas interativas de mídia contínua

Esta tese tem como alvo o projeto de estratégias que reduzam o impacto da interatividade na escalabilidade de protocolos de mídia contínua (ver seção 2.1.3). Para avaliar tais estratégias de forma abrangente, faz-se necessário o uso de um amplo conjunto de cargas de mídia contínua que cubra diversos domínios de aplicação. Na seção 2.2.1 são apresentados trabalhos de caracterização de carga que identificam e mensuram aspectos relevantes das cargas interativas. A identificação desses aspectos é fundamental para o projeto do gerador de cargas sintéticas realistas empregado nesta tese. Na sequência a seção 2.2.2 apresenta alguns trabalhos dedicados à criação de cargas sintéticas interativas.

### 2.2.1 Caracterização de cargas de mídia contínua

O sistema educacional de uma universidade norte-americana foi caracterizado em [63]. Foram encontradas distribuições para duração de sessões (sessão é uma seqüência de requisições de um usuário a uma mesma mídia), duração de requisições e duração dos períodos de inatividade entre requisições. Também foi calculada a frequência de operações de saltos e encontrando-se alguma localidade de acesso.

Acessos a vídeos de baixa resolução de um sistema educacional são avaliados em [41]. São obtidas distribuições para a duração de requisições e para os tempos de inatividade entre requisições. A avaliação dos saltos também indica alguma localidade de acesso.

A heterogeneidade do conteúdo de mídia contínua foi avaliada em [2], que identificou a presença de vídeos de duração e taxa de codificação variada nos acessos a um servidor de mídia contínua de uma universidade européia, além de uma forma simples de interatividade, a que chamou de acesso parcial. Nessa situação o vídeo é acessado a partir de sua posição inicial,

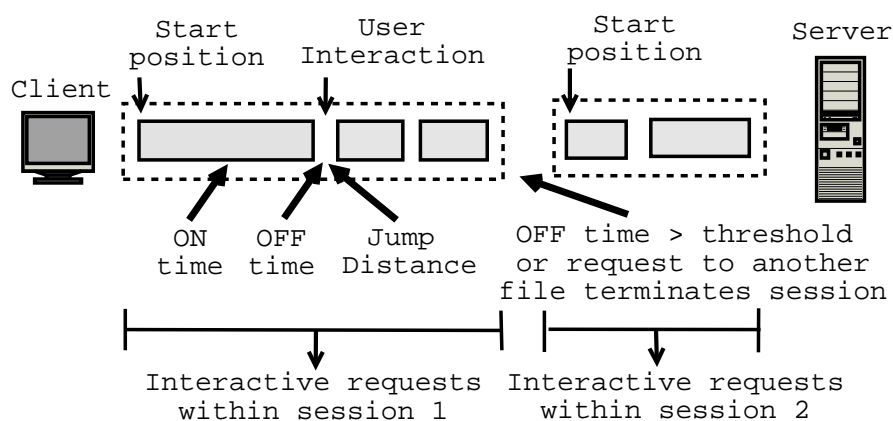


Figura 2.6: Modelo hierárquico do comportamento de clientes. Fonte: [25]

mas tem sua exibição interrompida. Avaliou-se também nesse estudo a popularidade de vídeos, verificando-se que um vídeo acessado recentemente tem alta probabilidade de ser acessado novamente em breve. A popularidade dos vídeos avaliados seguiu a distribuição Zipf [85].

Os acessos a serviços de mídia contínua feitos a partir de uma universidade norte-americana foram registrados e analisados em [21]. Ao medir características das sessões que acessam os mesmos arquivos, esse estudo identificou alta localidade temporal entre elas e concluiu que, para a carga em questão, seria proveitoso usar técnicas de compartilhamento de fluxos, com grande economia de banda.

A carga de sistemas educacionais de duas universidades norte-americanas foi usada em [5] para uma análise detalhada da interatividade, considerando-se um conjunto de requisições de um usuário a uma mesma mídia como uma sessão. Dentre diversos outros aspectos, tais como tamanho de arquivos, popularidade de arquivos e taxas de codificação, foram avaliadas as distribuições para tempo entre chegadas de requisições, duração de requisições e duração de períodos inativos entre requisições. Também foram avaliados quantidade de requisições interativas por sessão e percentuais de requisições interativas (pausa, saltos para a frente, saltos para trás). Identificaram-se diferentes padrões de acesso que variaram com a duração e a popularidade da mídia. Diferentemente de trabalhos anteriores, [5] identificou que a popularidade de arquivos é mais bem modelada pela concatenação de duas distribuições Zipf.

Análise de carga oriunda de uma empresa privada é feita em [19]. Esse trabalho concentra-se mais na análise da dinâmica de longo prazo do acesso a arquivos de mídia, ou seja, preocupa-se mais em avaliar como a popularidade de uma mídia varia com o tempo do que em analisar detalhadamente características tais como duração de sessões, acesso parcial ou localidade de referência. Essa mesma abordagem é retomada em [20].

Uma análise muito detalhada de cargas de mídia contínua, uma das mais completas existen-

tes na literatura, englobando diversos domínios de aplicação, foi efetuada em [25]. Diferentemente de trabalhos anteriores, essa caracterização não se ateve a cargas restritas ou específicas mas tratou de cargas de vídeo educacional, vídeo de entretenimento e áudio de entretenimento. Usou, também, cargas significativamente mais extensas que as usadas em caracterizações anteriores, cobrindo diversos perfis de interatividade. Avaliando diversos parâmetros das cargas e, especificamente, das sessões de usuários, identificou perfis de comportamento interativo típicos dos diversos tipos de cargas, de sorte que as características de uma sessão variam conforme o perfil. Verificou forte localidade de acesso em uma sessão, com a predominância de saltos curtos. Também identificou a existência de períodos de inatividade relativamente longos entre as requisições de uma sessão, se comparados à duração das requisições. Esse mesmo trabalho ainda pesquisou correlações entre os tipos de requisição interativa de uma sessão verificando que o tipo de uma requisição depende fortemente do tipo da requisição imediatamente anterior a ela.

A presente tese faz uso do modelo hierárquico de comportamento de clientes proposto em [25] e que é apresentado na figura 2.6. Esse modelo tem, num nível mais alto, uma sessão de cliente, que é quebrada, num nível inferior, em uma seqüência de requisições (ou ações) interativas. A sessão inicia-se com uma requisição que solicita a recuperação de um segmento do arquivo de mídia, a partir de uma certa posição inicial (*start position*). A requisição tem uma certa duração (*ON time*, ou período ativo), e, após a requisição ser atendida (ou interrompida), transcorre um tempo de inatividade (*OFF time*, ou período inativo) até o início da próxima requisição. A próxima requisição é disparada por uma ação interativa do usuário (reiniciar após uma pausa, efetuar um salto para a frente ou para trás, por exemplo). Uma sessão consiste numa série de requisições de acesso a uma mesma mídia e termina quando o cliente efetua uma requisição de acesso a outra mídia, ou após um certo período de inatividade.

Num estudo feito com carga coletada de diversos provedores de conteúdo sediados em um grande provedor de hospedagem, os autores de [40] fazem uma quantificação da perda de banda ocorrida em servidores de conteúdo que oferecem arquivos para *download* de mídia (em lugar de oferecer serviços de mídia contínua). Os autores indicam que a perda ocorre porque freqüentemente um usuário é levado a receber toda uma mídia apenas para examinar parte do conteúdo, processo que seria mais efetivo no caso de um serviço de mídia contínua.

Uma caracterização de grande volume de usuários de um serviço comercial de vídeo sob demanda é feita em [83]. Contudo, essa caracterização se volta mais para aspectos de longo prazo e para a popularidade de mídias, numa abordagem similar à de [19, 20].

Mais recentemente, foi publicada uma caracterização concernente a um sistema educacional

de uma universidade norte-americana [79]. Avaliando características das sessões dos usuários e os parâmetros das sessões interativas, essa análise buscou encontrar correlações entre esses parâmetros. Identificou-se uma correlação entre a posição inicial de uma requisição e sua duração, verificando-se também alta localidade de acesso nos saltos. Essa caracterização foi usada como base para a criação de um modelo da carga e para a síntese de carga a partir desse modelo.

Por fim, [7] faz a caracterização da carga de um sistema educacional brasileiro, que usa o servidor de vídeo RIO [60], um servidor de mídia contínua baseado em acesso aleatório [70]. Assim como em trabalhos anteriores, foram verificadas uma alta localidade de acesso em saltos e uma forte interatividade por parte dos usuários, conforme se poderia esperar em um servidor de vídeos educacionais.

### 2.2.2 Geração de cargas sintéticas de mídia contínua

A geração de carga sintética de mídia contínua interativa é essencial para uma avaliação ampla e abrangente da escalabilidade de protocolos, técnicas e estratégias dedicadas a esse tipo de carga. No entanto a disponibilidade de ferramentas nessa área específica é restrita, muito embora a modelagem de cargas interativas já tenha sido efetivada em diferentes níveis de detalhe. Nesta seção apresentam-se brevemente algumas dessas ferramentas de síntese de cargas de mídia interativa.

Uma primeira ferramenta de geração de cargas é o GISMO [48]. Usando distribuições de probabilidades com parâmetros extraídos de diversos trabalhos prévios de caracterização de cargas, o GISMO adota um modelo de acesso centrado numa sessão iniciada pela requisição de exibição de mídia feita por um usuário. Modela diversas características relevantes tais como o tempo entre chegadas de sessões, tamanhos de arquivos de mídia, popularidade de mídias, correlação temporal entre requisições e duração de ações interativas. É limitado quanto à modelagem de tempos de inatividade entre requisições. As distribuições usadas não foram levantadas de um conjunto amplo de cargas (como feito na ferramenta GENIUS, mostrada adiante).

A ferramenta MediSyn [76] foi desenvolvida a partir da caracterização de duas cargas de sistemas educacionais de uma grande empresa. O ponto forte da ferramenta é a modelagem das características de longo prazo das cargas de mídia contínua, tais como a introdução de arquivos novos, a variação da popularidade de uma mídia ao longo do tempo e os padrões de acesso diário. Faz uma modelagem restrita da interatividade, representando apenas acessos a prefixos de arquivos.

A ferramenta GENIUS (GENERator of INteractive User Sessions) [26] parte de uma carac-

terização de cargas significativamente mais ampla que as ferramentas anteriores [25], incluindo cargas de diversos domínios de aplicação (vídeo educacional, vídeo de entretenimento e áudio de entretenimento). Faz uma modelagem detalhada do comportamento interativo do usuário em uma sessão de acesso à mídia. Prevê diversas classes de objetos como forma de representar os diferentes perfis de comportamento identificados em [25]. Por fim, os parâmetros das distribuições de probabilidade empregadas pela ferramenta podem ser ajustados de forma a incorporar novas caracterizações de carga que venham a ser efetuadas. Para a validação do gerador, foram sintetizadas cargas com perfis variados e similares aos perfis de cargas reais. Dessa forma, foi possível comparar o impacto tanto das cargas sintéticas quanto das cargas reais na escalabilidade de protocolos de mídia contínua.

Em [79], além da caracterização da carga de um sistema educacional de universidade norte-americana, foi estabelecido um modelo que representa os dados caracterizados. Também foram realizados experimentos como forma de validar a carga sintetizada.

A caracterização de cargas efetuada em [7] foi usada como base para a proposição de um modelo, o qual foi parametrizado com os dados obtidos das cargas caracterizadas. O modelo foi validado, comparando-se o impacto das cargas sintéticas com o das cargas reais na escalabilidade de protocolo de mídia contínua.

Para este trabalho, as ferramentas GISMO e MediSyn foram consideradas inadequadas: o GISMO, por usar um conjunto de distribuições que não é derivado de um conjunto amplo de cargas reais, o que limita sua capacidade de gerar cargas realistas, e também por ser limitado quanto à representação dos tempos de inatividade entre requisições (pausas), uma característica marcante das cargas interativas reais [25]; já o MediSyn, por ser focado na modelagem das características de longo prazo das cargas de mídia contínua, características essas que não são aqui tratadas (ver seção 5.1), e por fazer uma modelagem limitada da interatividade, fator fundamental para esta tese.

A ferramenta GENIUS usa um modelo de sessões similar ao adotado nesta tese (ver figura 2.6). No entanto, optou-se, aqui, pela modelagem de trechos selecionados das diversas cargas reais empregadas, e que fossem considerados particularmente representativos delas. Nesse caso, o GENIUS tornar-se-ia de difícil aplicação, pois seria necessário calcular as diversas distribuições de probabilidade empregadas pela ferramenta para cada um dos trechos de carga selecionados. Se se considerar que para cada um dos trechos de carga empregados nesse trabalho (36 trechos, conforme mostra a seção 3.5), seria necessário calcular várias dezenas de distribuições. Portanto, em lugar do uso do GENIUS, optou-se pela criação de um novo modelo, capaz de calcular frequências e probabilidades diretamente a partir dos trechos selecionados

(ver seção 3.5), sem a necessidade do cálculo de distribuições.

Por fim, os dois últimos trabalhos relacionados, [7, 79], são mais recentes que a modelagem aqui efetuada. Uma comparação da qualidade das cargas sintetizadas por esses modelos com o das cargas sintetizadas pelo modelo aqui proposto é deixada como trabalho futuro.

**Conclusão:** Neste capítulo são revistos protocolos com compartilhamento de banda para a transmissão de mídia contínua. Esforços prévios no sentido de avaliar o impacto da interatividade, bem como suas limitações, são discutidos. São também revistos protocolos e variações de protocolos desenvolvidos no sentido de minimizar o impacto da interatividade. Fechando o capítulo, são mostrados resultados prévios de caracterização de cargas e síntese de cargas interativas. Esses resultados são fundamento para as etapas seguintes do trabalho: primeiramente como base para a síntese de um grande conjunto de cargas realistas; depois, para o uso dessas cargas na avaliação da escalabilidade de dois dos protocolos apresentado; e, por fim, como base para a elaboração de técnicas e estratégias para a redução do impacto da escalabilidade em protocolos de mídia contínua.



## 3 *Cargas interativas de mídia contínua*

Este capítulo aborda a interatividade em cargas de mídia contínua. Em primeiro lugar, o estudo sobre a interatividade efetivado neste capítulo é usado como base para a geração de um conjunto abrangente de cargas sintéticas realistas, necessário como fundamento para a avaliação quantitativa ampla, efetivada por meio de simulações, dos protocolos de mídia contínua estudados neste trabalho. Em segundo lugar, a compreensão dos fatores das cargas interativas que afetam a escalabilidade de protocolos de mídia contínua é usada como ponto de partida para estabelecer uma primeira classificação de cargas interativas, que simplifica significativamente a análise da escalabilidade. Essa compreensão é consolidada, neste capítulo, com a formulação do conceito de *dispersão*, primeiramente em duas dimensões (temporal e espacial) e, depois em uma dimensão apenas.

Inicialmente são identificados aspectos das cargas interativas de mídia contínua que são relevantes para a escalabilidade de protocolos (seção 3.1), estabelecendo-se uma primeira classificação de cargas de mídia contínua, conforme seu nível de interatividade. Na seção 3.2 é apresentada uma definição formal da dispersão. O impacto da dispersão sobre a escalabilidade de protocolos de compartilhamento de banda é usado na seção 3.3 para definir o coeficiente de dispersão dos protocolos *Bandwidth Skimming* e *Patching*. Na seqüência (seção 3.4) são apresentadas as cargas de mídia contínua usadas nesta tese. Por fim (seção 3.5), os aspectos identificados são usados para gerar uma ampla variedade de cargas sintéticas realistas, com graus diversos de interatividade, e que capturam as características das cargas reais mais relevantes para a escalabilidade dos protocolos e técnicas aqui avaliados.

### 3.1 **Dispersão temporal e espacial**

O objetivo desta seção é identificar um conjunto reduzido de aspectos fundamentais das cargas interativas e que são determinantes para a escalabilidade dos protocolos de mídia contínua. Inicialmente, note-se que, para *cargas seqüenciais*, a banda média de servidor requerida por um protocolo para atender a uma certa carga depende *apenas* da taxa normalizada de chegadas

$N$  [33, 38].

A situação é bastante mais complexa para cargas interativas, nas quais cada sessão de usuário consiste de um certo número de requisições interativas dirigidas a segmentos específicos da mídia selecionada. Nesse caso, os requisitos de banda média de servidor dependem não apenas de  $N$  mas também de diversos outros parâmetros, tais como número de requisições por sessão, posição inicial das requisições, duração das requisições, frequência de cada tipo de interação (pausa, salto etc.), período de inatividade entre duas requisições para a mesma mídia, e distância dos saltos (para frente ou para trás). Além disso, esses parâmetros podem variar grandemente conforme variem o tipo e o tamanho da mídia, e alguns deles, tais como a posição inicial e a duração das requisições, não são independentes [25]. Esses fatores transformam a exploração do espaço de projetos de protocolos de mídia contínua interativa numa tarefa bastante complexa.

A fim de compreender melhor o espaço de projeto de protocolos, é preciso identificar um conjunto reduzido de parâmetros, derivados da lista acima mostrada e que, capturando o impacto primário da interatividade na escalabilidade de protocolos, permitam distinguir o comportamento de protocolos e otimizações.

Numa primeira abordagem, o impacto das cargas interativas sobre a escalabilidade dos protocolos de mídia contínua é visto como sendo capturado por dois fatores preponderantes: a *dispersão temporal* e a *dispersão espacial* (uma segunda abordagem, complementar a esta, e que trata a dispersão em uma única dimensão, mais adequada a uma abordagem quantitativa, é vista na seção 3.2). A dispersão temporal captura o impacto da taxa de chegadas. Taxas de chegadas altas implicam em baixa dispersão temporal e, portanto, podem levar a uma maior probabilidade de uniões bem sucedidas. A dispersão espacial captura a sobreposição entre segmentos de mídias recuperados em duas requisições consecutivas (de um mesmo usuário, ou de usuários diferentes) para a mesma mídia. Em outras palavras, expressa a quantidade de mídia sobreposta entre requisições consecutivas. Baixa dispersão espacial pode aumentar a chance de uma longa sobreposição entre segmentos requisitados e, portanto, de uniões bem sucedidas de fluxos. Então, os aspectos de uma carga interativa que têm impacto fundamental na dispersão são a *taxa de requisições normalizada  $N$* , a *posição inicial das requisições*, e a *duração das requisições*<sup>1</sup>.

A figura 3.1 ilustra como esses aspectos afetam a dispersão temporal e a dispersão espacial e, como resultado, que impacto causam sobre a escalabilidade. A figura mostra quatro cargas, cada uma delas com dez requisições para a mesma mídia (com duração de 25 minutos) e durante

---

<sup>1</sup>Outros parâmetros das cargas são capturados pela posição inicial e pela duração das requisições. Por exemplo, a distância de salto entre duas requisições é dada pela posição final da requisição (calculada somando-se sua duração à sua posição inicial) e pela posição inicial da segunda requisição.

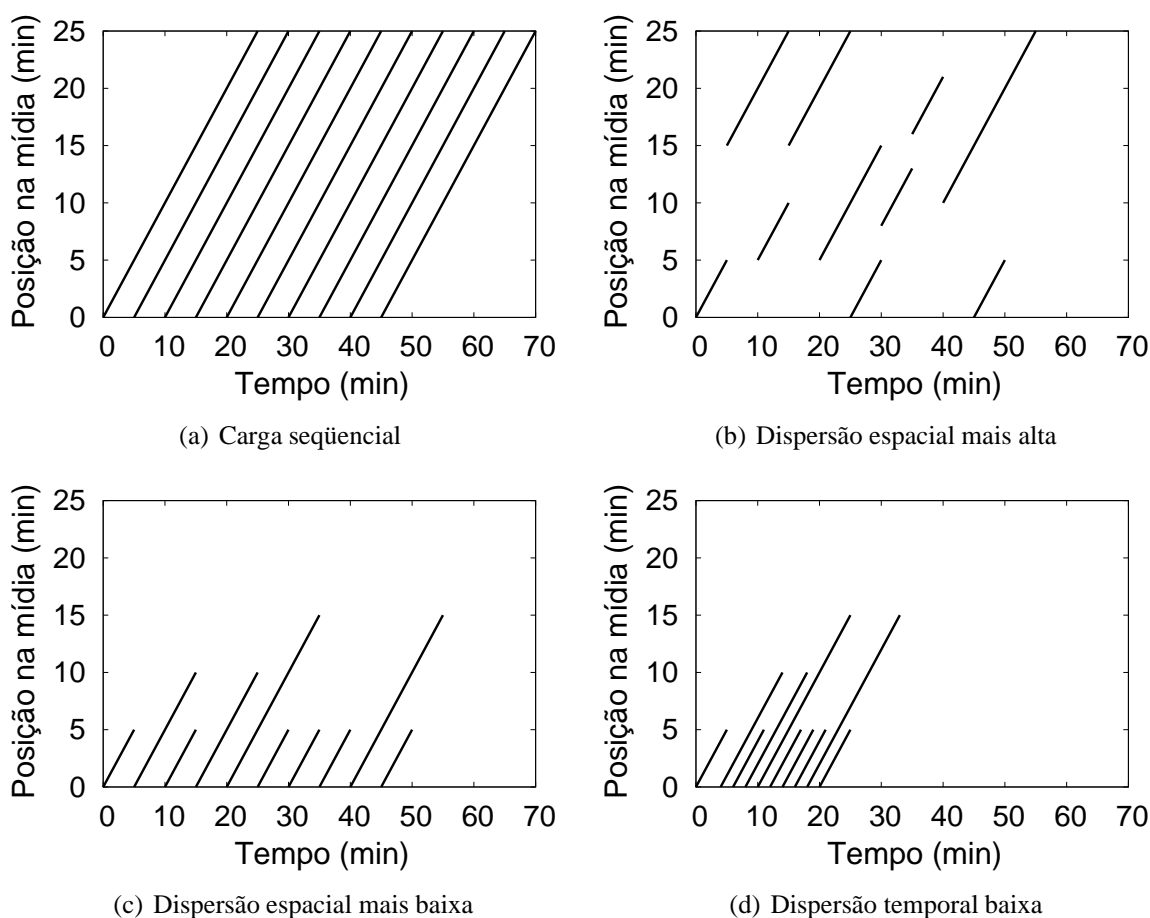


Figura 3.1: Dispersão em cargas de mídia contínua

o mesmo intervalo de tempo. Nos gráficos, cada linha representa uma requisição, com o eixo  $x$  marcando o momento de chegada da requisição, e o eixo  $y$  marcando as posições inicial e final, na mídia, do segmento requisitado. A figura 3.1(a) mostra uma carga seqüencial, com baixa dispersão temporal, sem dispersão espacial (como qualquer carga seqüencial) e, assim, com muitas oportunidades de uniões de fluxos. A carga na figura 3.1(b) tem a mesma taxa de chegadas  $N = 5$ , mas com variação na posição inicial e na duração das requisições e, assim, dispersão espacial mais alta. A figura 3.1(c) mostra uma carga com a mesma dispersão temporal, mas com dispersão espacial mais baixa, já que todas as requisições começam no início da mídia. Assim, há mais oportunidades para uniões de fluxos bem sucedidas na figura 3.1(c) que na 3.1(b). Por fim, a figura 3.1(d) mostra uma carga com a mesma dispersão espacial que a anterior, mas com dispersão temporal menor ( $N = 10$ ) e, assim, um número maior de uniões bem sucedidas.

A eficácia de diversas classes de otimizações e técnicas também pode ser afetada por outros fatores das cargas além da dispersão. Por exemplo, a *alta localidade* de referência em requi-

sições dentro de uma sessão de usuário, devida à alta frequência de pausas e saltos (frequentemente saltos curtos ou para trechos da mídia já exibidos), faz com que técnicas baseadas em *buffer* e na requisição antecipada de conteúdo – que será chamada *prefetching* – possam melhorar significativamente a escalabilidade dos protocolos. Mais especificamente, armazenar em *buffer* o conteúdo previamente requisitado apenas beneficia futuras requisições da mesma sessão que armazenou o conteúdo. Assim, o *número de requisições por sessão* também é relevante. Por fim, a eficácia das otimizações que se apóiam no *prefetching* de dados durante períodos de inatividade do usuário (durante pausas) é fortemente afetada pela *duração dos períodos de inatividade*.

Portanto, uma avaliação completa dos protocolos de mídia contínua para cargas interativas deve considerar os seguintes aspectos determinantes: taxa de chegadas, duração de requisições, número de requisições por sessão, localidade de referência e duração de períodos de inatividade.

A dispersão permite a classificação das cargas interativas em diferentes perfis de comportamento interativo e a representação do comportamento geral de cada perfil através de suas cargas típicas (o que simplifica a análise e a apresentação de resultados) [66, 67]. Nesta tese, as cargas foram classificadas em três grupos. Para estabelecer essa classificação, foram criados gráficos de escalabilidade para cada perfil de carga, similares aos apresentados na seção 5.3, e que apresentam a economia de banda média de servidor obtida por uma estratégia de otimização (ver seção 5.2) com respeito a um protocolo base (no caso, o protocolo empregado foi o *Bandwidth Skimming*). Os gráficos foram agrupados conforme a similaridade qualitativa da escalabilidade; em outras palavras, gráficos visualmente semelhantes foram agrupados numa categoria única. Por fim, foram levantadas características das cargas (posição inicial e duração de requisições, número de requisições por sessão) que refletiam esse agrupamento.

A avaliação do impacto da interatividade nos protocolos, efetivada por esta tese (Capítulos 4, 5), permite agrupar as cargas aqui empregadas (seções 3.4 e 3.5) segundo os seguintes níveis de interatividade:

**Interatividade Alta (IA):** Cargas com alta dispersão espacial. A duração média das requisições está abaixo de 20% da duração da mídia, e a posição inicial média está entre 30% e 60% da duração da mídia. Tipicamente, menos de 30% das requisições começam na posição inicial da mídia, e as sessões têm pelo menos três requisições. Vídeos educacionais longos têm, tipicamente, esse perfil.

**Interatividade Média (IM):** Cargas com nível intermediário de dispersão espacial. A duração média das requisições fica abaixo de 20% (assim como nas cargas IA), mas a posição inicial delas é mais concentrada (abaixo de 30% ou acima de 60% da duração da mí-

dia). Em geral, as sessões têm menos de três requisições. Vídeos de entretenimento têm, tipicamente, esse perfil.

**Interatividade Baixa (IB):** Cargas com pouca dispersão espacial. A duração média das requisições é mais longa (pelo menos 20% da média, em média), com a posição inicial fortemente concentrada no começo da mídia. Usualmente, têm-se cargas com menos de duas requisições por sessão. Cargas de áudio e de vídeos muito curtos (abaixo de 90 segundos) têm, tipicamente, esse perfil.

O agrupamento das cargas conforme o nível de interatividade simplifica enormemente a análise de resultados, uma vez que cargas de nível de interatividade similares apresentam resultados qualitativamente similares.

A localidade de referência e a duração de períodos de inatividade não são diretamente capturadas por essa categorização das cargas. Contudo, cabe notar que as cargas empregadas neste trabalho (reais e sintéticas) exibem alta localidade de referência, como observado em [25] para diversas cargas de mídia contínua interativa. Em média, 33% das interações são pausas, e 54% são saltos para trás. Além do mais, a distância média de salto está abaixo de 230 segundos. Também cabe notar que a duração média dos períodos de inatividade é de 286 segundos. Esses resultados motivam algumas das estratégias que são propostas no capítulo 5.

## 3.2 Definição formal para a dispersão

Na seção 3.1, a dispersão é apresentada de forma qualitativa, em duas dimensões: a dimensão espacial e a dimensão temporal. Essa abordagem qualitativa é usada em toda esta tese para classificar cargas e simplificar a análise de resultados. A presente seção ocupa-se da definição formal da dispersão e tem como alvo a sua medição em uma dimensão única.

Para definir a dispersão de uma carga, assume-se que todas as requisições  $r_k$  em  $R$  são para um mesmo objeto de mídia de duração  $T$  e que o tempo é discretizado e dividido em unidades de igual duração. Um *bloco* representa a menor quantidade de mídia que um servidor pode enviar para um cliente. Assumem-se blocos de um segundo de duração, mas qualquer outra unidade de tempo pode ser usada (ver seção 5.2). Blocos de uma requisição  $r_k$  são representados por  $r_k(t, p)$ , onde  $t$  é o instante no tempo em que o segmento se inicia e  $p$  é a posição de mídia requisitada. A notação usada é apresentada na Tabela 3.1.

A quantidade de vezes que a posição  $p$  da mídia é requisitada em toda uma carga é definida como  $Q_p$ . O potencial para compartilhamento,  $P$ , presente numa carga, ou seja, a quantidade de sobreposição presente na carga, é definido como:

$$P = \sum_{p=0}^{T-1} Q_p \quad (3.1)$$

Para medir  $D$ , a dispersão numa carga de mídia contínua, compara-se  $P$ , o potencial para compartilhamento de banda presente na carga (ou seja, a quantidade de sobreposição entre requisições), com o total de mídia entregue  $M$  ( $P$  não excede  $M$ ):

$$D = 1 - \left( \frac{P}{M} \right) \quad (3.2)$$

A sobreposição  $P$  pode variar desde 0, se não há nenhuma sobreposição entre as requisições, até a extensão completa das requisições (descontada a última requisição, que não se sobrepõe a nenhuma outra). Portanto, a dispersão varia no intervalo  $[0, 1)$ . Uma dispersão baixa implica em sobreposição mais longa e, como consequência, em maior potencial para compartilhamento de banda. Por outro lado, à medida que a dispersão aumenta, a quantidade de sobreposição decresce concomitantemente, o que também ocorre com o potencial para compartilhamento de banda também.

Diferentes protocolos usam diferentes algoritmos para selecionar quais requisições são candidatas a tentativas de união (e, portanto, quais são candidatas a compartilhar banda) e, por isso, exploram de forma diferenciada o potencial de compartilhamento de banda existente numa

Tabela 3.1: Notação

Símbolo	Definição
$D$	Dispersão de uma carga
$O$	Sobreposição entre requisições de uma carga
$M$	Total de mídia entregue
$T$	Duração da mídia
$R$	Seqüência de requisições para um objeto
$ R $	Número de requisições em $R$
$\Delta_{BS}$	Coefficiente de dispersão de uma carga transmitida com <i>Bandwidth Skimming</i>
$P_{BS}$	A quantidade de potencial para compartilhamento de banda de uma carga que o <i>Bandwidth Skimming</i> consegue explorar
$\Delta_{PA}$	Coefficiente de dispersão de uma carga na transmitida com <i>Patching</i>
$P_{PA}$	A quantidade de potencial para compartilhamento de banda de uma carga que o <i>Patching</i> consegue explorar
$r_k$	Uma das requisições à mídia ( $k = 1, \dots,  R $ )
$ r_k $	Duração da requisição $r_k$ ( $k = 1, \dots,  R $ )
$\delta_k$	Tempo entre a chegada de $r_{k-1}$ e a de $r_k$ ( $k = 2, \dots,  R $ )
$r_k(t, p)$	Um <i>bloco</i> de $r_k$ começando no momento $t$ e requisitando a posição da mídia indicada por $p$
$\phi_k$	O conjunto de blocos sobrepostos da requisição $k$ ( $k = 2, \dots,  R $ )
$\mu_k$	Quantidade de mídia em $\phi_k$ ( $k = 2, \dots,  R $ )

carga. É o que ocorre, por exemplo, com *Bandwidth Skimming* [32, 33], que tem escalabilidade proporcional a  $O(\log(N))$  com cargas seqüenciais [33], e com *Patching* [38], que tem escalabilidade proporcional a  $O(\sqrt{N})$  (com chegadas Poisson e cargas seqüenciais), e ambos apresentam ainda escalabilidade diferenciada também para cargas interativas (ver seção 2.1.3 e capítulo 4). Dentro desse quadro, esta tese tem como objetivo a medição do *impacto* da dispersão na escalabilidade de protocolos correntes de compartilhamento de banda.

### 3.3 Impacto da dispersão na escalabilidade dos protocolos *Bandwidth Skimming* e *Patching*

A seção 3.3.1 explora a medição do impacto da dispersão na escalabilidade do *Bandwidth Skimming* com cargas interativas, expresso pelo coeficiente de dispersão do *Bandwidth Skimming*. A medição desse coeficiente para cargas seqüenciais é abordada na seção 3.3.2. Já a seção 3.3.3 tem como foco o protocolo *Patching*. Por fim, a seção 3.3.4 mostra métricas para o coeficiente de dispersão de outros protocolos os quais podem ser medidos diretamente da carga e permitem estimar a banda média de servidor requerida pelo protocolo para a carga em questão.

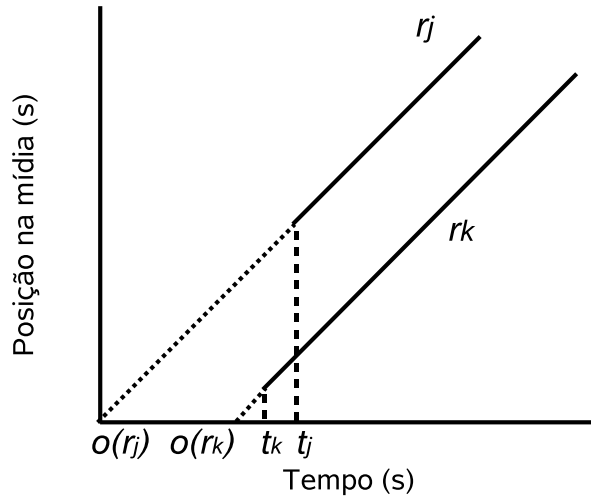


Figura 3.2: Extensão da chegada de requisições

### 3.3.1 Impacto da dispersão sobre a escalabilidade do protocolo *Bandwidth Skimming* com cargas interativas

Esta seção aborda o impacto da dispersão de uma carga de mídia interativa na escalabilidade do *Bandwidth Skimming* e com seleção de alvos segundo a política *Closest Target*. Para tanto define-se nos moldes da equação 3.2, o *coeficiente de dispersão* para o *Bandwidth Skimming*:

$$\Delta_{BS} = 1 - \left( \frac{P_{BS}}{M} \right) \quad (3.3)$$

onde  $M$  é a quantidade de mídia entregue, e  $P_{BS}$  é o potencial para compartilhamento de banda presente na carga o qual o *Bandwidth Skimming* consegue explorar.

Para medir  $P_{BS}$  seria necessário simular a execução do *Bandwidth Skimming* com a carga em questão, a fim de determinar quais segmentos de mídia são efetivamente compartilhados. Em lugar disso, esta tese faz uma estimativa de  $P_{BS}$  (uma avaliação quantitativa dessa estimativa é mostrada na seção 4.3), cujo cálculo não exige a efetiva execução do protocolo nem sua simulação. Essa estimativa é calculada comparando-se cada requisição  $r_k$  com cada requisição precedente  $r_j$  e determinando-se quais dos blocos poderiam, eventualmente, ser compartilhados pelo protocolo *Bandwidth Skimming*. Os blocos potencialmente compartilháveis são chamados *blocos sobrepostos*. Cada um dos blocos sobrepostos de  $r_k$  (sem repetição de blocos) é colocado no conjunto  $\phi_k$ , o *conjunto de blocos sobrepostos* da requisição  $r_k$  (conforme descrito a seguir). Usa-se  $\mu_k$  para representar a quantidade de mídia em  $\phi_k$  e, portanto,  $P_{BS} = \sum_{k=2}^{|R|} \mu_k$ .

Para comparar uma requisição  $r_k$  com as anteriores, é necessário que se defina um critério



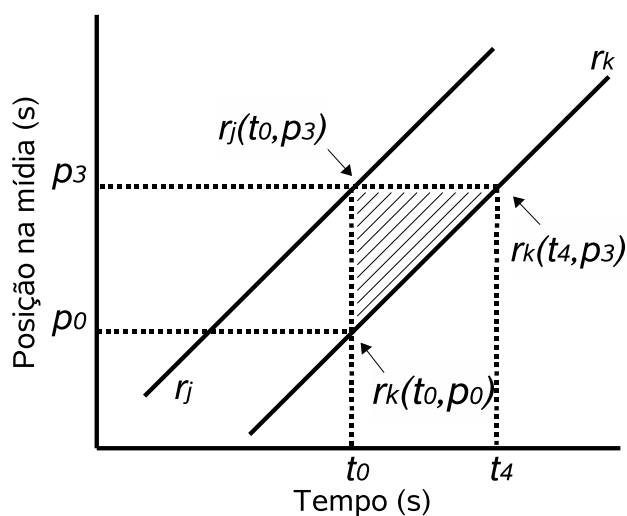


Figura 3.3: Duas requisições com sobreposição

de ordenação de requisições. Então, ordenam-se as requisições conforme  $o(r_k)$ , que é o momento em que a requisição, se fosse estendida para começar na posição inicial da mídia, teria chegado. Isso pode ser visto na figura 3.2, na qual as linhas grossas e as tracejadas representam, respectivamente, as requisições e suas extensões, e  $o(r_j) < o(r_k)$ . O momento da chegada de  $r_j$  é representado por  $t_j$ , e o da chegada de  $r_k$ ,  $t_k$ . Note-se que  $r_j$  é um alvo, no sentido do *Bandwidth Skimming* e com seleção de alvos pela técnica *Closest Target*, para  $r_k$ , mas  $r_j$  se inicia *depois* de  $r_k$ . No entanto, o momento em que a extensão de  $r_j$  corta o eixo  $x$  antecede o momento em que a extensão de  $r_k$  corta tal eixo. Finalmente, se duas requisições estendidas cortam o eixo  $x$  no mesmo momento, a ordenação é arbitrária.

As figuras 3.3 e 3.4 abordam a sobreposição entre duas requisições,  $r_j$  e  $r_k$ . Quaisquer blocos de  $r_j$  que sejam requisitados enquanto  $r_k$  já tenha se iniciado e que eventualmente venham a ser requisitados por  $r_k$  poderão, potencialmente, vir a ser compartilhados pelas duas requisições, se o protocolo *Bandwidth Skimming* com seleção de alvos *Closest Target* for empregado (as áreas hachuradas triangulares são usadas para evidenciar esse relacionamento entre os blocos). Portanto, esses blocos são incluídos no conjunto de blocos sobrepostos  $\phi_k$ . É o que acontece na figura 3.3, na qual o bloco  $r_j(t_0, p_3)$  é requisitado enquanto a requisição  $r_k$  já está ativa e requisitando o bloco  $r_k(t_0, p_0)$  e, mais adiante, requisitará  $r_k(t_4, p_3)$ . Portanto,  $r_j(t_0, p_3)$  e  $r_k(t_4, p_3)$  são sobrepostos e  $r_k(t_4, p_3)$  será incluído em  $\phi_k$ . A figura 3.4 mostra o primeiro e o último bloco do conjunto de blocos sobrepostos  $\phi_k$  (blocos  $r_k(t_2, p_2)$  e  $r_k(t_8, p_6)$ , respectivamente). A fim de manter a figura mais legível, são mostradas apenas as coordenadas dos blocos sobrepostos.

A figura 3.5 explora a sobreposição entre três requisições,  $r_i$ ,  $r_j$  e  $r_k$  (usando o *Bandwidth Skimming* com a política *Closest Target* como base). As requisições  $r_j$  e  $r_k$  são as mesmas já vistas nas figuras 3.3 e 3.4, enquanto  $r_i$  é uma requisição que se interpõe entre  $r_j$  e  $r_k$ . Mostram-se na figura 3.5 sobreposições entre  $r_j$  ( $r_j(t_1, p_2)$ ) e  $r_k$  ( $r_k(t_2, p_2)$ ), entre  $r_i$  ( $r_i(t_4, p_5)$ ) e  $r_j$  ( $r_j(t_2, p_5)$ ), e entre  $r_k$  ( $r_k(t_6, p_4)$ ) e  $r_i$  ( $r_i(t_3, p_4)$ ). Note-se que a quantidade de blocos de  $r_k$  que se sobrepõem a blocos de  $r_j$  é reduzida pela presença da requisição  $r_i$ , a qual está interposta entre as duas primeiras (uma vez que essa nova requisição altera a seleção de alvos no *Bandwidth Skimming*). Essa redução de sobreposição entre  $r_j$  e  $r_k$  ocorre porque o último bloco de  $r_k$  que pode ser, agora, compartilhado é  $r_k(t_7, p_6)$  e não mais,  $r_k(t_8, p_6)$  ( $t_7$  é anterior a  $t_8$ ). Em outras palavras,  $r_i$  reduz sobreposição entre  $r_k$  e  $r_j$  e, portanto, o potencial para compartilhamento de banda entre essas duas últimas requisições.

Em termos formais, na figura 3.5, os blocos  $r_k(t_2, p_2)$  e  $r_j(t_1, p_2)$  são considerados sobrepostos (para efeito do cálculo do coeficiente de dispersão do *Bandwidth Skimming*) porque:

1. ambos requisitam a mesma posição da mídia (a posição  $p_2$ );
2. o bloco  $r_j(t_1, p_2)$  é requisitado enquanto a requisição  $r_k$  está ativa e a posição  $p_1$  está sendo requisitada por  $r_k$  (bloco  $r_k(t_1, p_1)$ ); e
3. não há nenhuma requisição interposta entre  $r_j(t_1, p_2)$  e  $r_k(t_1, p_1)$ .

O conjunto de blocos sobrepostos de uma requisição pode, agora, ser expresso na forma mostrada a seguir. Sendo  $r_i, r_j$  e  $r_k$  requisições em  $R$  ( $j < i < k$ ), sendo  $t_a$  e  $t_b$  instantes no

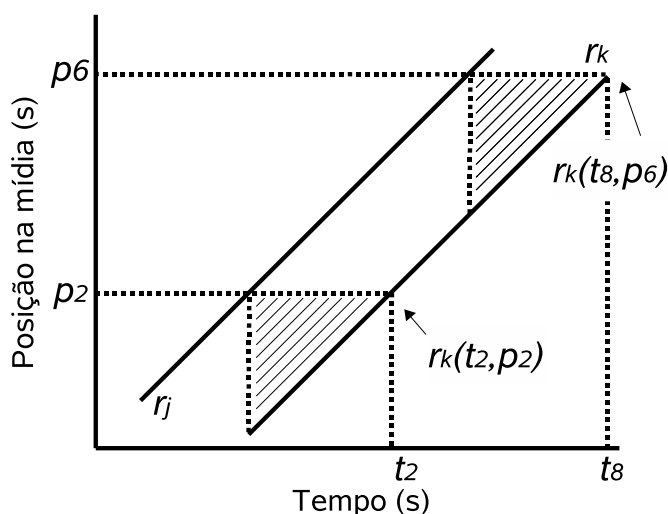


Figura 3.4: Blocos sobrepostos em duas requisições

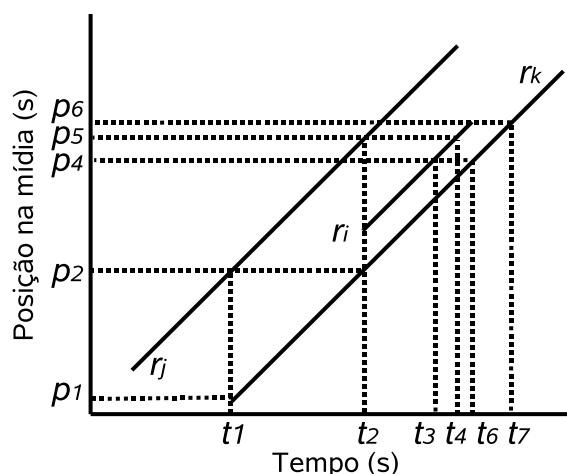


Figura 3.5: Impacto de uma terceira requisição

tempo e sendo  $p_a$ ,  $p_b$  e  $p_i$  blocos de requisições, o conjunto  $\phi_k$  é definido como se segue:

$$\begin{aligned} \phi_k = \{r_k(t_b, p_b) \mid & (\exists r_j(t_a, p_b)) \wedge \\ & (\exists r_k(t_a, p_a)) \wedge \\ & (\nexists (r_i(t_a, p_i) \wedge (p_a \leq p_i \leq p_b)))\} \end{aligned} \quad (3.4)$$

Definido  $\phi_k$ , o conjunto de blocos sobrepostos de uma requisição, e sendo  $\mu_k = |\phi_k|$ , tem-se, a partir da equação 3.3:

$$\Delta_{BS} = 1 - \left( \frac{\sum_{k=2}^{|R|} \mu_k}{\sum_{k=1}^{|R|} |r_k|} \right) \quad (3.5)$$

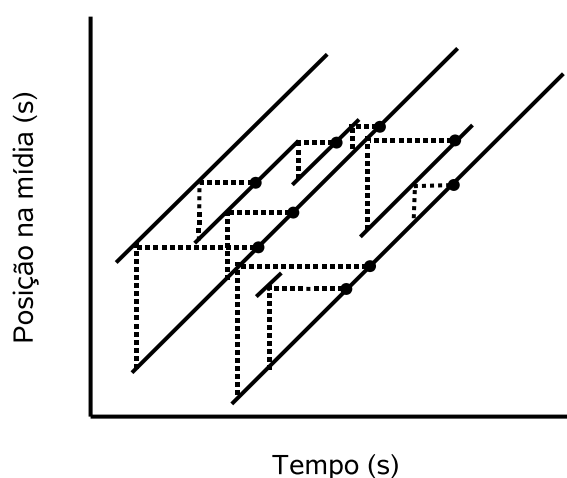


Figura 3.6: Sobreposição com sete requisições

Ao se aplicar essa métrica às cargas da figura 3.1, tem-se um pequeno coeficiente de dispersão de 0.28 para a figura 3.1(a), o qual cresce para 1 na figura 3.1(b) e decresce para 0.94 na figura 3.1(c) e novamente para 0.64 na figura 3.1(d), o que corresponde à visão intuitiva de que a dispersão cresce à medida que as requisições se tornam mais fragmentadas e espalhadas ao longo da mídia e decresce à medida que as taxas de chegadas aumentam.

Por fim, a figura 3.6 exibe uma carga com sete requisições e um certo número de blocos sobrepostos. As linhas tracejadas são usadas para marcar os blocos envolvidos numa dada sobreposição, sendo os blocos sobrepostos indicados com pontos. A figura mostra que a definição de blocos sobrepostos pode ser usada em cargas arbitrariamente complexas.

### 3.3.2 Impacto da dispersão sobre a escalabilidade do protocolo *Bandwidth Skimming* com cargas seqüenciais

Esta seção avalia o impacto da dispersão na escalabilidade do protocolo *Bandwidth Skimming* com cargas seqüenciais, com o objetivo de demonstrar a aplicabilidade do conceito de dispersão.

Trabalho anterior [33] mostra que, para cargas seqüenciais, a escalabilidade depende de um único fator, a taxa de chegadas  $N$ . Portanto, mostrar que a dispersão é um fator de impacto primário na escalabilidade do *Bandwidth Skimming* com cargas seqüenciais implica em mostrar que esse impacto é, de alguma forma, proporcional à taxa de chegadas  $N$ .

Para confirmar essa proporcionalidade entre a dispersão e a taxa de chegadas para o caso das cargas seqüenciais, inicia-se aplicando-se o resultado da seção anterior às cargas seqüenciais,

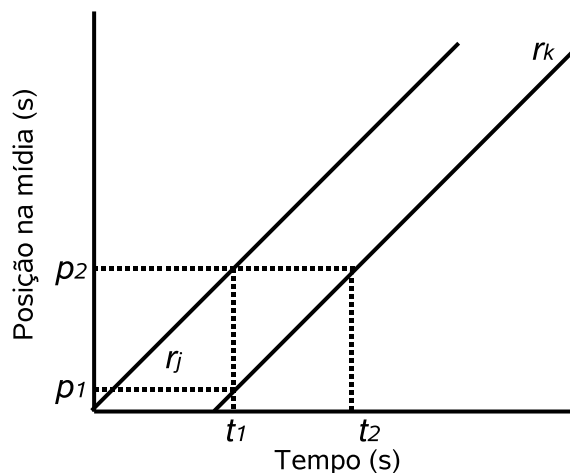


Figura 3.7: Sobreposição numa carga seqüencial

as quais são tratadas como um caso especial das cargas interativas (nesse caso especial, cada requisição requisita toda a mídia).

Para aplicar a definição de  $\phi_k$  a cargas seqüenciais, observa-se que, como todas as requisições de uma carga seqüencial têm a mesma duração, não é necessário fazer referência a requisições interpostas. Então, o termo da equação 3.4 que faz referência à sobreposição de segmentos pode se omitido, e a equação se reduz a:

$$\phi_k = \{r_k(t_a, p_b) \mid (\exists r_j(t_a, p_b)) \wedge (\exists r_k(t_a, p_a))\}$$

conforme ilustra a figura 3.7. Nessa figura exibem-se duas requisições e os blocos compartilhados,  $r_j(t_1, p_2)$  e  $r_k(t_2, p_2)$ ; outros blocos compartilhados não são mostrados, para simplificar a figura.

A figura 3.8 mostra o primeiro e o último blocos compartilhados entre  $r_k$  e  $r_j$  – que são  $r_k(t_c, p_a)$  e  $r_k(t_e, p_b)$ , respectivamente – ou seja,  $\phi_k$  é um conjunto que contém todos os blocos no intervalo fechado  $[r_k(t_c, p_a), r_k(t_e, p_b)]$ . Note-se que os blocos não compartilhados de  $r_k$  estão entre  $r_k(t_b, 0)$  e  $r_k(t_c, p_a)$  (esses blocos representam o prefixo recebido para serviço imediato, como pode ser visto na seção 2.1.1). Em outras palavras, sendo  $\delta_k$  o tempo entre a chegada de  $r_j$  e a de  $r_k$  ( $j = k - 1$ ) e sendo  $T$  a duração da mídia, a quantidade de mídia compartilhada entre duas requisições numa carga seqüencial é dada por:

$$\mu_k = \begin{cases} 0 & \text{se } T \leq \delta_k, k > 1 \\ T - \delta_k & \text{se } T > \delta_k, k > 1 \end{cases}$$

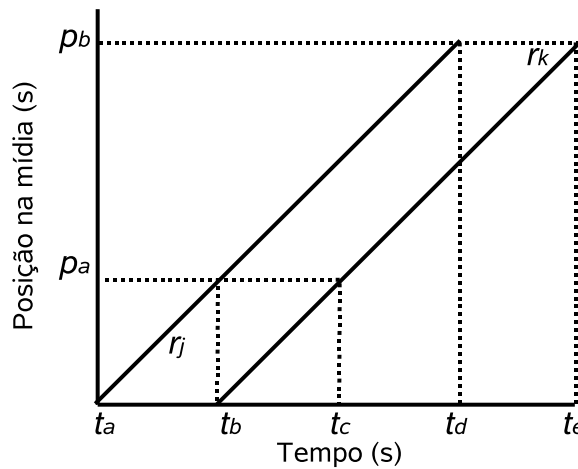


Figura 3.8: Primeiro e último blocos sobrepostos numa carga seqüencial

A seguir, como todas as requisições têm a mesma duração, o total de mídia requisitada é dado por  $M = |R| * T$ . Portanto, da equação 3.3 segue-se que o coeficiente de dispersão do *Bandwidth Skimming* para uma carga seqüencial é:

$$\Delta_{BS} = 1 - \left( \frac{\sum_{k=2}^{|R|} \mu_k}{|R| * T} \right)$$

Considerando-se chegadas de requisições por processo Poisson e taxa de chegadas  $N$ , tem-se que o tempo médio entre chegadas será  $T/N$ . Para taxas de chegadas mais altas, onde existam poucas requisições sem nenhuma sobreposição, a sobreposição entre duas requisições será, para  $k > 1$ :

$$\mu_k = T - \delta_k = T - T/N = T(1 - 1/N)$$

O total de sobreposição na carga será:

$$\sum_{k=2}^{|R|} \mu_k = (R - 1) * T(1 - 1/N)$$

Por fim, para  $R$  elevado, tem-se  $R \approx R - 1$ , e o coeficiente de dispersão do *Bandwidth Skimming* com cargas seqüenciais será:

$$\begin{aligned} \Delta_{BS} &= 1 - \left( \frac{\sum_{k=2}^{|R|} \mu_k}{|R| * T} \right) \\ &= 1 - \left( \frac{(R - 1) * T(1 - 1/N)}{|R| * T} \right) \\ &= 1 - (1 - 1/N) \\ \Delta_{BS} &= 1/N \end{aligned}$$

Dessa forma, conclui-se que, para *Bandwidth Skimming* com cargas seqüenciais, o valor do coeficiente de dispersão do protocolo é inversamente proporcional à taxa de chegadas  $N$  e pode, portanto, ser usado para o cálculo da banda média de servidor requerida pelo protocolo. Isso está de acordo com a visão intuitiva de que, numa carga seqüencial, a dispersão se reduz à sua dimensão temporal e pode, portanto, ser considerada fator de impacto primário na escalabilidade para esse tipo de carga.

Para taxas de chegadas mais baixas, essa medida superestima o potencial para compartilhamento de banda que um protocolo de união hierárquica de fluxos, tal como o *Bandwidth Skimming*, consegue obter. A figura 3.9 ilustra uma carga com cinco requisições, em que as linhas pontilhadas representam os segmentos que não seriam transmitidos se o protocolo *Bandwidth*

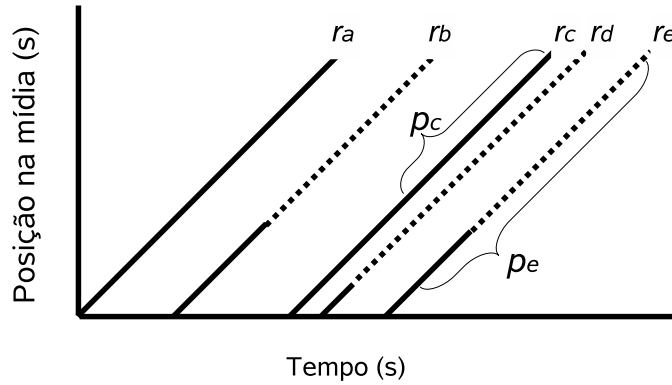


Figura 3.9: Sobreposição de requisições superestima a dispersão

*Skimming* fosse usado para transmitir as requisições. Assim,  $p_c$  superestima o potencial de compartilhamento de banda entre  $r_b$  e  $r_c$ , já que não há compartilhamento entre  $r_c$  e nenhuma das requisições precedentes. Contudo, à medida que a taxa de requisições aumenta (requisições  $r_c, r_d, r_e$ ), uniões de fluxos ocorrem logo após a chegada da requisição, e grande parte do conjunto de segmentos sobrepostos  $p_e$  pode ser realmente compartilhada. Embora aproximada, essa medida permite que sejam calculadas estimativas da banda média requerida pelo protocolo *Bandwidth Skimming* (ver seção 4.3). Além disso, permite que apenas dados que podem ser obtidos diretamente da carga sejam empregados, uma forma de medida bem mais simples, portanto, que a execução efetiva do protocolo, que seria necessária para a estimativa mais precisa da banda.

**Avaliação experimental:** Foi feita uma avaliação experimental do coeficiente de dispersão do *Bandwidth Skimming*. Para tanto, foram sintetizadas cargas com 100.000 requisições (de forma a atender à restrição de  $R \approx R - 1$ ) e chegadas Poisson (de forma a permitir que o tempo médio entre chegadas seja  $T/N$ ). Esses experimentos mostram que, a medida que  $N$  cresce, a diferença entre o  $N$  da carga sintética e o valor calculado como  $N = 1/\Delta_{BS}$  cai rapidamente. De fato, a diferença não excede a 0.37% para  $N \geq 5$ .

### 3.3.3 Impacto da dispersão sobre a escalabilidade do protocolo *Patching*

Esta seção apresenta uma métrica para o coeficiente de dispersão do protocolo *Patching*, e que é assim definida:

$$\Delta_{PA} = 1 - \left( \frac{P_{PA}}{M} \right) \quad (3.6)$$

onde  $M$  é a quantidade de mídia transmitida, e  $P_{PA}$  é o potencial para compartilhamento de banda de uma carga o qual o protocolo *Patching* consegue explorar.

Como o protocolo *Patching* foi projetado para trabalhar com cargas seqüenciais, faz-se necessário adaptá-lo para trabalhar com cargas interativas. A adaptação foi feita como em [67], e, como em [12], escolheu-se uma forma pouco intrusiva de extensão do protocolo. Essa adaptação estende o protocolo, aplicando a janela ótima [12] apenas para as requisições que se iniciam no primeiro bloco da mídia. Portanto, apenas esses fluxos terão a oportunidade de se unir a outros fluxos.

A fim de determinar  $P_{PA}$ , dividem-se as requisições  $R$  em dois grupos:  $R_m$  é o conjunto das requisições servidas por fluxo *Multicast*, e  $R_u$  é o conjunto das requisições que se iniciam sendo servidas por fluxo *Unicast*. Note-se que, para cada requisição  $r_u \in R_u$ , existe, no máximo, uma requisição  $r_m \in R_m$  servida por um fluxo *Multicast*, que é alvo de  $r_u$  e que, na adaptação do *Patching* adotada neste trabalho, ambas se iniciam na primeira posição da mídia.

Em segundo lugar, compara-se  $r_u$  com seu alvo  $r_m$ , selecionando os blocos de  $r_u$  que se sobrepõem a blocos de  $r_m$ . Esses blocos são colocados em  $\phi_u$ , o conjunto de blocos sobrepostos de  $r_u$ . Note-se que para cada  $r_u$  existe apenas um único alvo  $r_m$  (quando existe um alvo). Ambas as requisições se iniciam na primeira posição da mídia, mas o progresso de  $r_u$  em direção ao alvo  $r_m$  pode ser interrompido antes que  $r_u$  alcance  $r_m$  (em virtude de alguma ação interativa), e não ocorrerá compartilhamento algum. Portanto a quantidade de mídia sobreposta é dada por  $\mu_u = \min(|r_u| - (u_u - u_m), 0)$ , e, sendo  $P_{PA} = \sum^{r_u \in R_u} \mu_u$ , tem-se:

$$\Delta_{PA} = 1 - \left( \frac{\sum^{r_u \in R_u} \mu_u}{M} \right) \quad (3.7)$$

### 3.3.4 Métricas de impacto da dispersão sobre a escalabilidade de outros protocolos

As seções 3.3.1, 3.3.2 e 3.3.3 ocuparam-se de métricas para o coeficiente de dispersão dos protocolos *Bandwidth Skimming* e *Patching*. Esta seção oferece algumas indicações sobre a elaboração de métricas para o coeficiente de dispersão de outros protocolos.

Em primeiro lugar, nota-se que o aspecto chave para a medição do coeficiente de dispersão é a determinação do conjunto de blocos sobrepostos de uma requisição,  $\phi_k$ . Para determinar  $\phi_k$  é necessário ordenar as requisições e especificar quais delas podem, potencialmente, compartilhar conteúdo. Por fim, é necessário estabelecer uma regra capaz de estimar se um dado bloco de uma requisição será ou não compartilhado com um bloco de uma requisição anterior. Com esta



regra, podem-se determinar  $\phi_k$  e, a seguir,  $\mu_k$ .

Tabela 3.2: Sumário das cargas de vídeo (CV=Coeficiente de Variação)

Carga	eTeach	TV/UOL	MANIC
Período	02/09/00 a 17/10/01	18/01/02 a 01/03/02	07/02/01 a 24/01/04
Nº de arquivos únicos	230	42.439	745
Nº de requisições	46.958	1.453.117	25.833
Mídia armazenada (h)	80	1.303	683
Número médio de requisições/dia (CV)	114 (2,3)	33.793 (0,4)	24 (1,4)
Número médio de sessões/dia (CV)	31 (2,4)	27.718 (0,4)	8 (14,1)

### 3.4 Cargas reais de mídia contínua interativa

Neste trabalho são usadas cargas reais oriundas de cinco diferentes serviços de mídia contínua. Tais cargas incluem diversos perfis de comportamento interativo, que abrangem variados domínios de aplicações de mídia contínua, tais como cargas de áudio (com baixo nível de interatividade), vídeos curtos (de entretenimento e educacionais, ambos com nível médio de interatividade) e vídeos longos (educacionais, com alto nível de interatividade). São, portanto, um recurso que permite cobrir amplamente as características de cargas interativas que afetam significativamente a escalabilidade da banda média de protocolos e técnicas voltadas para a interatividade. Esta tese usa cargas já previamente caracterizadas em [25]. Adicionalmente, é usada outra carga educacional do sistema MANIC [65] de importante universidade norte-americana, a University of Massachusetts Amherst. A análise dessa carga levou a resultados muito próximos ao apresentado em [25] para cargas de vídeo educacional.

Tabela 3.3: Sumário das cargas de áudio (CV=Coeficiente de Variação)

se Carga	RADIO/UOL	ISP
Período	07/01/02 a 30/01/02	01/06/03 a 07/06/03
Nº de arquivos únicos	70.479	42.746
Nº de requisições	5.385.822	4.160.889
Mídia armazenada (h)	4.775	2.765
Número médio de requisições/dia (CV)	224.409 (0,2)	594.413 (0,2)
Número médio de sessões/dia (CV)	209.426 (0,2)	572.032 (0,2)

O sistema eTeach [34] é um sistema educacional da Universidade de Wisconsin-Madison

que opera desde o outono de 2000 e fornece conteúdo para aulas não presenciais. O conteúdo distribuído pelo serviço compreende desde anúncios em vídeos de até 5 minutos até aulas de duração variável, que podem se estender a até 55 minutos. Os estudantes acessam esse sistema, de dentro do campus da universidade, por meio de uma rede de alto desempenho.

O sistema MANIC [65] é usado para aulas presenciais e não presenciais. Na versão cuja carga foi aqui usada, o acesso ao serviço é feito diretamente na estação de trabalho do estudante por meio de um *CD-ROM*, com aulas de até 77 minutos de duração. Embora esse sistema seja usado a partir de *CD-ROM*, os perfis de acesso interativo, típicos de vídeos educacionais [25], também ocorrem nessa carga.

Os provedores UOL [80] e ISP oferecem conteúdo para milhares de usuários conectados à Internet. O serviço RADIO/UOL e o provedor ISP oferecem músicas (áudio com tamanho variável) e permitem que o usuário escolha diretamente a música a ser escutada e que se conecte a um canal de programação predefinida. O serviço TV/UOL oferece vídeos curtos com anúncios (até 1,5 minutos), *clips* de vídeo (1,5-5 minutos) e notícias, eventos e entrevistas (5-25 minutos).

Todas as cargas aqui empregadas tiveram suprimidas quaisquer informações sobre a identidade dos usuários. Um sumário das características das cargas pode ser visto nas Tabelas 3.2 e 3.3. As cargas oriundas de provedores de serviços de entretenimento apresentam volume de dados muito superior às cargas de serviços educacionais. Contudo, caracterização prévia e detalhada dessas cargas efetivada em [25] mostra que existem similaridades entre elas. Dessa forma, as conclusões e os resultados para uma carga podem ser estendidos para outras. Além disso, a interatividade é mais expressiva nas cargas educacionais. Em virtude desses aspectos, o uso de cargas educacionais na exploração da escalabilidade de protocolos de compartilhamento de banda é essencial.

A figura 3.10 mostra perfis típicos de comportamento interativo e que estão presentes nas cargas reais aqui empregadas. Cada um dos gráficos nessa figura apresenta um conjunto de requisições interativas emitidas para uma mesma mídia ao longo de um certo período de tempo (o período de tempo não é representado no gráfico). As requisições são ordenadas no eixo  $x$  pela sua posição inicial e, em caso de empate, pela sua duração. No eixo  $y$  têm-se as posições, na mídia, em que uma requisição se inicia e termina. A linha mais grossa indica a posição inicial das requisições, e a linha mais fina, a posição final. Então, para cada posição no eixo  $x$ , têm-se, no eixo  $y$ , a posição inicial e a final de uma requisição.

As figuras 3.10(a), 3.10(b) e 3.10(c) mostram perfis contidos em cargas reais. Nas cargas com nível alto de interatividade (figura 3.10(a)), vê-se que a posição inicial é mais distribuída ao longo da mídia que nos outros perfis de acesso (figuras 3.10(b) e 3.10(c)). Na figura 3.10(a),

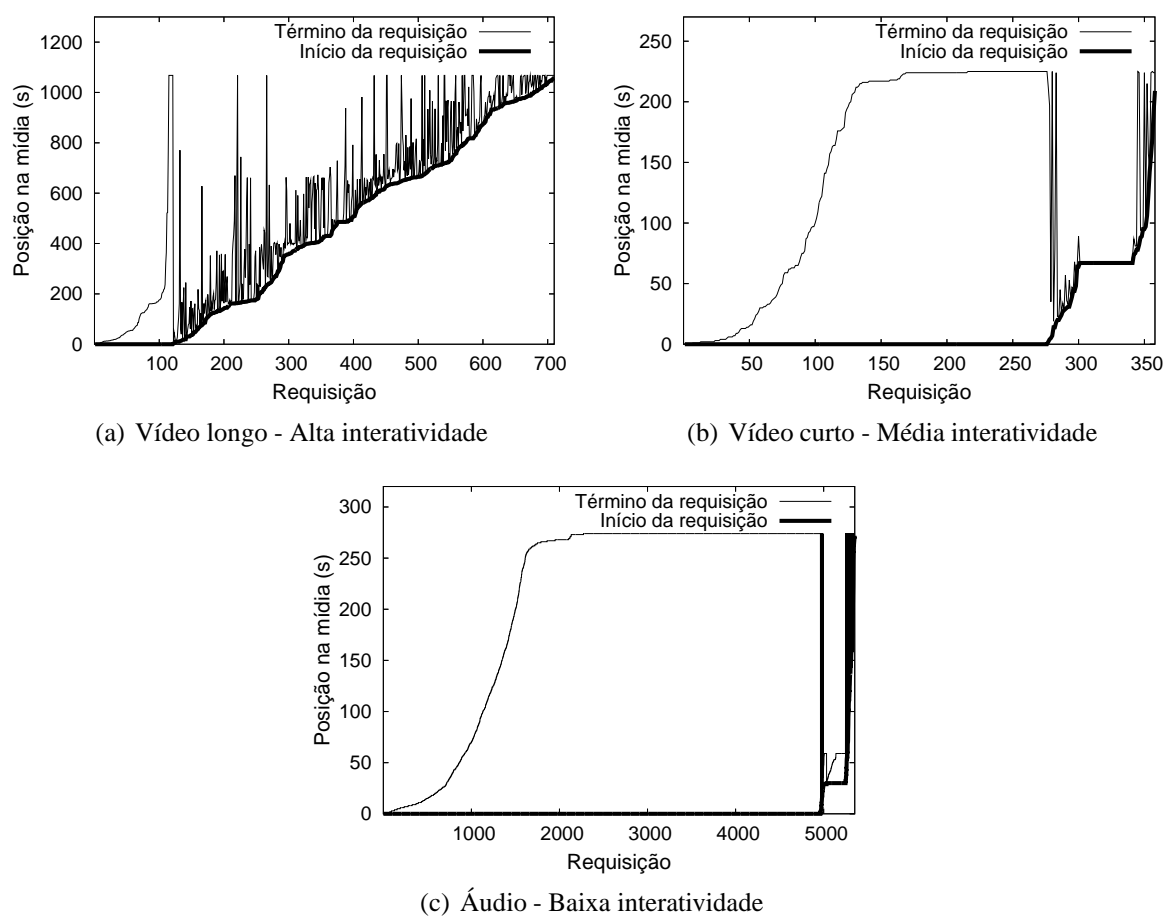


Figura 3.10: Perfis de comportamento interativo para cargas reais

existem pouco mais de 100 requisições que se iniciam na posição 0 da mídia. Menos de 20% delas iniciam-se na posição 0. A duração das requisições é bastante variável, e poucas requisições percorrem completamente a mídia. Cargas com esse perfil têm maior número de requisições interativas por sessão de um cliente que cargas com outros perfis. Nesse perfil, característico de vídeos de longa duração (acima de 10 minutos), há uma ocorrência maior de saltos que nos outros.

Nas cargas com nível médio de interatividade (figura 3.10(b)), a posição inicial é mais concentrada no início da mídia que nos perfis de acesso com nível alto de interatividade (figura 3.10(a)). Na figura 3.10(b), existem mais de 250 requisições que se iniciam na posição 0 da mídia. Mais de 70% delas iniciam-se na posição 0 da mídia. Contudo, a posição inicial das requisições ainda é mais distribuída que nos perfis de acesso com baixo nível de interatividade (figura 3.10(c)). Muitas requisições percorrem toda a mídia, ou grande parte dela, e a duração das requisições é menos variável que no caso de perfis de interatividade alta (figura 3.10(a)). Como muitas requisições percorrem toda a mídia, o número de requisições por sessão cai com relação aos perfis de interatividade alta. Esse perfil é característico de vídeos de curta duração (abaixo de 5 minutos), e nele há uma ocorrência de saltos menor que nos perfis de interatividade alta.

Nas cargas com nível baixo de interatividade (figura 3.10(c)), a posição inicial é mais fortemente concentrada no início da mídia que em qualquer dos perfis anteriores. Na figura 3.10(c), existem quase 5.000 requisições que se iniciam na posição 0 da mídia. A maior parte das requisições percorre toda a mídia, ou grande parte dela. Com isso, a duração das requisições é maior e tem menor variabilidade que nos perfis anteriores. O número médio de requisições por sessão também cai com relação aos perfis anteriores, uma vez que grande parte das requisições percorre toda a mídia. Esse perfil é característico de arquivos de áudio.

### **3.5 Síntese de cargas realistas de mídia contínua interativa**

Embora as cargas reais apresentadas na seção 3.4 incluam um conjunto amplo e variado de perfis de comportamento interativo, a avaliação quantitativa da escalabilidade necessita de um grande conjunto de cargas de perfil similar e que cubra uma ampla faixa de taxas de chegadas de requisições. Esta seção descreve como as características das cargas reais são modeladas e como esse modelo é usado com a finalidade de gerar um rico conjunto de cargas sintéticas realistas.

Foi criado um gerador de cargas sintéticas interativas que usa, como entradas, um conjunto selecionado de sessões de acesso a uma dada mídia (seção 3.1) – esse conjunto é chamado um

*trace* real – e uma taxa de chegada de sessões e que produz, como saída, um *trace* sintético com padrões interativos similares aos do *trace* real original. Diferentes cargas sintéticas podem ser produzidas a partir do mesmo *trace* real ao se variarem as taxas de chegadas de sessões e/ou ao se variarem as sementes dos geradores de números aleatórios empregados para as diversas variáveis aleatórias. Note-se que a taxa de chegada de requisições  $N$  de uma carga sintética é função da taxa de chegada de sessões e do número médio de requisições por sessão.

O gerador constrói, para um dado *trace* real de entrada, um modelo probabilístico de transição de estados (um modelo de histograma). Esse modelo representa o comportamento típico de um usuário em uma sessão interativa e é usado para gerar uma série de sessões. O modelo probabilístico é representado por meio de um diagrama de estados. Para reduzir o número de estados no modelo, a mídia é dividida em segmentos de 10 segundos. Com a varredura do *trace* real determina-se quais são os estados, quais as transições possíveis a partir de cada estado, qual a duração média de cada estado (para todos os estados que não têm a duração fixa de 10 segundos) e qual a frequência de cada transição. A frequência é usada para calcular a probabilidade de cada transição.

Nos *traces* sintéticos gerados, assume-se que o processo de chegadas de sessões é Poisson [5, 25]. O comportamento do usuário em cada sessão, incluindo o número de requisições por sessão, as distâncias de saltos e os períodos de inatividade durante a pausa, é extraído do perfil de carga. Nos *traces* sintéticos, as durações das pausas, assim como as dos segmentos parcialmente exibidos, são exponencialmente distribuídas com média igual à duração dos estados correspondentes (note-se que, com pausas com duração exponencial, torna-se pouco provável gerar-se um estado de pausa com duração exatamente 0). Um esquema geral do diagrama que omite, por simplicidade, tempos e frequências é mostrado na figura 3.11. Todas as probabilidades de transição são calculadas a partir das frequências relativas medidas na carga. Uma transição que nunca ocorra na carga não será representada no diagrama. Os tipos de estados presentes no diagrama do modelo são:

1. *Início de Sessão – SOSession*: é o estado inicial do diagrama e não tem duração associada. A ele sempre se segue a exibição de um segmento (*Segmento Comum* ou *Fim de Requisição*);
2. *Segmento Comum – Círculos*: Representa um segmento que é exibido completamente e, então, é seguido por outro. A duração desse estado é fixa e depende unicamente da duração de um segmento (usam-se 10 segundos). Esse estado pode ser seguido por outro estado *Segmento Comum* (caso em que a requisição continua) ou por um estado *Fim de Requisição* (caso em que a requisição termina). Uma requisição longa percorrerá uma

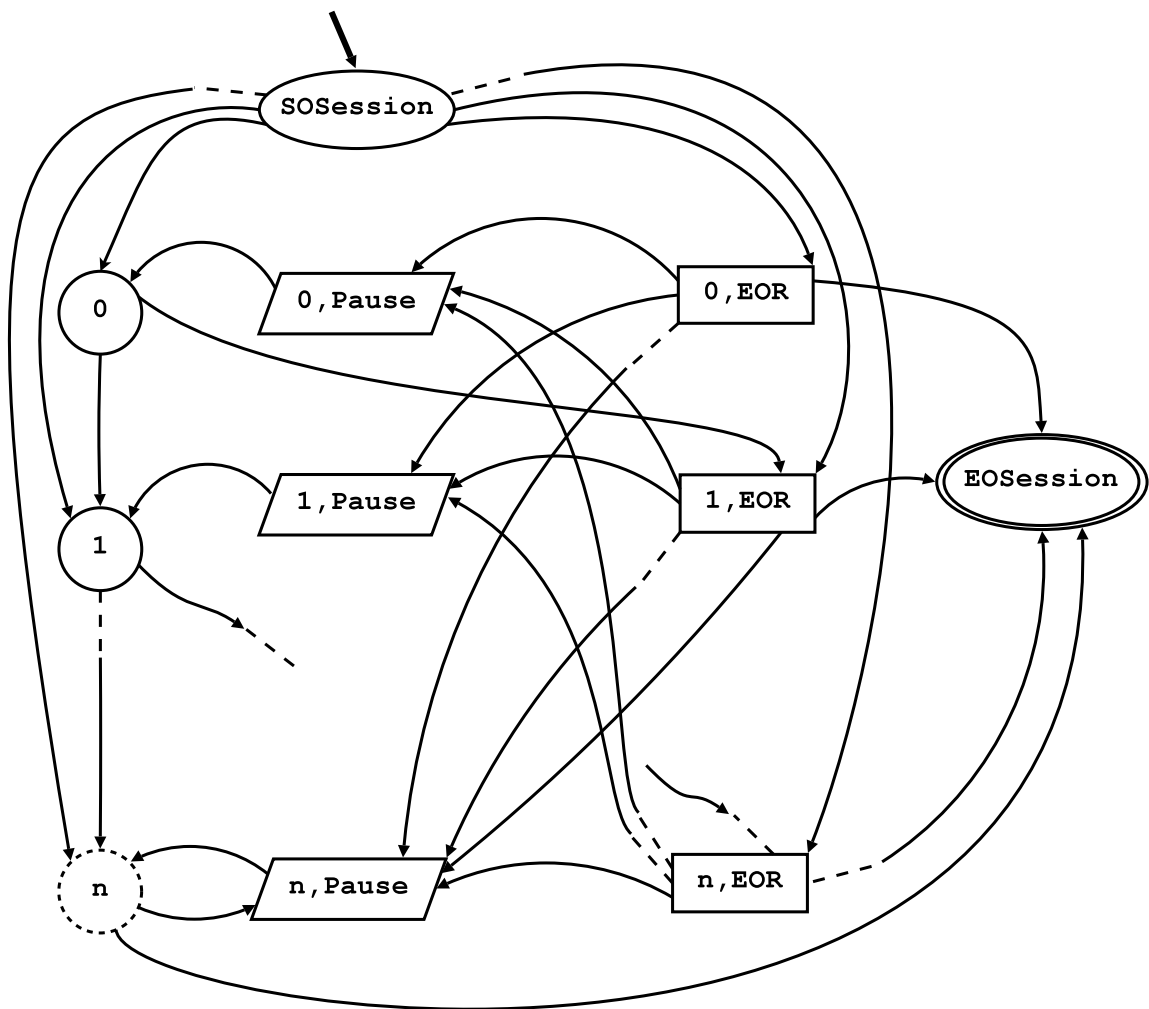


Figura 3.11: Modelo de histograma de uma carga simples

longa cadeia de estados desse tipo;

3. *Segmento Final – Círculo Tracejado*: representa um segmento que é exibido completamente e é o último segmento da mídia. A duração desse estado é fixa e depende unicamente da duração de um segmento (usam-se 10 segundos). Esse estado pode ser seguido por um estado *Pausa* (caso em que a requisição continua) ou por um estado *Fim de Sessão* (caso em que a requisição termina);
4. *Fim de Requisição – EOR*: esse estado corresponde a um segmento que termina uma requisição mas não a sessão corrente. Representa um segmento que não foi percorrido completamente. A duração desse estado é exponencial, com média igual à da duração no *trace* real para requisições que terminem nesse segmento. Esse estado pode ser seguido por um estado *Pausa* (caso em que a sessão prossegue) ou por um estado *Fim de Sessão* (caso em que a sessão termina);
5. *Pausa – Pause*: representa o período de pausa após um estado de *Fim de Requisição*

e antes do início da próxima requisição. A duração desse estado é exponencial, com média dada pela média dos tempos de pausa das requisições que percorreram esse estado. Essa é uma representação aproximada da duração de pausas, que, segundo [25], é mais bem aproximada pela distribuição *Weibull*. Essa aproximação também dificulta a geração de pausas com duração 0. No entanto, a validação do modelo (ver adiante) mostra que essa aproximação não impede que o modelo seja eficaz em capturar, pelo menos numa primeira aproximação, as características fundamentais das cargas interativas;

6. *Fim de Sessão – EOSession*: é o estado final do diagrama. Ao atingir esse estado, termina a geração de requisições.

Um mesmo segmento da mídia pode ser percorrido completamente em uma requisição (será gerado um estado *Segmento Comum* para esse caso) e parcialmente em outra (será gerado um estado *EOR* para esse caso). É o que ocorre com o segmento 1 na Figura 3.11. Portanto, para cada segmento de mídia podem ser criados até dois estados.

Foram gerados perfis de carga para 36 *traces* reais de sessões, cada um para um objeto popular distinto, os quais foram selecionados dentre as cinco cargas reais das quais esta tese faz uso. Esses *traces* reais foram selecionados porque: (1) contêm um número elevado de requisições (acima de 200) e (2) exibem diferentes perfis de comportamento interativo. Três perfis típicos são mostrados na figura 3.12.

Para avaliar a precisão do gerador de cargas sintéticas, foram usadas três abordagens distintas. A primeira é meramente visual e pode ser observada na figura 3.12. Nessa figura são mostrados perfis de cargas sintéticas de nível alto (figura 3.12(a)), médio (figura 3.12(b)) e baixo (figura 3.12(c)), similares às cargas reais apresentadas na figura 3.10. A comparação visual entre as duas figuras mostra grande similaridade na posição inicial e na duração das requisições.

No entanto, a mera inspeção visual é insuficiente para indicar que a carga sintética reproduz, de forma adequada, os parâmetros da carga real que a originou. Então, as abordagens seguintes buscam medir características de ambas as cargas (real e sintética), de forma a permitir uma comparação quantitativa. Para tanto, para cada um dos 36 *traces* reais é gerado um *trace* sintético correspondente. O *trace* sintético gerado tem taxa de chegadas  $N$  e número de requisições próximos aos do *trace* real correspondente.

A segunda forma de avaliar o gerador é, então, comparar a distribuição acumulada da posição inicial das requisições e a da duração das requisições para cada par de *trace* real e *trace* sintético<sup>2</sup>. Para ambos os parâmetros e para todos os pares de *traces* (o real e o sintético cor-

---

<sup>2</sup>Esses dois parâmetros são escolhidos por serem os capturados pela dispersão espacial (ver seção 3.1).



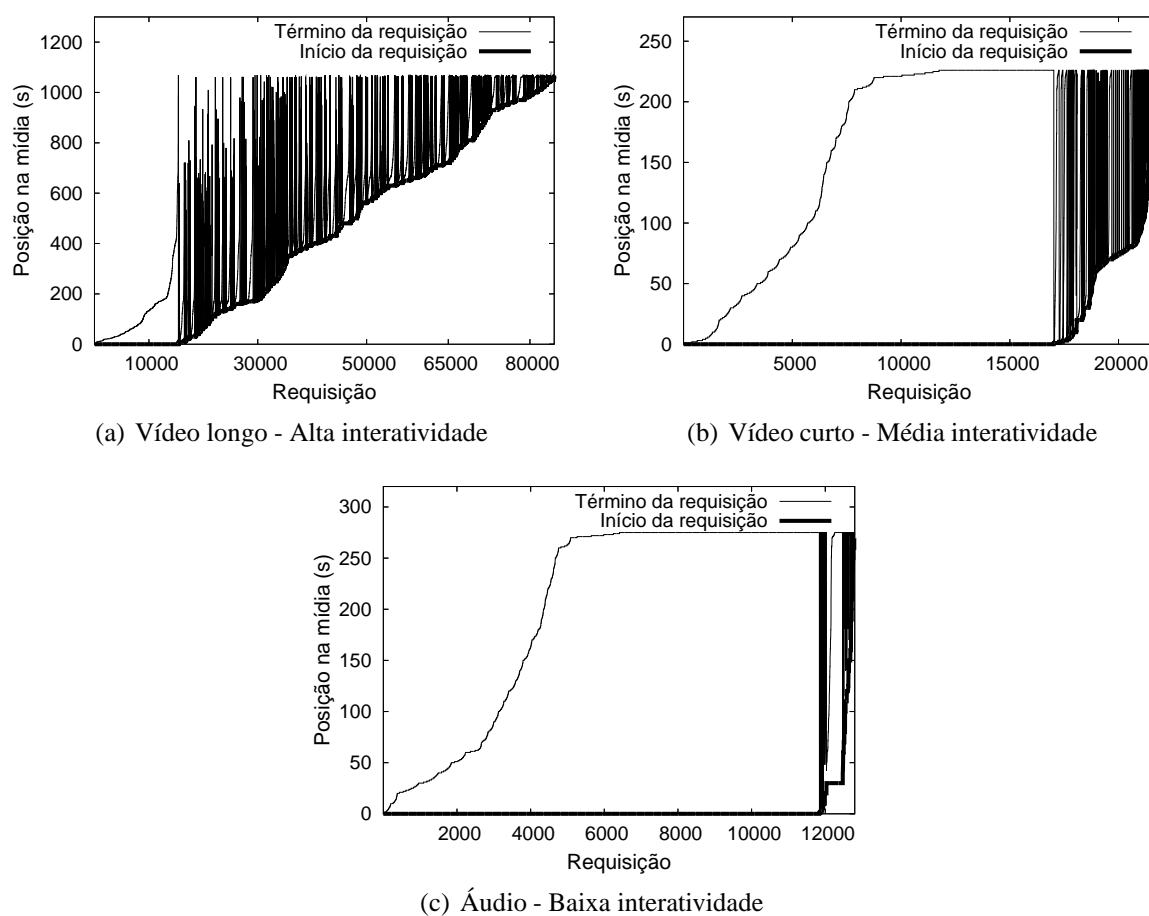


Figura 3.12: Perfis de comportamento interativo para cargas sintéticas

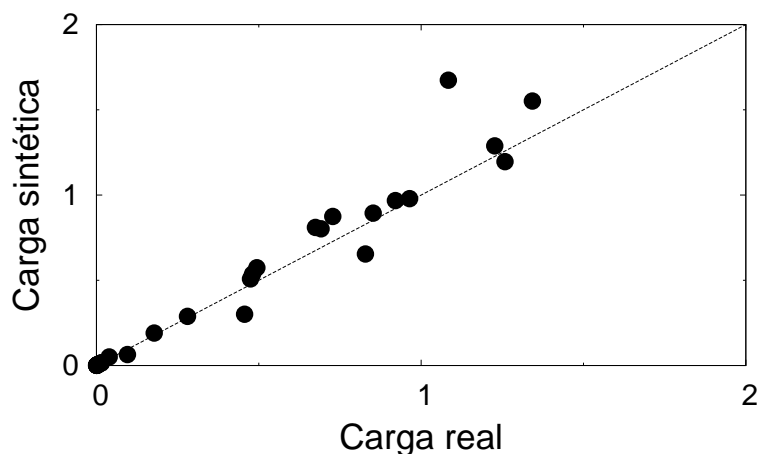


Figura 3.13: Número médio de fluxos simultâneos

respondente), a soma das diferenças quadráticas entre as distribuições para a carga real e a sintética, normalizadas pela escala do eixo x, está abaixo de 0.03.

A terceira comparação é feita, medindo o número médio de fluxos simultâneos (isto é, a banda requerida para a transmissão *Unicast*) requerido para cada par de *traces*<sup>3</sup>. A figura 3.13 mostra os resultados, com um ponto para cada par de *traces*<sup>3</sup>. As medidas para cargas reais e sintéticas são bastante próximas, com erros abaixo de 27% em média. Assim, dada a complexidade dos padrões reais de acesso interativo, considera-se que o modelo seja eficaz em capturar, pelo menos numa primeira aproximação, as características chaves das cargas reais empregadas nesta tese.

**Conclusão:** Este capítulo discute diversos aspectos relativos às cargas de média contínua interativa. Define o conceito de dispersão nas dimensões temporal e espacial, usando-o para classificar cargas de média contínua, a fim de simplificar a análise de resultados de simulações efetuadas em capítulos seguintes desta tese. Define também uma métrica para a dispersão e mostra como calcular o coeficiente de dispersão para o protocolo *Bandwith Skimming*. Esse coeficiente é usado em capítulos seguintes para a avaliação de escalabilidade do *Bandwith Skimming* e como base para o projeto de uma estratégia adaptativa. As cargas reais apresentadas neste trabalho são usadas como base para a síntese de um conjunto rico de cargas sintéticas realistas foram apresentadas.

---

<sup>3</sup>A carga baixa por objeto nos servidores analisados (abaixo de dois fluxos simultâneos) e, portanto, com poucas oportunidades para compartilhamento e de economia de banda, é, justamente, o que motiva a geração de cargas sintéticas realistas. São as cargas sintéticas que permitem investigar o comportamento dos protocolos para cargas mais intensas, nas quais há mais oportunidades para compartilhar banda e para as quais se faz necessário economizar banda de servidor.

Os capítulos imediatamente posteriores avaliam protocolos de mídia contínua correntes e novas otimizações para todos os perfis de cargas sintéticas identificados nas cargas empregadas nesta tese. Já que cargas de uma mesma categoria exibem padrões de interatividade similares e, assim, impacto qualitativamente similar sobre a escalabilidade de protocolos, são mostrados resultados representativos de uma instância de cada perfil. Para cada perfil, varia-se a taxa de requisições  $N$  para avaliar o impacto da intensidade de carga. Os resultados são a média de cinco cargas, cada uma com 100.000 requisições, criadas com o mesmo perfil e sementes diferentes do gerador de números aleatórios, e têm um desvio padrão abaixo de 2% da média para todos os casos.

## 4 *O impacto da interatividade nos protocolos correntes de mídia contínua*

Este capítulo avalia a escalabilidade, para cargas interativas realistas, de dois protocolos que representam o estado-da-arte em termos de protocolos de mídia contínua, *Patching* e *Bandwidth Skimming*. A avaliação usa o conjunto de cargas sintéticas interativas descrito no capítulo 3, que inclui uma ampla faixa de taxas de chegadas, e assim cobre uma região do espaço de projeto de protocolos de mídia contínua bastante mais ampla que a coberta por trabalhos anteriores [11, 12, 14, 23, 32, 33, 38, 43, 55]. Também é feita uma análise detalhada da escalabilidade do *Bandwidth Skimming*, e é mostrada expressão para sua escalabilidade, obtida experimentalmente, mais precisa para cargas IA e IM que o resultado prévio [74]. Por fim, a seção 4.3 avalia os requisitos de banda de servidor do *Bandwidth Skimming* em função do coeficiente de dispersão.

### 4.1 *Comparando a escalabilidade dos protocolos Patching e Bandwidth Skimming*

Foram construídos simuladores para *Patching* e *Bandwidth Skimming* para medir a banda média de servidor<sup>1</sup> requerida para sustentar uma carga dada. Os simuladores foram validados para cargas seqüenciais, comparando, para uma ampla faixa de taxas de chegadas, a banda de servidor medida com o valor correspondente definido pelo resultado analítico desenvolvido em trabalhos anteriores [33, 38]. Os resultados analíticos e os experimentais diferem entre si por, no máximo, 12%.

Os simuladores foram estendidos, com pequenas modificações, para operar sob cargas interativas. Fluxos agora podem ser iniciados e terminados em posições arbitrárias da mídia. Uma requisição que chega para uma posição que está sendo transmitida para um fluxo já existente une-se automaticamente a esse fluxo. Finalmente, uma decisão teve de ser tomada com

---

<sup>1</sup>Definida como sendo a medida da banda média de servidor usada, considerando-se que o servidor tem banda infinita e que os clientes são atendidos imediatamente [74].

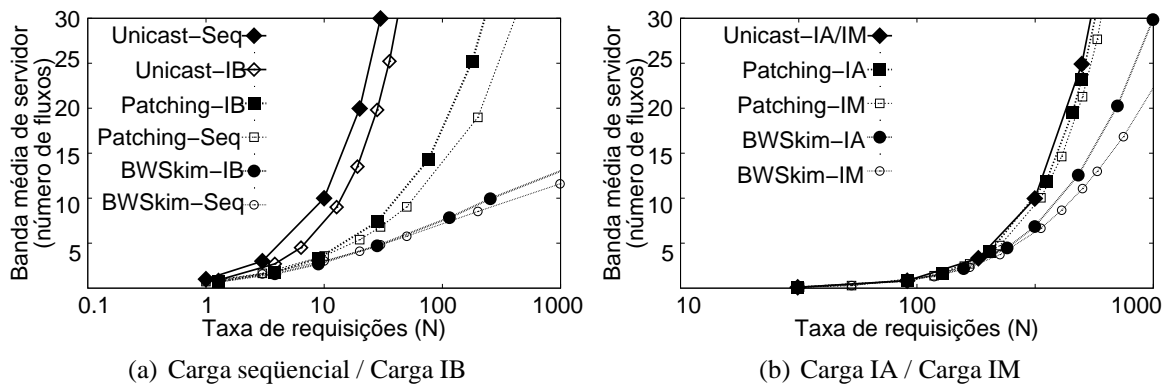


Figura 4.1: Banda média de servidor para *Unicast*, *Patching* e *Bandwidth Skimming*

respeito à aplicação da janela ótima do *Patching* (veja a seção 2.1.1) para as requisições interativas. Assim como em [12], optou-se pela modificação menos intrusiva: aplica-se a janela ótima [38] apenas para requisições que recuperam um prefixo da mídia. Assim, apenas fluxos que se principiem na posição inicial da mídia podem executar uniões, e, portanto, requisições que *não* principiem na posição inicial da mídia serão atendidas por fluxos *Unicast*. Sem tal janela, e também sem nenhum outro parâmetro a ajustar, a adaptação do *Bandwidth Skimming* para cargas interativas é simples e direta. Não se usa *buffer* no cliente. Todos os resultados são a média de cinco experimentos com cargas criadas a partir do mesmo modelo e variando-se as sementes dos geradores de números aleatórios, com 100.000 requisições em cada carga e com desvio padrão abaixo de 2% da média, para todos os casos.

A figura 4.1 mostra a banda média de servidor, em número de fluxos, requerida pelo *Patching* pelo *Bandwidth Skimming*, e pela transmissão *Unicast* (para propósito de comparação) para perfis de cargas típicas, com variados graus de interatividade e de taxas de requisição. A figura 4.1(a) mostra os resultados para cargas seqüencial de perfil IB, enquanto que perfis típicos de cargas IA e IM são mostrados na figura 4.1(b). Já que a banda média de servidor para *unicast* é similar para as cargas IA e IM, a figura 4.1(b) mostra apenas uma das duas curvas. Note-se que uma requisição em uma carga IA transmite menos dados que uma requisição numa carga IB. Assim, as figuras 4.1(a) e 4.1(b) não podem ser comparadas diretamente.

Claramente, *Bandwidth Skimming* tem escalabilidade melhor que *Patching*, para todos os perfis de carga. Comparado com *Patching*, *Bandwidth Skimming* reduz a banda média de servidor em até 88% para cargas IB e em até 74% para cargas IA. Resultados similares são observados para todos os outros perfis de carga analisados.

Apesar da superioridade com relação ao *Patching*, a escalabilidade do *Bandwidth Skimming* também sofre forte impacto das cargas interativas. A elevada dispersão espacial das cargas IA e

IM leva muitas tentativas de união de fluxos a falhar. Dados recebidos por antecipação, durante essas tentativas de fusão, são perdidos e, freqüentemente, requisitados novamente pelo mesmo cliente. A economia de banda é, então, muito modesta, especialmente para taxas de chegadas de requisições muito baixas.

## 4.2 Escalabilidade do *Bandwidth Skimming* com cargas interativas em função de $N$

Nesta seção, é feita uma avaliação, em duas etapas, da escalabilidade do *Bandwidth Skimming* para cargas interativas realistas. Na primeira etapa, avalia-se, para cargas de baixa, média e alta interatividade, quão precisa é a escalabilidade da banda média obtida por [74] e expressa pela equação (2.1). Na segunda etapa, identifica-se uma nova expressão para a escalabilidade do *Bandwidth Skimming* em função da taxa de chegadas  $N$ , para cargas com perfis de interatividade média e alta. Essa expressão é obtida por meio de simulações usando as cargas sintéticas realistas descritas no capítulo 3.

Inicialmente reescreve-se a equação (2.1) [74] para a forma:

$$B_{\text{média}}^{\text{interatividade}}(N) = k \left( \frac{2}{\pi} N \right)^{0.5} \quad (4.1)$$

A seguir, efetuam-se simulações usando o protocolo *Bandwidth Skimming* com médias cobrindo os diversos perfis de interatividade (36 médias ao todo), geradas conforme mostrado na seção 3.5. Para cada média empregam-se valores crescentes de  $N$  (definido na seção 3.1) e calcula-se a banda média usada ao longo de cada simulação. Por fim, compara-se a banda média obtida nas simulações com a banda obtida pela aplicação de valores variados de  $k$  à relação (4.1).

A figura 4.2 mostra resultados de simulações para três cargas de perfis, IA, IM e IB. Os resultados para outras cargas de mesmo perfil são qualitativamente similares aos apresentados na figura. A figura também mostra curvas calculadas pela relação (4.1) para três valores de  $k$ . No eixo  $x$  têm-se valores crescentes de  $N$ , e, no eixo  $y$ , a banda média correspondente. Na figura observa-se que, para cargas com nível médio a alto de interatividade (figuras 4.2(a) e 4.2(b)), a banda média cresce mais rapidamente que o previsto pela relação (4.1). Já para cargas com nível de interatividade baixo (figura 4.2(c)), o crescimento da banda média é similar ou inferior ao previsto por essa relação. Esse é o resultado esperado, uma vez que em perfis de baixa interatividade as requisições são, em média, mais longas e o desempenho do protocolo *Bandwidth Skimming* sofre menor degradação.

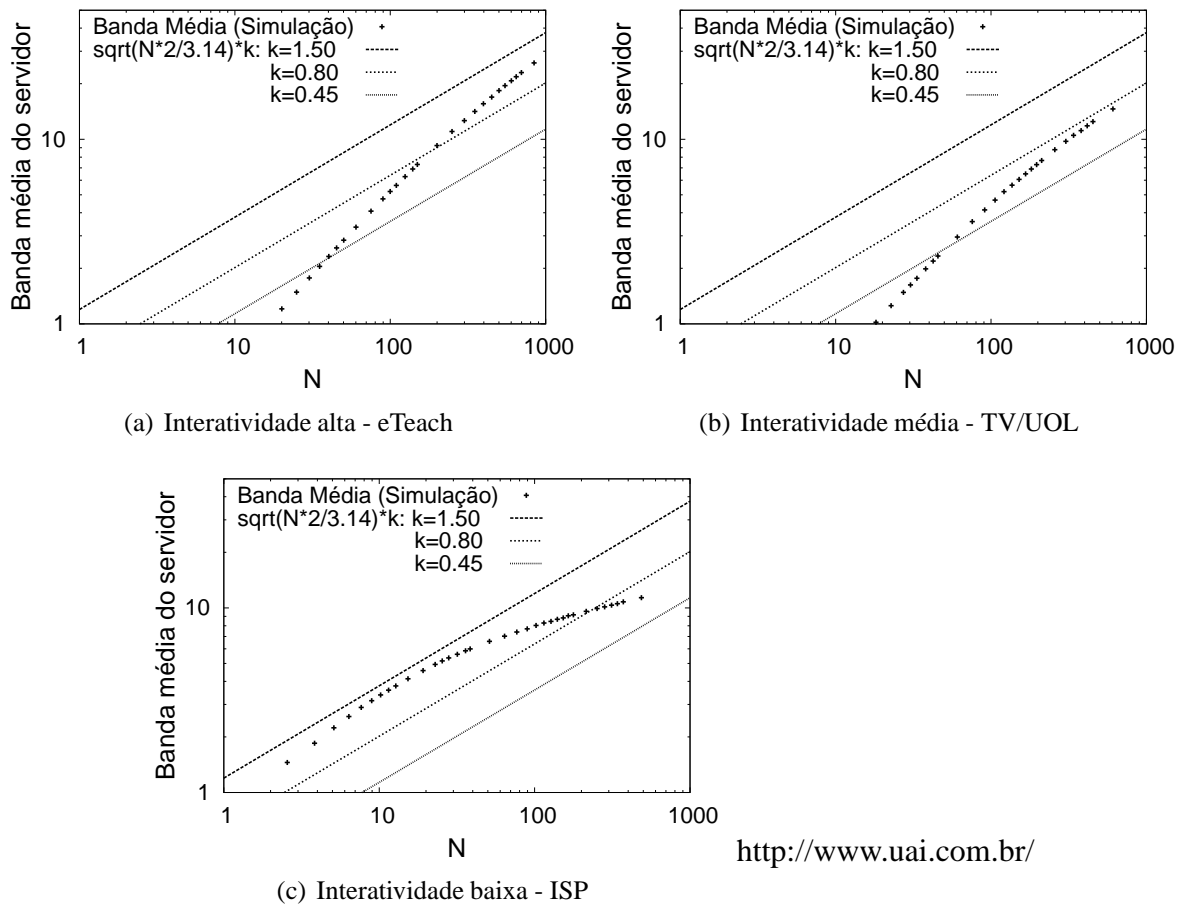


Figura 4.2: Banda média do *Bandwidth Skimming* e banda calculada pela equação (4.1) para três valores de  $k$

A seguir, é feito o ajuste de uma curva polinomial para representar o crescimento da banda média no *Bandwidth Skimming* com cargas de perfil de alta e de média interatividade (para as cargas avaliadas neste trabalho). Sendo o crescimento da banda média medida para perfis de alta e média interatividade nas figuras 4.2(a) e 4.2(b) polinomial, segue-se a orientação encontrada em [4] e efetua-se a regressão da banda média medida à relação:

$$B_{media}^{interatividadeBS}(N) = aN^b \quad (4.2)$$

Essa relação foi transformada para um domínio linear para o cálculo da regressão [4]. Não se faz essa determinação para os perfis de baixa interatividade porque o crescimento de banda média para esses casos é logarítmico [74]. Em lugar disso, nota-se que o resultado prévio dado pela equação (4.1) pode ser aplicado para essas cargas.

Para todas as cargas com perfil de alta ou média interatividade, o coeficiente de correlação  $R^2$  da regressão encontra-se no intervalo  $[0,94,0,99]$ , o que evidencia que o ajuste das curvas à equação (4.2) é satisfatório (acima de 0,9). Ainda para essas cargas, os valores de  $b$  situam-se

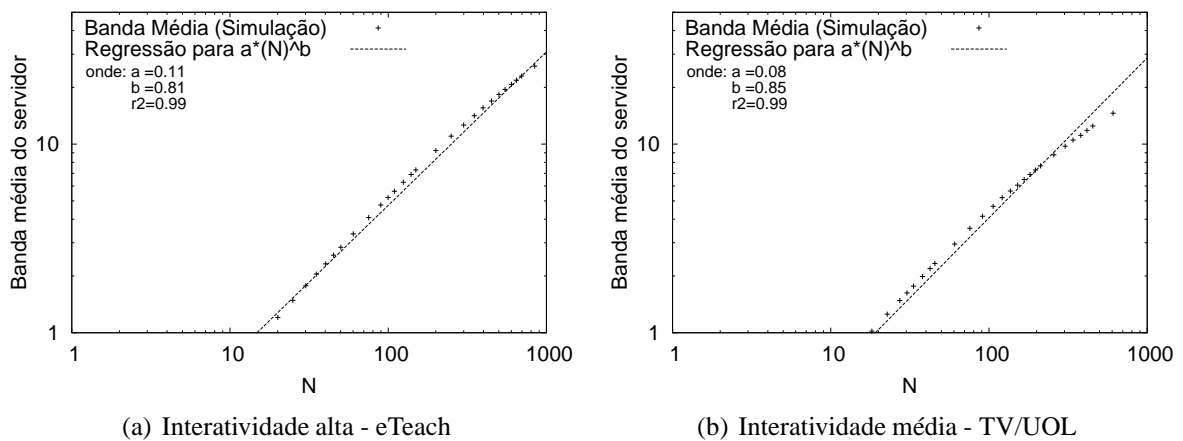


Figura 4.3: Regressão da banda média do *Bandwidth Skimming* à equação (4.2)

na faixa  $[0,68,0,81]$  e, portanto, acima do valor de 0,5 previsto pelo resultado prévio expresso pela equação (4.1). Por fim, os valores de  $a$  situam-se no intervalo  $[0,08,0,46]$ . Exemplos de regressão para as cargas das figuras 4.2(a) e 4.2(b) são mostrados na figura 4.3.

Concluindo, obtém-se, por meio do ajuste de uma curva polinomial, uma expressão para a escalabilidade da banda média para o *Bandwidth Skimming* que pode ser aplicada a perfis de média e de alta interatividade. Deve-se notar que os valores de  $a$  e  $b$  são específicos para as cargas empregadas neste trabalho e podem se alterar caso se usem outras cargas.

A expressão obtida permite uma avaliação mais precisa da escalabilidade das cargas aqui empregadas que o resultado analítico anterior, cujo uso levaria a subestimar ou a superestimar a carga média a ser suportada por um servidor. Além disso, verificou-se que o resultado prévio para escalabilidade da banda média obtido por [74] e expresso pela equação (4.1) pode ser usado para representar a escalabilidade da banda média do *Bandwidth Skimming* para cargas com perfil de baixa interatividade. Em nenhuma das simulações aqui efetuadas, a banda média excedeu a banda média calculada com o valor de  $k = 1.5$ . Portanto, esse valor pode ser usado para aplicar o resultado prévio a estudos de planejamento de capacidade, ainda que, conforme observado na figura 4.2(c), os valores obtidos superestimem a necessidade de banda para taxas de chegada mais altas. Essa contribuição é significativa, uma vez que o resultado prévio obtido por [74] não fornece indicação para cálculo do valor de  $k$ .



### 4.3 Escalabilidade do *Bandwidth Skimming* com cargas interativas em função da dispersão

Nesta seção, a dispersão é usada para estimar a banda média de servidor requerida pelo protocolo *Bandwidth Skimming* com cargas de mídia interativas. Esse resultado é usado no capítulo 7 como uma referência para estimar se um dado protocolo ou estratégia provê economia de banda com relação ao *Bandwidth Skimming*.

Com cargas sequenciais, a taxa de chegadas  $N$  indica o número médio de clientes que estão ativos e requisitando a exibição de mídia. Assim,  $N$  também representa o número médio de fluxos *Unicast* simultâneos necessários para entregar a carga. Já com cargas interativas, nas quais cada requisição recebe um segmento da mídia, o número médio de clientes ativos e, portanto, o número médio de fluxos *Unicast* que devem estar ativos simultaneamente para entregar a carga são dados por:

$$C = \lambda L \quad (4.3)$$

onde  $L$  é a duração média das requisições.

As figuras 4.4 e 4.5 são usadas para ilustrar por que, nas cargas interativas, a escalabilidade do *Bandwidth Skimming* será avaliada em termos da economia de banda obtida em relação à transmissão *Unicast*. Na figura 4.4 são mostradas duas seqüências de requisições para a mesma mídia de 20 minutos de duração e com o mesmo coeficiente de dispersão ( $\Delta_{BS} = 0,4$ ). Na figura 4.4(a) são requisitados (e entregues) 119 minutos ( $N = 6, C = 3$ ), enquanto que na figura 4.4(b) são requisitados (e entregues) 76 minutos ( $N = 4, C = 1,9$ ). Na figura 4.5 são mostrados os dados efetivamente transmitidos para atender essas requisições com o protocolo *Bandwidth Skimming*. Na figura 4.5(a), são transmitidos 72 minutos ( $N = 6, C = 1,9$ ), com uma

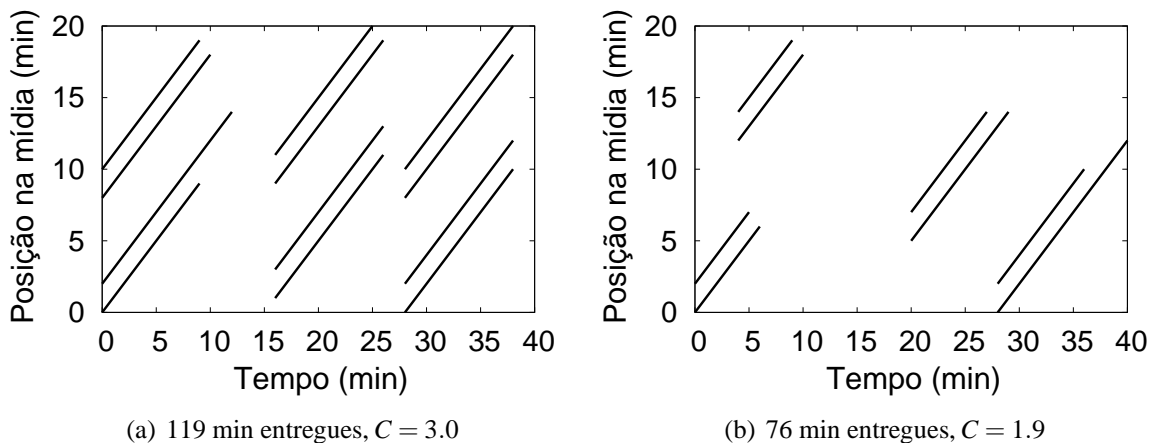


Figura 4.4: Transmissão com *Unicast* ( $\Delta_{BS} = 0,4$ )

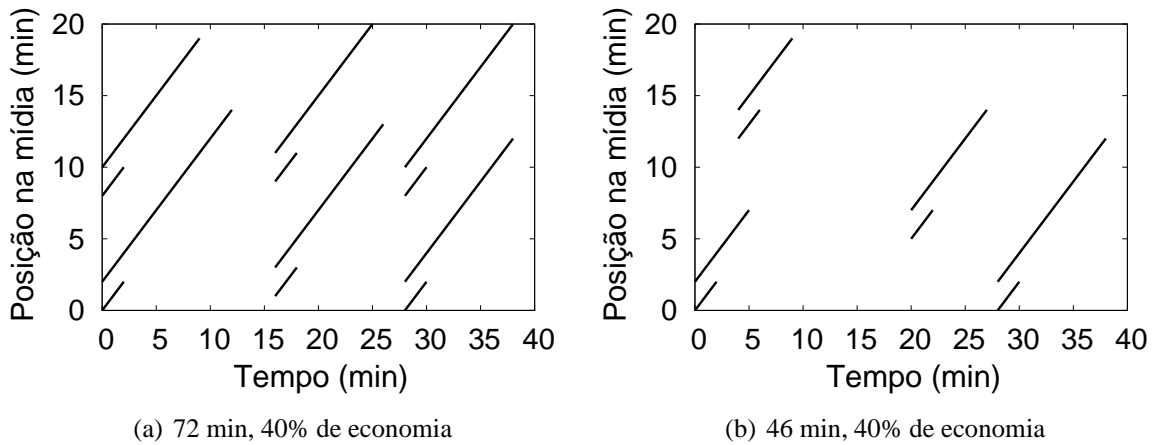


Figura 4.5: Transmissão com *Bandwidth Skimming*

economia de 40% sobre *Unicast*, enquanto que na figura 4.5(b), são transmitidos 46 minutos ( $N = 4$ ,  $C = 1,2$ ), com uma economia também de 40% sobre *Unicast*. Vê-se, então, que a economia de banda com relação ao *Unicast* é similar nas duas cargas em que  $\Delta_{BS}$  é similar. Essa relação entre o coeficiente de dispersão do *Bandwidth Skimming* e a economia de banda obtida em relação à transmissão *Unicast* é avaliada mais detalhadamente a seguir.

A relação entre o coeficiente de dispersão do *Bandwidth Skimming* e a economia de banda obtida em relação à transmissão *Unicast* é mostrada na figura 4.6, que apresenta os resultados de 959 experimentos com 36 perfis de interatividade diferentes. Cada experimento emprega, como nas seções 4.1 e 4.2, uma carga sintética com 100.000 requisições e o mesmo simulador para o *Bandwidth Skimming* e avalia a banda média de servidor requerida para a carga simulada. No eixo  $x$ , mostra-se o coeficiente de dispersão, e, no eixo  $y$ , a economia de banda média de servidor sobre a transmissão com *Unicast*. Pode-se notar que a escalabilidade se reduz à medida que a dispersão (e seu impacto sobre a escalabilidade) aumenta (o aumento dá-se seja pela interatividade mais alta, seja pela taxa de chegadas  $N$  mais baixa). De fato, foi justamente a esse impacto medido neste capítulo com o coeficiente de dispersão  $\Delta_{BS}$ , que motivou o projeto, no capítulo 5, de estratégias capazes de melhorar a escalabilidade do *Bandwidth Skimming* para cargas com níveis médios a altos de interatividade (cargas com maior dispersão). Por fim, essa figura evidencia uma correlação entre o coeficiente de dispersão e a banda média de servidor requerida pelo *Bandwidth Skimming* com cargas interativas.

Os resultados apresentados na figura 4.6 são usados para estimar a banda média de servidor requerida pelo protocolo *Bandwidth Skimming*. Deve-se notar que a figura 4.6 permite estimar diretamente a *economia de banda* que o *Bandwidth Skimming* obtém em relação ao *Unicast*. Para estimar a banda, é necessário considerar a banda requerida pelo *Unicast*, e que corresponde ao número médio de clientes ativos  $C$  (essa estimativa é usada, pelo protocolo adaptativo

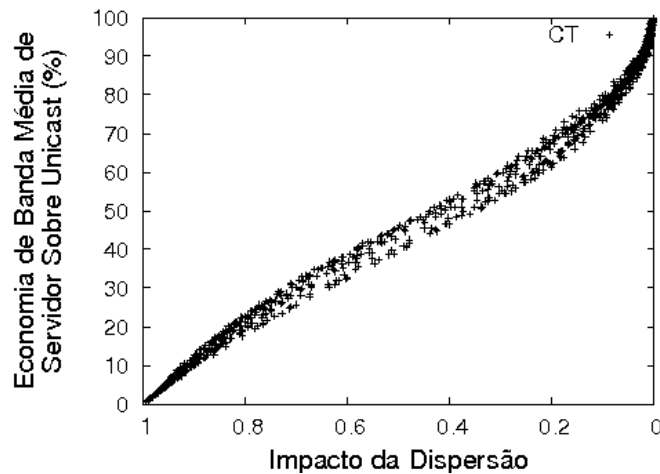


Figura 4.6: Economia do *Bandwidth Skimming* sobre *Unicast*

apresentado no capítulo 7, como uma salvaguarda para desconectar todas as estratégias dedicadas a cargas interativas caso o protocolo adaptativo comece a requerer mais banda que o *Bandwidth Skimming* original). A estimativa para a banda requerida é calculada conforme se segue:

$$P_{BS}(\Delta_{BS}, C) = S_u(\Delta_{BS}) * C \quad (4.4)$$

onde  $S_u(\Delta_{BS})$  é a economia de banda sobre *Unicast*, que corresponde, na figura 4.6, ao coeficiente de dispersão  $\Delta_{BS}$ . Para encontrar esse valor, constrói-se uma tabela na qual cada entrada contém a economia média e o coeficiente de dispersão médio para um conjunto de cargas com dispersão próxima, recuperando-se então, a entrada que melhor corresponde a  $\Delta_{BS}$ .

Esse método foi aplicado às 959 cargas descritas no início desta seção, e o erro, que é calculado como a diferença percentual entre a banda obtida com a simulação e a banda estimada com a equação 4.4, é, em média, de 5%. O erro situa-se abaixo de 10% para 86% das cargas, e não excede a 25% para nenhuma das cargas avaliadas

**Conclusão:** Este capítulo avalia o impacto da interatividade sobre a escalabilidade de dois protocolos; *Patching* e *Bandwidth Skimming*. Analisa-se mais detalhadamente o impacto registrado para o *Bandwidth Skimming*, selecionado como objeto principal do trabalho por apresentar melhor escalabilidade. Essa análise traduz-se uma forma de escolher alvos para as estratégias de minimização desse impacto e de mostrar que o coeficiente de dispersão pode ser usado para estimar a escalabilidade do mesmo protocolo. O efeito degenerativo da fragmentação também é aqui identificado.

O próximo capítulo discute diversas estratégias que, dedicadas a cargas interativas e aplicadas ao *Bandwidth Skimming*, lidam com os aspectos negativos da dispersão temporal e da

dispersão espacial, evitando perdas de dados inerentes ao protocolo original e reduzindo o número de vezes que um mesmo dado deve ser transmitido. Contudo, deve-se observar que, embora sejam apresentadas como extensões ao *Bandwidth Skimming*, essas estratégias podem ser aplicadas, com pequenas modificações, a outros protocolos, como o *Patching*.

## ***5 Estratégias de transmissão de mídia contínua para cargas interativas***

Este capítulo examina estratégias para transmissão de mídia contínua, as quais têm por objetivo reduzir o impacto da interatividade na escalabilidade de protocolos de compartilhamento de banda. A fim de varrer o espaço de projeto de protocolos de compartilhamento de banda, esse exame é dividido em etapas. Em primeiro lugar, as restrições consideradas para rede e para servidor são apresentadas na seção 5.1. A seção 5.2 descrever uma série de estratégias que exploram aspectos das cargas de mídia interativas (ou seja, que levam em consideração o comportamento dos usuários de serviços de mídia contínua). Essas estratégias, que também se servem de características dos protocolos de compartilhamento de banda, são avaliadas em detalhe na seção 5.3. Na seqüência (seção 5.4), estratégias que exploram o período inativo são propostas e avaliadas. A seção 5.5 apresenta estratégias híbridas que combinam estratégias individuais, seja para juntar os esforços destas e aproveitar ao máximo as possibilidades de economia de banda, seja para compensar eventuais deficiências delas e, assim, obter economia para faixas mais amplas de dispersão.

### **5.1 Modelagem da rede e arquitetura de comunicação**

Todos os algoritmos e as estratégias propostas neste capítulo e em capítulos subseqüentes fazem uma série de suposições quanto à modelagem de rede e à arquitetura de comunicação subjacentes. Quanto ao cliente, não são considerados atrasos de processamento nem de armazenamento.

Com relação ao servidor, as restrições consideradas são as seguintes:

1. o servidor conta com banda ilimitada;
2. o tempo de resposta é imediato, e não há atrasos nem custos de processamento;

3. as estratégias tratam da transmissão de uma única mídia para múltiplos usuários, e o impacto do atendimento a múltiplas mídias não é considerado;
4. toda as mídias têm taxa de codificação constante (CBR – constant bit rate), e, por conseguinte, todos os fluxos requerem a mesma banda;
5. as avaliações de estratégias baseiam-se em banda média de servidor, estando portanto focadas nos cenários em que o sistema simulado se encontra numa situação estável. A estratégia adaptativa (capítulo 7) mostra uma abordagem para as situações em que a carga não é estável.

Com relação à arquitetura de comunicação, são consideradas as seguintes restrições:

1. a troca de mensagens é instantânea, e nenhum atraso ou sobrecarga é considerado;
2. toda a mídia transmitida é recebida integralmente e no tempo adequado, e atrasos, perdas e retransmissões não são considerados.

A topologia de rede não é levada em conta no modelo. O impacto das otimizações sobre questões relacionadas à topologia é discutido no capítulo 6.

Mesmo com as simplificações indicadas o projeto de técnicas para melhorar a escalabilidade de protocolos de compartilhamento de banda é uma atividade que envolve um conjunto complexo de aspectos e compromissos. Portanto, o tratamento das restrições enumeradas aqui é deixado como trabalho futuro.

## 5.2 Descrição das estratégias

Esta seção apresenta quatro estratégias de transmissão de mídia contínua que têm como alvo reduzir ainda mais os requisitos de banda média de servidor para cargas altamente interativas, para as quais os protocolos *Bandwidth Skimming* e *Patching* originais não conseguem prover boa escalabilidade. Como base para o projeto das estratégias aqui propostas, centrou-se foco no protocolo *Bandwidth Skimming*, já que ele apresenta escalabilidade superior à do *Patching*. Contudo, várias das estratégias propostas são gerais, podendo também ser aplicadas a outros protocolos de compartilhamento de fluxos (ver discussão a seguir).

O principal ponto comum a todas as estratégias é economizar banda de servidor pela redução do número de vezes que um servidor tem de transmitir o mesmo segmento para um ou mais clientes. Assim, essas estratégias usam um *buffer*, local ao computador do usuário, para

---

**Algoritmo 1** Exibição de mídia pelo cliente (com *buffer* local)

---

```
1: loop
2:   Aguardar pedido de exibição
3:   while “usuário não interagir” e “exibição não atingir o fim da mídia” do
4:     if “bloco solicitado está no buffer” then
5:       if “protocolo está ativo” then
6:         “desativar protocolo”
7:       end if
8:       “exibir a partir do buffer”
9:     else
10:      if “protocolo está inativo” then
11:        “ativar protocolo para receber a mídia solicitada”
12:      end if
13:      “exibir o bloco recebido”
14:    end if
15:  end while
16:  “desativar protocolo”
17: end loop
```

---

armazenar segmentos de mídia que tenham sido previamente requisitados ou recebidos, ou que estejam sendo transmitidos para outros clientes. Requisições futuras de um mesmo usuário para segmentos que estejam armazenados em seu *buffer* podem ser servidas localmente, sem que tenham de ser despachadas para o servidor. Note-se que um *buffer* existe apenas durante uma sessão de um usuário. Em outras palavras, o conteúdo do *buffer* é descartado ao final de cada sessão (esse comportamento é consistente com a idéia de que uma nova sessão fará acesso a outra mídia, ou será efetivada por uma nova ativação do cliente após algum período de inatividade). Deve-se notar também que, embora seja apresentado como sendo local ao usuário (sendo gerenciado pelo programa cliente de exibição de mídia), o *buffer* pode estar situado num agente (ou seja, num servidor *proxy*), trabalhando em benefício do cliente.

O projeto das estratégias aqui propostas é dirigido pelas características das cargas reais de mídia contínua. Em particular, a alta localidade de referência encontrada nas cinco cargas reais empregadas (ver seção 3.1) motivou o armazenamento em *buffer* dos segmentos de mídia que estão próximos do segmento correntemente em exibição. Além disso, algumas das estratégias exploram características inerentes ao protocolo *Bandwidth Skimming* original (ver seção 2.1.1), sendo elas a árvore hierárquica de uniões de fluxos e as perdas de dados ocorridas no caso de tentativas fracassadas de união. As novas estratégias, denominadas LOCALITY, SILENT PREFETCH, KEEP MERGE BUFFER e PRESERVE MERGING TREE, são descritas a seguir.

Um esquema geral do funcionamento de um programa cliente com *buffer* local é apresen-

---

**Algoritmo 2** Cliente com estratégia LOC

---

```
1: loop
2:   Aguardar pedido de exibição
3:   while “usuário não interagir” e “exibição não atingir o fim da mídia” do
4:     if “bloco solicitado está no buffer” then
5:       if “protocolo está ativo” then
6:         “desativar protocolo”
7:       end if
8:       “exibir a partir do buffer”
9:     else
10:      if “protocolo está inativo” then
11:        “ativar protocolo para receber a mídia solicitada”
12:      end if
13:      “exibir o bloco recebido”
14:      “armazenar o bloco exibido no buffer local”
15:    end if
16:  end while
17:  “desativar protocolo”
18: end loop
```

---

tado no algoritmo<sup>1</sup> 1 (esse algoritmo mostra apenas como o cliente faz a exibição de conteúdo a partir do *buffer*; a obtenção de conteúdo para preenchimento do *buffer* é objeto das estratégias apresentadas a seguir). O algoritmo considera que o tempo é discretizado e dividido em unidades de igual duração (ver seção 3.2). A mídia é dividida em blocos de um segundo de duração, mas qualquer outra unidade conveniente de tempo poderia ser usada. Nesse algoritmo, o cliente aguarda um pedido de exibição de mídia emitido pelo usuário (linha 2). Uma vez recebido o pedido, o algoritmo do cliente inicia um *loop* de exibição de mídia, do qual só sairá quando toda a exibição atingir o fim da mídia, ou quando o usuário efetuar uma interação, solicitando seja a interrupção da exibição, seja um salto na exibição (linha 3). Nesse *loop*, o cliente verifica se o bloco solicitado já está disponível em *buffer* (o armazenamento do conteúdo no *buffer* é visto adiante, na descrição das estratégias). Se estiver, a exibição é feita a partir do conteúdo já armazenado. Se não estiver, um protocolo é ativado para requisitar ao servidor a transferência da mídia a ser exibida. Note-se que, embora o protocolo aqui considerado seja o *Bandwidth Skimming*, qualquer outro protocolo de serviço imediato poderia ser empregado.

LOCALITY (LOC) armazena em *buffer*, no cliente, todos os segmentos de mídia exibidos pelo usuário, conforme mostrado no algoritmo 2, que, na linha 14, armazena em *buffer* o bloco de mídia recebido e exibido. Essa estratégia explora a alta localidade de referência e a fração significativa de saltos para trás encontradas nas cargas reais aqui empregadas [5, 25]. Note-se

---

<sup>1</sup>Esta tese usa a mesma definição informal para o termo *algoritmo* apresentada em [82] segundo a qual *algoritmo é uma seqüência finita de instruções, cada uma executável por uma máquina em tempo finito, e cada uma produzindo o mesmo resultado para os mesmos dados.*



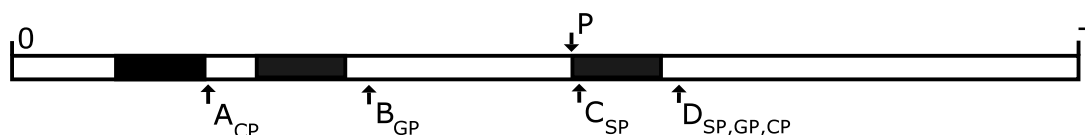


Figura 5.1: Escolha de fluxos de período inativo

que o conteúdo armazenado somente se torna útil para futuras requisições dentro da sessão que o armazenou. Assim, essa estratégia tem seu melhor desempenho para cargas altamente interativas, que tipicamente contêm um grande número de requisições por sessão e de saltos para trás (ver seção 3.4). Note ainda que, sob esse protocolo, o conteúdo armazenado pelo cliente depende unicamente dos segmentos de mídia requisitados pelo usuário numa sessão e não, da taxa de requisições  $N$ .

KEEP MERGE BUFFER (KMB) armazena no *buffer* os segmentos de mídia que seriam, de outra forma, descartados no caso de uma tentativa fracassada de união (ver seção 2.1.1). Note-se que o cliente tem pouco controle sobre qual conteúdo é armazenado em seu *buffer*, já que esse conteúdo depende do número e da duração de cada tentativa fracassada de união, bem como do fluxo alvo de cada tentativa.

PRESERVE MERGING TREE (PMT) tenta evitar que tentativas de união em andamento sejam interrompidas por um usuário que execute uma pausa. Considere o fluxo  $X$ , que está sendo correntemente escutado por apenas um cliente e que está envolvido em duas tentativas de união (ou seja, ele é alvo do fluxo  $Y$  e tenta, ele próprio, unir-se ao fluxo  $Z$ ). Se o usuário que recebe o fluxo  $X$  emite uma pausa, o servidor estende o fluxo  $X$  para permitir que as uniões em andamento sejam concluídas. Caso se tenha sucesso, economiza-se banda. O cliente que executou a pausa continua a escutar e a armazenar em *buffer* a extensão do fluxo  $X$ . Se o usuário retomar a exibição num ponto próximo ao ponto de parada (alta localidade de referência), os próximos segmentos serão encontrados no *buffer*.

SILENT PREFETCH (SP) explora períodos em que o cliente está *silencioso*, ou seja, períodos em que o cliente não está recebendo dados do servidor, para receber dados que estejam correntemente sendo entregues para outros usuários. Essa recepção antecipada chama-se *prefetch*. O cliente está *silencioso* quando o usuário está inativo ou recebendo conteúdo a partir do *buffer* local. O funcionamento da estratégia é o seguinte. Quando o cliente interrompe o fluxo em atenção a uma requisição do usuário, ele passa a receber o fluxo ativo que seja o mais próximo (ou seja, o fluxo que transmite os segmentos mais próximos) do segmento de mídia exibido mais recentemente pelo usuário. O cliente também escuta o alvo desse fluxo (conforme o *Bandwidth Skimming* original), se ele existir. Se os fluxos terminam durante o período silen-

---

**Algoritmo 3** Cliente com estratégia SP

---

```
1: loop
2:   Aguardar pedido de exibição
3:   while “usuário não interagir” e “exibição não atingir o fim da mídia” do
4:     if “bloco solicitado está no buffer” then
5:       if “protocolo está ativo” then
6:         “desativar protocolo”
7:         “ativar protocolo para receber fluxo (e alvo do fluxo) que transmita dado mais
           próximo e adiante do último bloco exibido”
8:       end if
9:       “exibir a partir do buffer”
10:    else
11:      if “protocolo está inativo” then
12:        “ativar protocolo para receber a mídia solicitada”
13:      end if
14:      “exibir o bloco recebido”
15:      “armazenar o bloco exibido no buffer local”
16:    end if
17:    end while
18:    “desativar protocolo”
19:    if “usuário emitiu pausa” then
20:      “ativar protocolo para receber fluxo (e alvo do fluxo) que transmita dado mais próximo
           e adiante do último bloco exibido”
21:    end if
22:  end loop
```

---

cioso, o cliente contata o servidor para obter o(s) próximo(s) fluxos(s) a escutar. Uma visão geral do algoritmo do cliente que emprega a estratégia SP é mostrada no algoritmo 3. Nesse algoritmo, as linhas 7 e 20 ativam a escuta de fluxos nos períodos inativos. A figura 5.1 ilustra a escolha de fluxos<sup>2</sup>, apresentando o conteúdo do *buffer* local do cliente para uma mídia de duração  $T$  num dado instante. Os fluxos disponíveis para escuta estão transmitindo os pontos  $A$ ,  $B$ ,  $C$  e  $D$  (os índices nos fluxos indicam qual estratégia seleciona o fluxo em questão). Os trechos marcados em preto indicam área já preenchida, e  $P$  indica o último ponto da mídia exibido. O fluxo  $C$  será escutado porque é o mais próximo do último ponto exibido  $P$ . Seu alvo  $D$  também será escutado. Note-se que o conteúdo armazenado sob a política SP no *buffer* de um cliente depende das requisições feitas por outros clientes e, assim, também da taxa de chegadas. Contudo, o cliente tem algum controle sobre o conteúdo do *buffer*, porque ele sempre tenta armazenar segmentos em falta e que sejam próximos da posição corrente de exibição, portanto mais prováveis de se requisitar em acessos subseqüentes. Isso não impede a escuta de conteúdo já armazenado, mas reduz a fragmentação do *buffer*. Por fim, essa escolha justifica-se porque operações de pausa e saltos para a frente são freqüentes em cargas interativas [25].

---

<sup>2</sup>A figura também ilustra a escolha de fluxos para as estratégias GP e CP, apresentadas na seção 5.4.

Se o espaço em *buffer* for restrito, essas estratégias tiram partido da alta localidade de referência presente nas cargas (ver seção 3.4), como um guia para usar da melhor forma possível o espaço limitado. A idéia fundamental é manter em *buffer* os segmentos mais prováveis de se requisitar num futuro próximo. Assim, sempre se priorizam os segmentos que estão mais próximos (seja à frente, seja atrás) do segmento que está sendo exibido pelo usuário.

Note-se que, embora as estratégias sejam apresentadas como extensões ao *Bandwidth Skimming* original, elas podem ser também aplicadas a clientes com banda de recepção limitada, como no protocolo original [32]. Além disso, diferentemente de KMB e PMT, as estratégias LOC e SP (e também GP e CP vistas adiante, na seção 5.4) são gerais e podem ser aplicadas, com pequenas modificações, a qualquer protocolo de compartilhamento de banda.

### 5.3 Avaliação

Esta seção provê uma avaliação quantitativa de cada uma das novas estratégias, para cargas e taxas de chegada de requisições variadas. O simulador de *Bandwidth Skimming*, já validado para cargas seqüenciais, foi adaptado para cargas interativas e para implementar as novas estratégias, como descrito na seção anterior. A seguir, são mostrados alguns resultados fundamentais que capturam os compromissos mais relevantes para a escalabilidade.

Inicialmente, deve-se notar que o armazenamento de dados em *buffer* no cliente modifica a seqüência de requisições que é enviada para o servidor. Essa modificação na carga, percebida pelo servidor, causa impacto na eficácia de *qualquer protocolo* baseado no armazenamento de dados em cliente, e na eficácia das estratégias em particular, em duas direções opostas. Por um lado, o conteúdo previamente armazenado não precisa ser retransmitido pelo servidor em acessos futuros, o que reduz os requisitos de banda de servidor. Por outro lado, se os dados armazenados em *buffer* são fragmentados, as requisições do usuário são divididas em diversas requisições menores e espacialmente dispersas, antes de seu envio ao servidor. A dispersão espacial aumentada leva a um número menor de uniões bem sucedidas e, como resultado, a um aumento dos requisitos de banda. A fragmentação de requisições é discutida em maior detalhe no fim desta seção.

As figuras 5.2(a), 5.2(b) e 5.2(c) apresentam, para cada estratégia proposta, a economia de banda média de servidor, com respeito ao *Bandwidth Skimming*, como uma função da taxa de requisições  $N$ . Valores negativos indicam aumento percentual de banda em relação ao protocolo original. As figuras mostram, respectivamente, resultados típicos para perfis de carga IA, IM e IB e *buffer* de tamanho ilimitado.

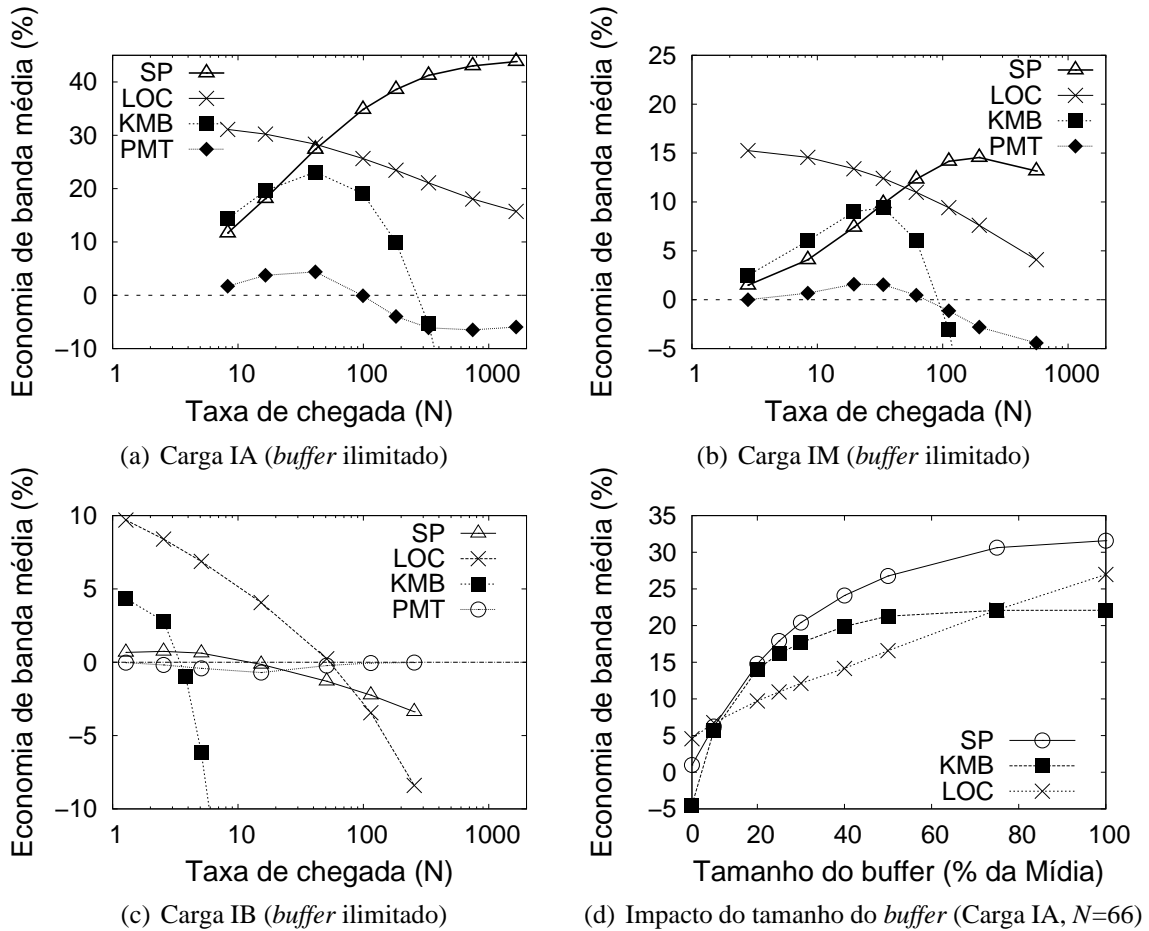


Figura 5.2: Economia de banda média de servidor com respeito ao *Bandwidth Skimming*

KMB, SP e PMT compartilham o mesmo comportamento geral para todas as cargas analisadas. Note-se que os dados armazenados no *buffer* de um cliente por essas estratégias dependem dos fluxos que estão sendo correntemente transmitidos e, portanto, de  $N$ . Para valores pequenos de  $N$ , a dispersão temporal é alta, havendo poucas oportunidades para armazenar dados e, com isso, de reduzir a retransmissão de dados pelo servidor. Apesar de limitado, o impacto dessa pequena redução sobrepõe o impacto da maior dispersão espacial introduzida, e a economia de banda cresce, embora modestamente. À medida que  $N$  cresce e mais dados são armazenados em *buffer*, o impacto da redução na retransmissão domina, e a economia de banda cresce. Após um certo valor de  $N$ , que varia conforme o protocolo e a carga, o impacto da dispersão espacial crescente passa a dominar e a sobrepôr os benefícios do uso de *buffers*: a economia de banda começa a cair<sup>3</sup>. De fato, para taxas de requisição altas, essas estratégias podem requerer banda média mais alta que a requerida pelo *Bandwidth Skimming*. Por outro lado, os dados armazenados por LOC no *buffer* de um cliente dependem apenas do comportamento do usuário que esse

<sup>3</sup>Para a carga IA (5.2(a)), a economia de banda para SP decresce para valores de  $N$  que estão acima do máximo mostrado no gráfico.

cliente atende. Portanto, os dados armazenados por LOC no *buffer* de um cliente dependem do comportamento interativo do cliente e *não*, de  $N$ . Então, à medida que  $N$  cresce, o impacto da dispersão espacial começa a dominar, e a economia de banda diminui.

A taxa de requisições na qual cada estratégia provê maior economia de banda depende primariamente da dispersão espacial da carga. Para cargas IA (figura 5.2(a)) e IM (figura 5.2(b)), as estratégias são capazes de prover economia, mesmo para taxas de requisição altas. Para cargas IB (figura 5.2(c)), o impacto negativo da fragmentação faz com que a banda média de servidor cresça, mesmo para valores baixos de  $N$ .

Em geral, SP – que recebe, como *prefetch*, segmentos da mídia próximos ao ponto que está sendo exibido – consegue maior economia de banda para *taxas de requisição relativamente elevadas* ( $N \approx 1000$ ) e para cargas com número elevado de requisições por sessão (figura 5.2(a)). LOC provê grande economia de banda para *taxas de requisições baixas* (conforme discutido acima) e para cargas com número elevado de requisições por sessão e com saltos freqüentes para trás (figura 5.2(b)). Por outro lado, KMB obtém seus melhores resultados para *taxas de chegada intermediárias*, mostrando um comportamento degenerativo acelerado para valores mais altos de  $N$ . Isso ocorre porque os *buffers* de KMB armazenam dados obtidos durante tentativas fracassadas de união. Segmentos armazenados em tentativas subseqüentes de união podem estar distantes uns dos outros, aumentando a fragmentação do *buffer* e das requisições efetivamente enviadas ao servidor (ver abaixo).

Finalmente, a economia modesta de banda obtida com PMT pode ser explicada por dois fatores principais. Em primeiro lugar, a estratégia opera apenas num subconjunto dos fluxos, ou seja, opera apenas nos fluxos recebidos por somente um cliente. Em segundo lugar, estender fluxos altera a estrutura original da árvore hierárquica de união de fluxos. Embora isso seja benéfico em alguns cenários, pode ser prejudicial aos fluxos que, de outra forma, seriam unidos e deixam de sê-lo na nova hierarquia formada. De forma mais geral, foram experimentadas diversas variações de protocolos que compartilham o princípio comum de alterar a estrutura de uniões a fim de evitar perdas. Todas essas variações obtiveram economia de banda muito modesta (quando a obtiveram) e, assim, não são compensadoras.

Em resumo, as estratégias KMB, LOC e SP permitem economia significativa de banda, cada uma delas para diferentes taxas de requisições. Além do mais, quanto mais interativa é uma carga, mas significativa é a economia de banda obtida. Em particular, a maior economia de banda é obtida com SP (42%) para a carga IA, na figura 5.2(a). A principal justificativa para essa observação é que as requisições de usuários nessas cargas já apresentam elevada dispersão espacial. Então, o impacto da dispersão espacial crescente torna-se pouco relevante. Além

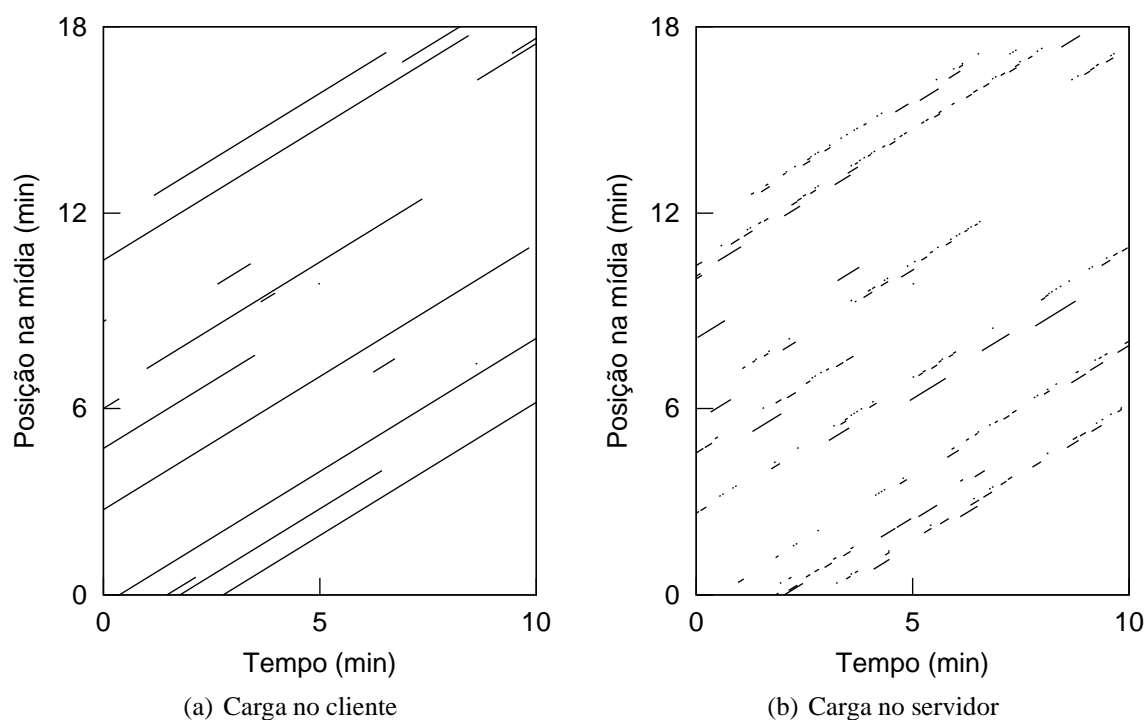


Figura 5.3: Fragmentação de requisições sob KMB (carga IA,  $N=66$ , *buffer* ilimitado)

disso, o número de requisições por sessão é mais alto em cargas de perfil IA que em cargas de outros perfis, o que favorece SP e LOC. Assim, essas estratégias são capazes de melhorar a escalabilidade do *Bandwidth Skimming* exatamente na região do espaço de projeto onde essa melhora é mais necessária, ou seja, para cargas de média e de alta interatividade, nas quais esta produz maior impacto sobre a escalabilidade de protocolos de mídia contínua (ver capítulo 4).

A seguir, o impacto da restrição de espaço para *buffer* em clientes é analisado. Essa análise é relevante porque, para mídias de longa duração e taxa de codificação elevada, a redução de espaço em *buffer* pode trazer economia significativa na implementação de um cliente. A figura 5.2(d) mostra a economia de banda média para LOC, SP e KMB para uma mesma carga IA,  $N = 66$  e tamanhos de *buffer* variados, expressos como uma fração da duração da mídia. As curvas para SP e KMB mostram claramente que o ganho obtido com o aumento do espaço em *buffer* decai à medida que o espaço em *buffer* cresce (existem *diminishing returns*). Se o *buffer* do cliente for restrito a 25%, KMB obtém ganhos próximos aos obtidos com *buffer* ilimitado. Situação similar ocorre com SP, que com *buffer* restrito a 40% obtém ganhos próximos aos obtidos com *buffer* ilimitado. Já a estratégia LOC sempre se beneficia de mais espaço em *buffer*, desde que o número de requisições por sessão seja elevado e os saltos para trás sejam frequentes.

A fim de ilustrar o fenômeno da fragmentação de sessões, discutido acima, a figura 5.3

---

**Algoritmo 4** Escolha gulosa de fluxo

---

- 1: {Esta escolha recebe como parâmetros a lista de fluxos ativos no servidor (incluindo o bloco que cada um desses fluxos está transmitindo), a lista de blocos ocupados no *buffer* local, e a indicação de qual foi o último bloco exibido.}
  - 2: Se não existir fluxo ativo, a escolha termina e não há fluxo a escutar.
  - 3: Se existir fluxo ativo, escolher o fluxo mais próximo do último bloco exibido e que transmita dados inéditos.
  - 4: Se tal fluxo não existir, escolher o fluxo que estiver mais próximo de transmitir dados inéditos.
- 

mostra um extrato (algumas das requisições) da carga IA já mostrada na figura 5.2(a),  $N = 66$  e *buffer* ilimitado no cliente. A figura 5.3(a) mostra a carga que os usuários enviam a seus programas clientes, e figura 5.3(b), as requisições efetivamente recebidas pelo servidor sob KMB. A carga no servidor contém um número muito maior de pequenas requisições dispersas espacialmente. Note-se que a fragmentação é um efeito geral do uso de *buffers*. Portanto, qualquer protocolo que se apóie no uso de *buffers* para armazenar dados no cliente (tal como em [55]) estará sujeito, em maior ou menor intensidade, aos efeitos da fragmentação. Contudo, dependendo de como é o uso do *buffer*, alguns protocolos, como o KMB, sofrem penalização mais forte. Outro efeito colateral da fragmentação de requisições é o aumento na sincronização requerida entre servidor e clientes, uma vez que o número de requisições aumenta. O custo extra de sincronização justifica-se se for compensado por uma economia significativa de banda (KMB para taxas de requisição intermediárias, por exemplo); do contrário, é necessário desenvolver algum esforço para reduzir a fragmentação de requisições.

## 5.4 Explorando o período inativo

A existência de períodos de inatividade relativamente longos entre as requisições de um usuário a uma mesma mídia (232 segundos em média nos 36 perfis de carga usados) motivou o projeto de SILENT PREFETCH (SP), estratégia que explora tais períodos para receber conteúdo antecipadamente e colocá-lo em *buffer* local ao cliente (seção 5.2). A presente seção explora mais detalhadamente o uso desses períodos de inatividade, apresentando duas novas estratégias que buscam contornar deficiências de SP. Assim, como SP, elas escutam um máximo de dois fluxos, seguindo uma configuração típica para o *Bandwidth Skimming* original (capítulo 2). O número de requisições por sessão tem impacto nessas estratégias, já que o conteúdo armazenado em um período inativo somente será usado em períodos ativos subsequentes. Essas estratégias, assim como qualquer estratégia que use *buffer* local ao cliente, estão sujeitas à fragmentação da carga imposta ao servidor (seção 5.3).

---

**Algoritmo 5** Escolha de fluxos na estratégia GP

---

- 1: Solicitar ao servidor a lista de fluxos ativos e o bloco que está sendo transmitido por cada fluxo.
  - 2: Escolher o primeiro fluxo a escutar, usando a ESCOLHA GULOSA DE FLUXO (Algoritmo 4), e remover o fluxo escolhido da lista de fluxos ativos.
  - 3: Usar novamente a ESCOLHA GULOSA DE FLUXO para escolher o segundo fluxo a escutar.
  - 4: **while** “usuário não interagir” **do**
  - 5:   **if** “um dos fluxos terminar” **then**
  - 6:     Solicitar ao servidor a lista de fluxos ativos e a posição que está sendo transmitida por cada fluxo.
  - 7:     Usar a ESCOLHA GULOSA DE FLUXO para escolher o novo fluxo a escutar.
  - 8:   **end if**
  - 9: **end while**
  - 10: Interromper a escuta de fluxos.
- 

### 5.4.1 Estratégias

GREEDY PREFETCH (GP) evita a escuta de conteúdo já armazenado, buscando povoar ao máximo o *buffer* local e, com isso, evitar a retransmissão mais fortemente que SP (já que SP não previne a escuta de conteúdo já armazenado). Dois fluxos são escolhidos independentemente (diferentemente de SP, um não precisa ser alvo do outro). Os fluxos escutados são os mais próximos do último ponto exibido da mídia (os fluxos escolhidos são os mais próximos, estejam eles à frente ou atrás do último ponto exibido, de forma a explorar a localidade de referência) e que transmitam dados inéditos. Caso não exista fluxo que transmita dados inéditos, a escolha recai sobre o fluxo que está mais próximo de transmitir conteúdo inédito. Essa escolha justifica-se por estar de acordo com a política de preenchimento agressivo do *buffer* local. Por fim, caso algum fluxo seja interrompido, um novo fluxo é escolhido, usando-se os mesmos critérios acima expostos.

O algoritmo do cliente que emprega a estratégia GP, embora similar ao de um cliente que empregue a estratégia SP (algoritmo 3), usa outros critérios para a escolha de fluxos a escutar nos períodos inativos. O processo de escolha de fluxos é esquematizado nos algoritmos 4 e 5. O algoritmo 4 apresenta a escolha de um fluxo, conforme descrição acima. O algoritmo 5 mostra como esse procedimento é empregado para a escuta de dois fluxos simultâneos.

A figura 5.1 exemplifica a escolha de fluxos feita por GP. Os fluxos escutados são *D* e *B*, nessa ordem (*C* não é escolhido porque transmite conteúdo já armazenado). GP não previne a fragmentação do *buffer*, pois um fluxo pode preencher parcialmente uma área vazia e criar novos fragmentos. É o que acontecerá na figura 5.1 se o fluxo *B* terminar antes de alcançar o ponto *P*. Se não há fluxo disponível com conteúdo inédito, escolhe-se o que transmite o mínimo



**Algoritmo 6** Escolha cooperativa de fluxo

- 
- 1: {Esta escolha recebe como parâmetros a lista de fluxos ativos no servidor (incluindo a posição que cada um desses fluxos está transmitindo), a lista de posições ocupadas no *buffer* local e a indicação de qual foi o último bloco exibido.}
  - 2: Se não existir fluxo ativo, a escolha termina e não há fluxo a escutar.
  - 3: Se existir fluxo ativo, escolher o fluxo mais próximo do último bloco exibido e que transmita dados no início de uma área vazia do *buffer* ou no meio de uma área ocupada.
  - 4: Se tal fluxo não existir, nenhum fluxo é escolhido.
- 

de conteúdo já armazenado.

COOPERATIVE PREFETCH (CP) tem como objetivo evitar a fragmentação permitida por GP. O primeiro fluxo é escolhido, seguindo os mesmos critérios de GP e objetivando o preenchimento agressivo do *buffer* local. Já o segundo é escolhido de forma a minimizar a fragmentação e, para tanto, nunca se escolhem fluxos cujo conteúdo ocupe o meio de uma área vazia (como o fluxo *B* na figura 5.1). Em lugar disso, escolhe-se o fluxo mais próximo do último ponto exibido e que esteja no início de uma área vazia ou no meio de uma área já ocupada. Dessa forma, caso o fluxo termine antes de preencher toda a área, garante-se que ela não será dividida e sim, apenas reduzida. No exemplo da figura 5.1, o primeiro fluxo escolhido será *D*, e o segundo, *A*. Os dois fluxos cooperam entre si, um preenchendo o *buffer*, o outro removendo a fragmentação. Mas, como o segundo fluxo não previne receber conteúdo já armazenado, o preenchimento do *buffer* é menos agressivo que em GP.

**Algoritmo 7** Escolha de fluxos na estratégia CP

- 
- 1: Solicitar ao servidor a lista de fluxos ativos e a posição que está sendo transmitida por cada fluxo.
  - 2: Escolher o primeiro fluxo a escutar, usando a ESCOLHA GULOSA DE FLUXO, e remover o fluxo escolhido da lista de fluxos ativos.
  - 3: Usar a ESCOLHA COOPERATIVA DE FLUXO (Algoritmo 6) para escolher o segundo fluxo a escutar.
  - 4: **while** “usuário não interagir” **do**
  - 5:   **if** “um dos fluxos terminar” **then**
  - 6:     **if** “o primeiro fluxo terminou” **then**
  - 7:       Usar a ESCOLHA GULOSA DE FLUXO para escolher o novo fluxo a escutar, e remover o fluxo escolhido da lista de fluxos ativos.
  - 8:     **end if**
  - 9:     **if** “o segundo fluxo terminou” **then**
  - 10:       Usar a ESCOLHA COOPERATIVA DE FLUXO para escolher o novo fluxo a escutar.
  - 11:     **end if**
  - 12:   **end if**
  - 13: **end while**
  - 14: Interromper a escuta de fluxos.
-

O algoritmo 6 apresenta a escolha do segundo fluxo, conforme descrição acima. O algoritmo 7 mostra como esse procedimento e também o do algoritmo 4 são empregados para a escuta de dois fluxos simultâneos.

### 5.4.2 Avaliação

Esta seção apresenta uma avaliação quantitativa das novas estratégias GP e CP e compara-as com a estratégia prévia de período inativo SP. Para tanto, o mesmo simulador usado nas seções anteriores foi estendido para a avaliação de GP e SP. Assim como os outros resultados, os aqui apresentados são a média de cinco execuções, com desvio padrão que não excede a 2% dessa média.

A medida usada para avaliar a escalabilidade das estratégias propostas é a economia de banda do servidor em relação ao protocolo *Bandwidth Skimming* original. A figura 5.4 resume os principais resultados obtidos para as três estratégias e para diferentes níveis de interatividade. O eixo  $x$  apresenta valores crescentes da taxa de chegada de requisições normalizada  $N$ . O eixo  $y$  mostra a economia de banda obtida comparativamente ao protocolo original. Valores acima de 0 indicam economia de banda. Resultados para outras cargas da mesma classe são qualitativamente similares aos apresentados.

As curvas para as três estratégias apresentam um comportamento geral similar. Para valores mais baixos de taxa de chegada de requisições  $N$  a dispersão temporal é alta, e, conseqüentemente, há poucas oportunidades para união de fluxos. O pouco conteúdo que venha a ser armazenado pelo cliente já permite redução na retransmissão e leva a alguma economia de banda de servidor. À medida que a taxa de chegada cresce, mais conteúdo é armazenado. Como resultado, o impacto da redução de retransmissão domina, e a economia obtida cresce. A partir de certo valor de  $N$  a dispersão espacial introduzida na carga entregue ao servidor domina, e a economia de banda começa a decrescer (seção 5.3). O ponto em que o pico ocorre depende da carga e da estratégia.

Um aspecto notável da estratégia GP é verificado quando se comparam as figuras 5.4(a) e 5.4(b). Na primeira, GP obtém economia significativamente superior à de SP, enquanto que a situação se inverte na segunda. Essa diferença deve-se ao preenchimento agressivo do *buffer* presente nas estratégias gulosas. Na carga da figura 5.4(b) a duração média das requisições é superior em 25% ao tempo médio de inatividade. Com requisições relativamente longas, GP consegue ocupar 90% do tempo disponível com escuta de conteúdo, mesmo com taxas tão baixas como  $N = 48$  (100% do tempo é ocupado com  $N = 291$ ). Com isso, a capacidade de GP de armazenar conteúdo e, portanto, de evitar retransmissão e economizar banda esgota-

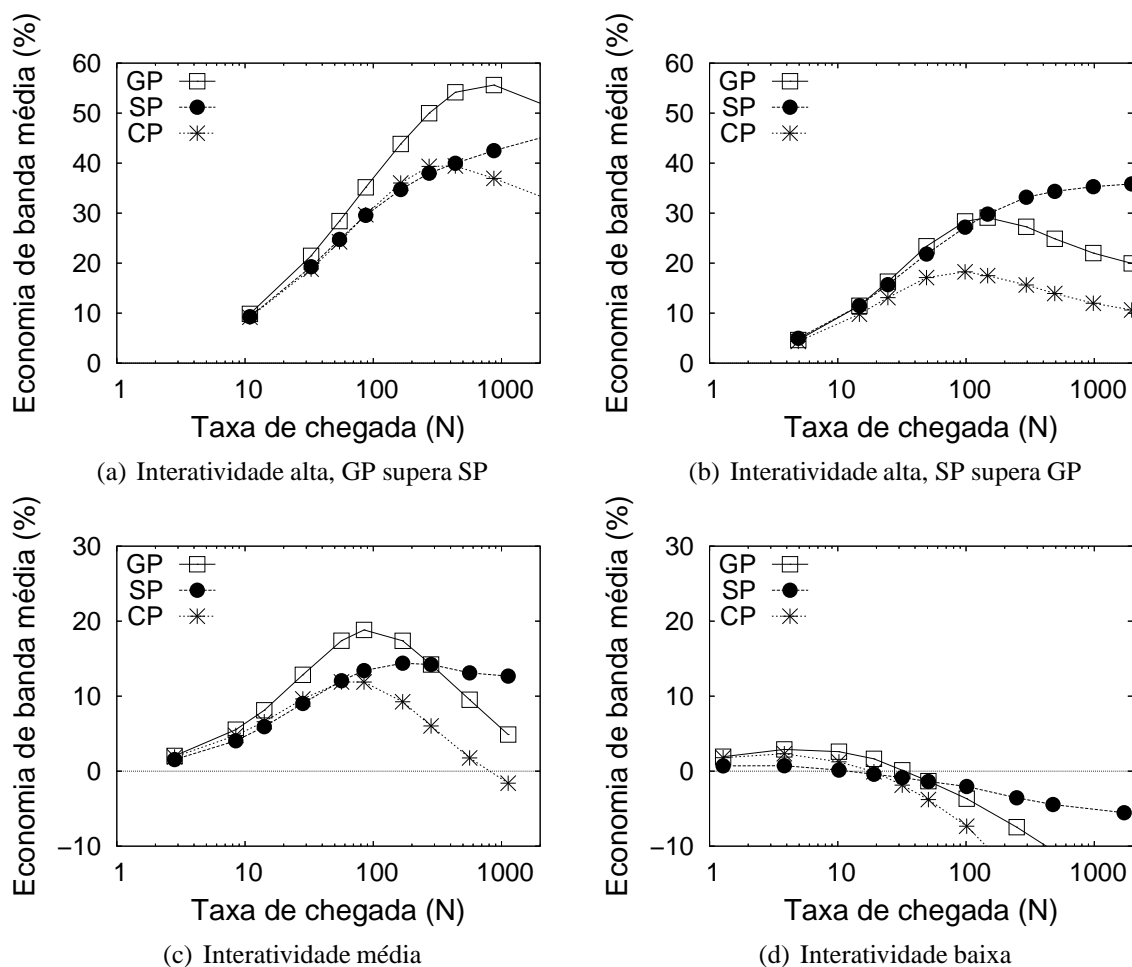


Figura 5.4: Economia de banda de servidor para diferentes níveis de interatividade

se a taxas de chegada relativamente baixas. Já SP, que é mais restrita na escolha de fluxos, somente chega a ocupar 90% do tempo de inatividade com escuta de conteúdo com  $N = 486$  (uma avaliação do aproveitamento do tempo de inatividade para a escuta de fluxos é vista mais adiante). Portanto, SP consegue economia de banda crescente, mesmo para taxas de chegada mais altas. Essa situação é similar em todas as cargas nas quais a duração média das requisições excede o tempo médio de inatividade.

SP não sofre fortemente o efeito degenerativo da fragmentação, e, portanto, sua economia de banda somente decresce para taxas de chegada mais altas (figura 5.4(b)). É a estratégia que permite economia para faixa mais ampla de taxas de chegada. GP é mais agressiva ao preencher o *buffer* e obtém economia superior a qualquer outra estratégia (figuras 5.4(a) e 5.4(c)). Como GP está sujeita ao efeito degenerativo da fragmentação, a economia de banda cai abruptamente. CP não se mostrou efetiva para reduzir a fragmentação causada pelo preenchimento agressivo do *buffer*. A economia obtida com o preenchimento agressivo é tão elevada, que GP sempre supera a economia obtida com CP (figura 5.4).

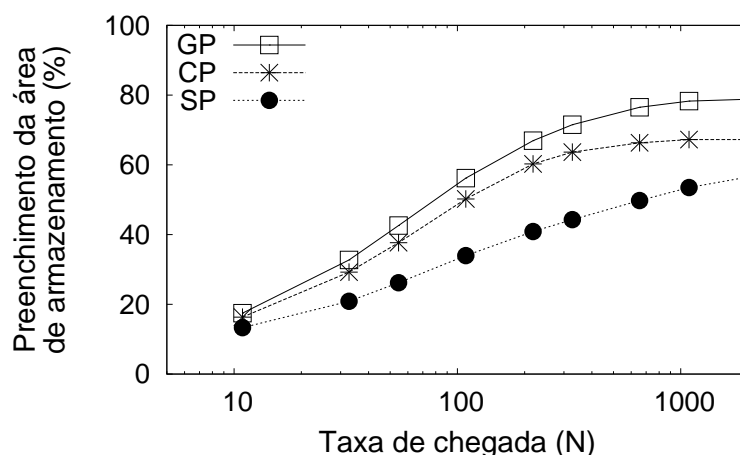


Figura 5.5: Preenchimento do *buffer* (Carga IA)

A figura 5.5 ilustra o preenchimento agressivo do *buffer* de GP, mostrando o percentual médio de preenchimento do *buffer* local dos clientes para diversas taxas de chegada, para a carga da figura 5.4(a). A capacidade de GP de economizar banda depende justamente desse preenchimento agressivo. Enquanto o percentual de preenchimento cresce rapidamente com a taxa de chegada, a economia de banda cresce. Para taxas mais altas, o preenchimento cresce mais lentamente. A economia de banda atinge então seu máximo e começa a decrescer. CP não previne a escuta de conteúdo já armazenado, o que limita sua capacidade de armazenamento. No caso de SP, que não previne a escuta de dados armazenados e que somente escolhe fluxos que transmitam dados à frente do último bloco recebido, a capacidade de preencher o *buffer* é ainda mais reduzida, conforme mostrado na figura 5.5.

Para cargas IM os resultados são similares, mas, como essas cargas são menos dispersas, o impacto da fragmentação é mais forte, e a economia de banda, mais reduzida. Para cargas IB, ainda menos dispersas que as cargas IM, o impacto é ainda mais significativo.

A figura 5.6 mostra como cada política usa o tempo de inatividade para a escuta de fluxos. O percentual do tempo de inatividade efetivamente usado para escuta de fluxos é mostrado como uma função da taxa de chegada  $N$ , para a carga da figura 5.4(b). GP e CP, com critérios para a escolha de fluxos menos restritivos que os de SP, permitem usar um percentual maior do tempo inativo para escuta; em contrapartida, chegam a ocupar completamente o tempo disponível para escuta (99% do período de inatividade com  $N = 40$ ), e, a partir desse ponto, a economia obtida pelas otimizações deixa de crescer, já que se torna impossível obter mais conteúdo para o *buffer*. Nessa situação, SP segue usando percentuais crescentes do tempo de inatividade, e sua economia sobrepuja a de GP. Isto ocorre nas cargas em que a duração média das requisições é grande se comparada ao tempo de inatividade (figura 5.4(b)). Com requisições relativamente longas

(25% maiores que o tempo médio de inatividade), GP já consegue escutar fluxos suficientes para ocupar 90% do tempo disponível, mesmo com taxas tão baixas como  $N = 48$ .

A fragmentação criada pelos protocolos é ilustrada na figura 5.7, que mostra, para a carga da figura 5.4(a) e otimização SP, o número médio e o tamanho médio (como um percentual da duração da mídia) dos segmentos no *buffer* dos clientes. A fragmentação cresce com a taxa de chegada, até que o *buffer* se torna tão preenchido, que a fragmentação começa a diminuir.

Outras estratégias foram exploradas no intuito de reduzir a fragmentação, porém com economia muito modesta. Exigir que os fluxos a escutar não causem fragmentação limita seriamente o repertório de fluxos disponíveis e, portanto, a quantidade de dados armazenados. A redução de retransmissão e a economia obtida são, nesses casos, pequenas.

Até aqui, este trabalho considerou que o cliente dispunha de *buffer* suficiente para comportar toda a mídia. O impacto da restrição dessa área é avaliado a seguir. A figura 5.8 mostra o impacto da restrição do tamanho do *buffer* para as duas estratégias mais eficientes, SP e GP ( $N = 652$ , carga da figura 5.4(a)). No eixo  $x$ , tem-se o tamanho da área como um percentual da duração da mídia; no eixo  $y$ , tem-se a economia de banda sobre o *Bandwidth Skimming* original. Com área restrita a 40% da duração da mídia, SP apresenta economia de banda próxima àquela obtida quando não há restrições. Já GP, que preenche a área mais agressivamente, é capaz de tirar partido do aumento da área disponível. Em síntese, estratégias gulosas, como GP, exigem que o cliente disponha de área suficiente para armazenar toda a mídia, ou os benefícios obtidos pela estratégia serão limitados. Já SP permite reduzir a área de armazenamento disponível no cliente, sem com isso reduzir significativamente a economia de banda obtida.

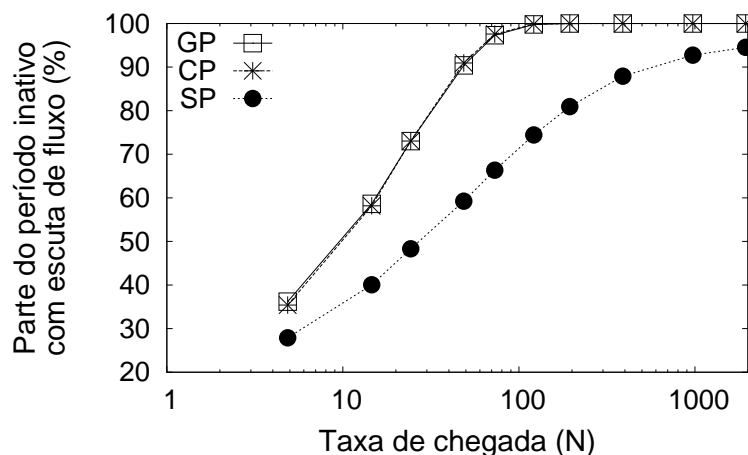
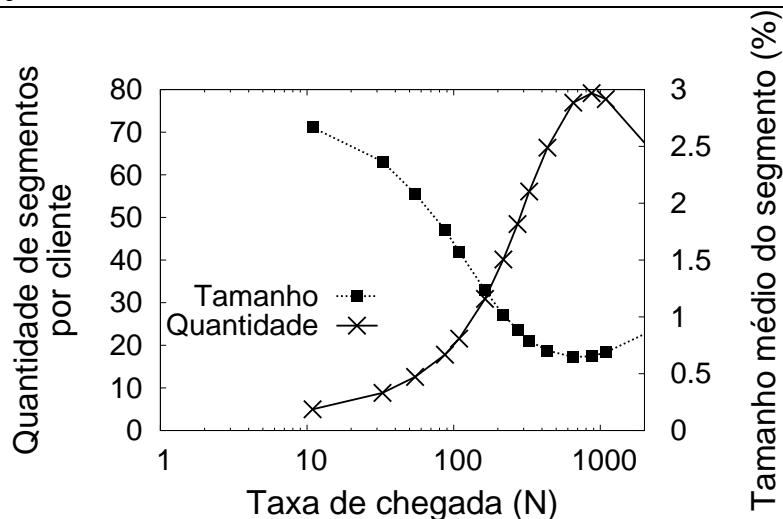
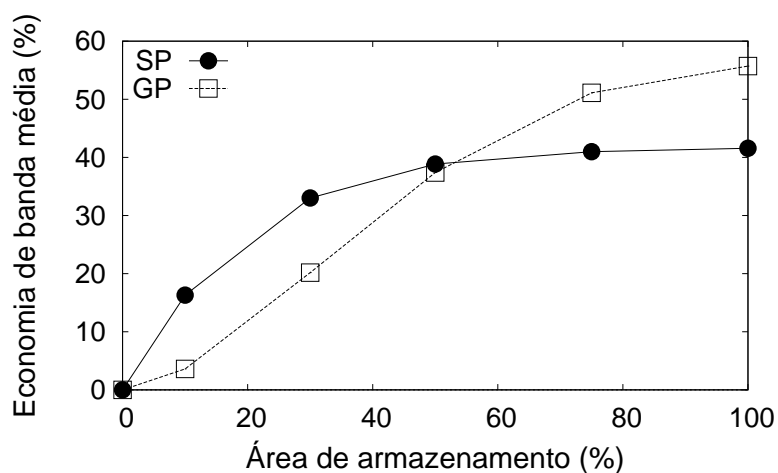


Figura 5.6: Porcentagem do período inativo usada para escutar fluxos

Figura 5.7: Fragmentação do *buffer* de SP, carga IAFigura 5.8: Impacto da restrição no *buffer* (Carga IA,  $N=652$ )

### 5.4.3 Aspectos de implementação

Assim como em SP, no processo de seleção de fluxos usado por GP e CP faz-se necessário enviar, do servidor ao cliente, informação sobre fluxos disponíveis para escuta. Mas, como o número de fluxos ativos é limitado (seção 2.1.3), torna-se viável enviar mensagem do servidor ao cliente com a lista de fluxos disponíveis, de forma que o próprio cliente selecione os mais adequados. Embora este trabalho se refira ao *buffer* local ao cliente, este poderia se situar num agente, de forma que o cliente dispensasse qualquer modificação. Por fim, muitas das estratégias propostas poderiam ser aplicadas, sem grande esforço, a outros protocolos de compartilhamento de banda.

## 5.5 Estratégias híbridas

Esta seção leva adiante a exploração do espaço de projeto de protocolos de mídia contínua, avaliando estratégias híbridas que combinam esforços das estratégias individuais mais competitivas, LOC, KMB, SP, CP e GP. As estratégias híbridas reduzem significativamente a banda de servidor ao longo de uma região do espaço de projeto mais ampla que a de qualquer técnica individual.

### 5.5.1 Estratégias

Estratégias híbridas combinam esforços de diferentes técnicas com o intuito de explorar as diferentes características de cada uma delas. De um lado, buscam-se estratégias que atuem sobre uma faixa mais ampla de taxas de chegada; de outro, estratégias com economia de banda mais expressiva, ainda que em cenários mais restritos.

HYBRID SYNERGIC (HS) é uma estratégia que combina as técnicas mais bem sucedidas no período ativo, LOC e KMB, com a técnica de período inativo menos sujeita à fragmentação, SP. O objetivo dessa estratégia híbrida é usar SP para minimizar o efeito degenerativo da fragmentação introduzido por KMB (seção 5.3).

HYBRID CONSERVATIVE (HC) combina apenas técnicas menos sujeitas ao efeito degenerativo da fragmentação, SP e LOC, a fim de obter economia de banda em faixas amplas de taxas de chegada. Em contrapartida, essa escolha limita a economia de banda máxima que a estratégia pode obter.

HYBRID AGGRESSIVE (HA) objetiva o preenchimento agressivo do *buffer*, como forma de maximizar a economia de banda por redução de retransmissão. Para tanto, combina LOC, que armazena todo o conteúdo exibido, KMB, que evita o descarte do conteúdo do *merge buffer*, e GP, que dá preferência ao armazenamento de conteúdo inédito. Em contrapartida, essa estratégia potencializa o efeito da fragmentação.

### 5.5.2 Avaliação

O mesmo simulador usado nas seções 5.3 e 5.4.2 foi estendido para permitir a combinação de estratégias. A figura 5.9 apresenta a economia de banda média de servidor em relação ao *Bandwidth Skimming* original para as estratégias híbridas para cargas IA, IM e IB típicas.

HC tem sucesso ao combinar técnicas que não apresentam efeito degenerativo acentuado.

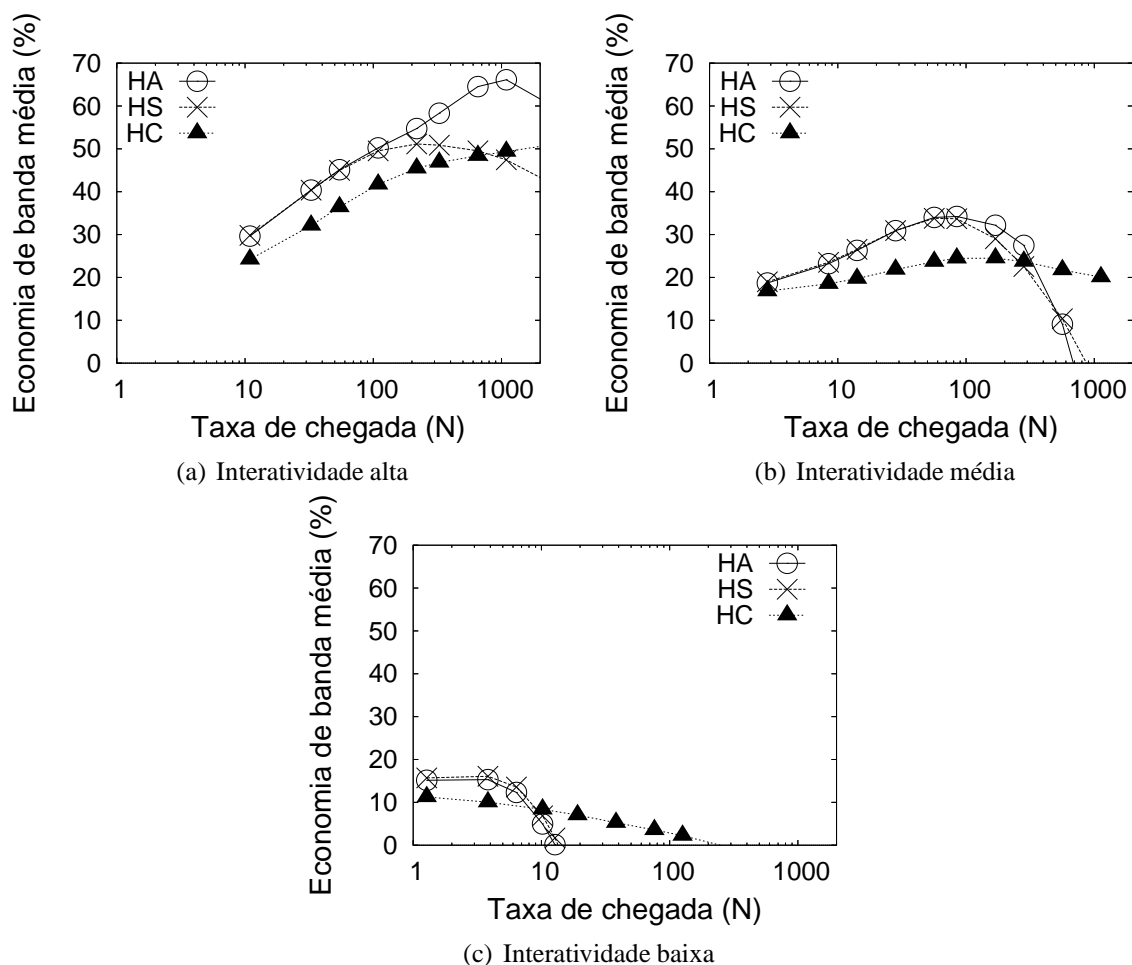


Figura 5.9: Economia de banda de servidor para estratégias híbridas

A economia de banda obtida excede a economia das estratégias individuais, e a faixa de taxas de chegada na qual se obtém economia significativa é ampliada. Das três estratégias híbridas avaliadas, HC é a menos sujeita ao efeito degenerativo da fragmentação. É o que se pode ver nas figuras 5.9(a), 5.9(b) e 5.9(c): nelas as outras estratégias híbridas, HA e HS, apresentam queda de economia de banda significativamente mais acentuada que HC à medida que a taxa de chegadas aumenta.

HA obtém melhor economia de banda que HC, especialmente para taxas de chegada mais baixas, já que armazena conteúdo de forma agressiva e tem mais sucesso ao evitar retransmissões. No entanto, HA é mais sujeita ao efeito degenerativo da fragmentação, motivado pelas estratégias agressivas, GP e KMB, que emprega. O efeito degenerativo, mais evidente nas figuras 5.9(b) e 5.9(c) também está presente na figura 5.9(a), porém para taxas de chegadas mais altas que as nela mostradas.

HS falha na tentativa de usar SP e LOC a fim de suprimir o efeito degenerativo da fragmentação herdado de KMB. Além disso, o volume de dados armazenados no *buffer* local por SP e



LOC, que não são estratégias gulosas, não é suficiente para obter tanta economia com redução da retransmissão quanto HA, que substitui a estratégia conservativa SP pela estratégia gulosa GP.

Esse efeito degenerativo, detectado nas estratégias individuais (seções 5.3 e 5.4) e que não é contornado completamente por nenhuma das estratégias híbridas, dá ensejo ao projeto de um protocolo adaptativo que monitore a carga e decida pelo preenchimento mais ou menos agressivo do *buffer*, ou mesmo por não usá-lo e que é objeto do capítulo 7.

O impacto da restrição de espaço no *buffer* é avaliado a seguir. Assim como na seção 5.2, a localidade de referência guia a gerência do *buffer* do cliente. Os eventuais conflitos entre conteúdo obtido por diferentes otimizações são resolvidos, dando-se preferência para armazenamento ao conteúdo mais próximo do último segmento de mídia exibido. A figura 5.10 mostra o impacto da restrição do tamanho do *buffer* para as estratégias híbridas. No eixo *x*, tem-se o tamanho da área como um percentual da duração da mídia, e, no eixo *y*, tem-se a economia de banda sobre o *Bandwidth Skimming* original. Para cargas IA e  $N = 652$  (figura 5.10(a)), HA preenche a área agressivamente e tira partido do aumento da área disponível para armazenamento. HC e HS não preenchem a área de forma tão agressiva e obtêm, com *buffer* restrito a 50% da duração da mídia, economia de banda próxima à obtida sem restrições nessa área. A figura 5.10(b) mostra, para a estratégia HA e uma carga IA, o impacto da restrição de área em diversas taxas de chegada. HA, que preenche o *buffer* agressivamente, é fortemente afetada pela restrição do *buffer*. Para taxas de chegada mais altas, mesmo uma pequena redução no *buffer* tem forte impacto na economia de banda obtida pela estratégia.

Em síntese, similarmente ao que ocorre para estratégias individuais (seções 5.3 e 5.4), estratégias agressivas no preenchimento do *buffer* (HA) obtêm economias de banda superiores, mas

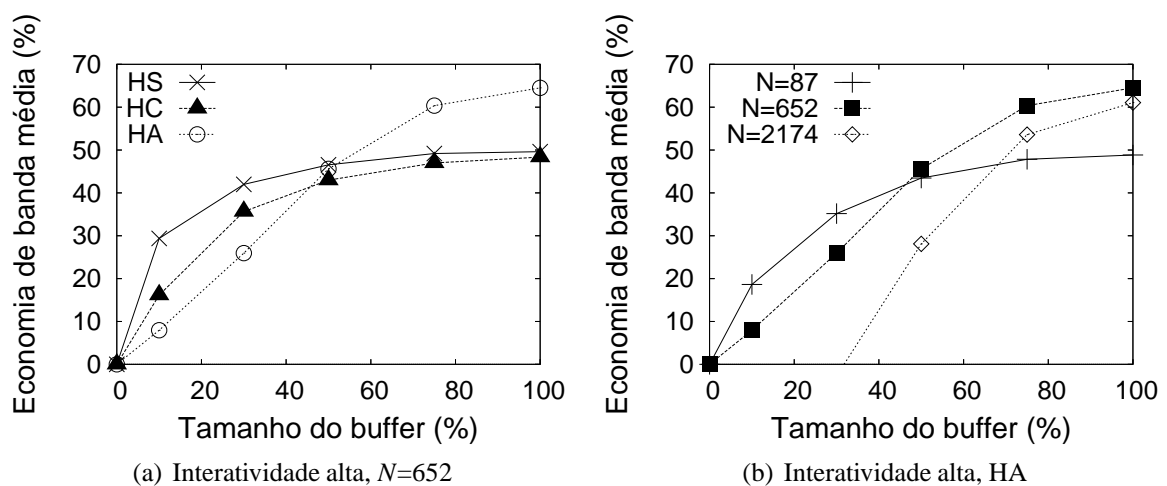


Figura 5.10: Impacto da restrição do *buffer*

exigem que o cliente disponha de área suficiente para armazenar toda a mídia; caso contrário, seus benefícios serão muito limitados. Já estratégias menos agressivas (HS, HC) permitem ao cliente operar com *buffer* reduzido a 50% da duração da mídia e, ainda assim, obter economia de banda expressiva. Como a estratégia sinérgica HS falha em suprimir ou mesmo em minimizar o efeito degenerativo da fragmentação, conclui-se que ela pode ser deixada de lado, usando-se HA para taxas de chegadas baixas e médias, através da qual se obtém economia de banda mais expressiva, e HC para taxas de chegadas mais altas, uma vez que é a estratégia que menos sofre impacto do efeito degenerativo da fragmentação.

### 5.5.3 Estratégias híbridas como envelopes das estratégias individuais que as integram

Uma estratégia híbrida emprega, simultaneamente, duas ou mais estratégias individuais (ver seções anteriores deste capítulo), a fim de obter conteúdo e armazená-lo em área local ao cliente, com o objetivo de reduzir retransmissão. Portanto, torna-se natural esperar que uma estratégia híbrida obtenha maior redução de retransmissão do que qualquer uma das estratégias individuais que ela empregue. No entanto, é perfeitamente possível que, num certo cenário, uma estratégia híbrida, ao ocupar a área local de forma diferente e possivelmente mais fragmentada que a ocupada por uma das estratégias individuais, incorra num impacto mais forte da fragmentação de requisições que essa estratégia individual incorre.

A figura 5.11 ilustra dois desses cenários, mostrando a economia de banda média de servidor para duas estratégias híbridas (HA e HC) e para as estratégias individuais que as compõem. Na figura 5.11(a) são mostradas a estratégia HA e as estratégias GP, KMB e LOC, que a in-

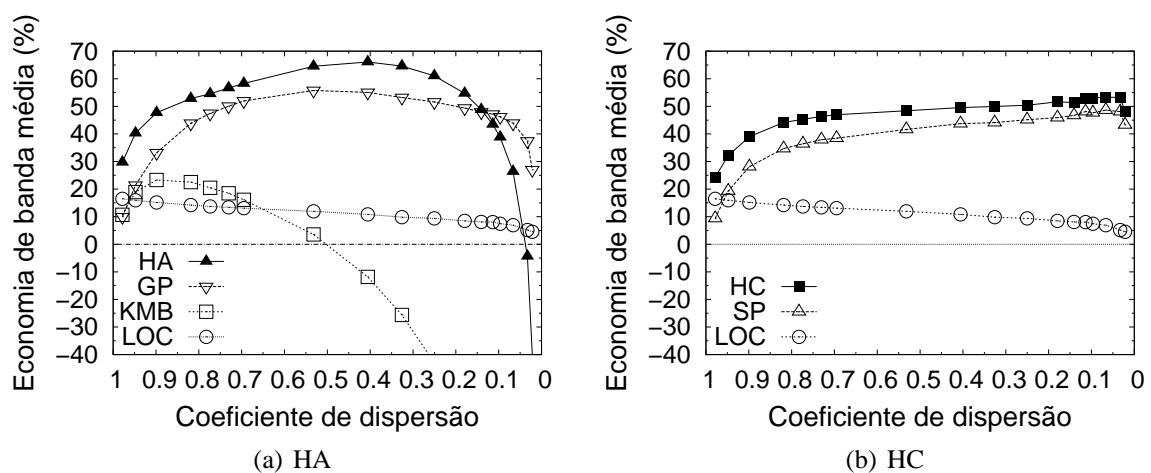


Figura 5.11: Economia de banda com cargas de vídeo educacional para duas estratégias híbridas e para as estratégias individuais que elas empregam

tegram. Embora atinja economia de banda média de servidor que supera a de qualquer das estratégias individuais, à medida que a dispersão cai e aproxima-se de 0,1 HA torna-se menos econômica que GP e, a seguir, menos econômica que LOC. Já na figura 5.11(b) são mostradas a estratégia HC e as estratégias SP e LOC, que a integram. HC atinge economia superior à de SP e à de LOC e, em momento algum, é menos econômica que qualquer uma delas.

#### 5.5.4 Aspectos de implementação

A implementação das políticas híbridas propostas não é significativamente mais complexa que a das políticas individuais que as integram. Isso ocorre porque os eventuais conflitos entre conteúdo obtido por diferentes estratégias são prontamente resolvidos pelo cliente, dando-se preferência para armazenamento ao conteúdo mais próximo do último segmento de mídia exibido, o que simplifica enormemente as interações entre as estratégias integrantes da híbrida.

**Conclusão:** Este capítulo apresenta diversas técnicas capazes de obter significativa economia de banda de servidor com relação ao *Bandwidth Skimming* original (até 66%). No entanto, nenhuma das técnicas pode ser aplicada indistintamente a qualquer carga. Isso motiva o projeto da estratégia adaptativa apresentada em capítulo subsequente. As técnicas híbridas aqui apresentadas (e também as estratégias de período inativo) podem trazer impacto para as bandas de cliente e de rede. No capítulo seguinte é mostrada uma análise preliminar desse impacto, com a finalidade de identificar os principais compromissos nele envolvidos.

## 6 *Requisitos de banda de cliente e de rede*

As estratégias de período inativo – SILENT PREFETCH (SP), GREEDY PREFETCH (GP) e COOPERATIVE PREFETCH (CP) – requerem que um cliente escute no máximo *dois* fluxos durante seus períodos silenciosos (ver capítulo 5). Portanto, a banda que estaria disponível para *outros usos* que não a transferência de fluxos (para usar o navegador ou para transferir arquivos, por exemplo) pode ser reduzida.

Similarmente, também há impacto sobre a banda de rede. Se os clientes que estiverem executando essa escuta silenciosa escutarem fluxos que, de outra forma, não estariam sendo enviados para sua região da rede, estes fluxos deverão percorrer canais de rede adicionais, e assim novos fluxos serão criados na rede.

Com a finalidade de identificar os principais compromissos envolvidos no impacto das estratégias de período inativo sobre a banda de rede e a banda de servidor, este capítulo faz um estudo preliminar do impacto de SP na banda do cliente (seção 6.1) e na banda de rede (seção 6.2), usando para isso, topologias canônicas. Esses compromissos são usados, a seguir, para avaliar o impacto das estratégias híbridas na banda de rede em uma topologia real (seção 6.3).

### 6.1 **Banda de cliente para a estratégia SP**

A fim de aliviar o impacto de SP sobre a banda de cliente, mas ainda obter os benefícios da economia de banda que tal estratégia oferece, considera-se uma variante do protocolo, na qual o cliente escuta apenas *um* fluxo (o fluxo mais próximo) durante os períodos silenciosos. Esse protocolo é aqui chamado *SP-Conservative*, e a estratégia original é renomeada para *SP-Aggressive*.

A figura 6.1 mostra a economia de banda média de servidor, com respeito ao *Bandwidth Skimming*, para ambas as variações do protocolo, para uma carga IA típica, e com *buffer* ilimitado no cliente. São mostrados os resultados apenas para uma carga IA para ilustrar os principais

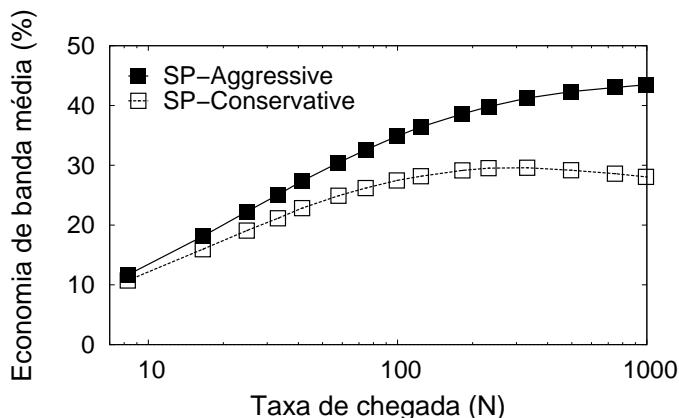


Figura 6.1: SILENT PREFETCH - Economia de banda de servidor com respeito ao *Bandwidth Skimming* (*buffer* ilimitado no cliente)

compromissos, já que SP tem seu melhor desempenho para esta classe de perfil de carga.

O que se paga para obter uma redução de 50% na banda de cliente é uma penalidade de até 30% nos requisitos de banda média de servidor para as cargas e taxas de requisição analisadas. Vendo-se a situação por outro ângulo, enquanto *SP-Aggressive* provê economia de banda de até 42%, a maior economia obtida por *SP-Conservative* é de 28%.

Para espaço limitado de *buffer* de cliente, as economias são modestas, como esperado: para um *buffer* de 25% (50%) da duração da mídia, *SP-Conservative* reduz a banda média de servidor em até 15% (23%), em comparação com a economia de 18% (30%) obtida por *SP-Aggressive*, para a mesma carga IA.

## 6.2 Banda de rede para a estratégia SP

Esta seção apresenta uma análise preliminar da banda de rede requerida pela estratégia SP. A meta dessa análise é prover alguns resultados iniciais com respeito aos compromissos primários, do ponto de vista da rede. Uma análise mais detalhada é efetuada em [57].

Para a análise aqui proposta faz-se uso de uma topologia canônica, como a mostrada na Figura 6.2. Nessa topologia, tem-se um nó servidor conectado a  $k$  sítios clientes por meio de um único canal compartilhado conectado a mais  $k$  canais disjuntos, cada um servindo um sítio cliente distinto. Cada sítio cliente representa um grupo de clientes próximos. Além disso, um canal simples é usado para representar a árvore de distribuição que conecta o canal compartilhado a cada um dos usuários de um mesmo sítio. Essa topologia, embora seja uma abstração do sistema real, permite capturar, num primeiro nível, os fatores dominantes que impactam os

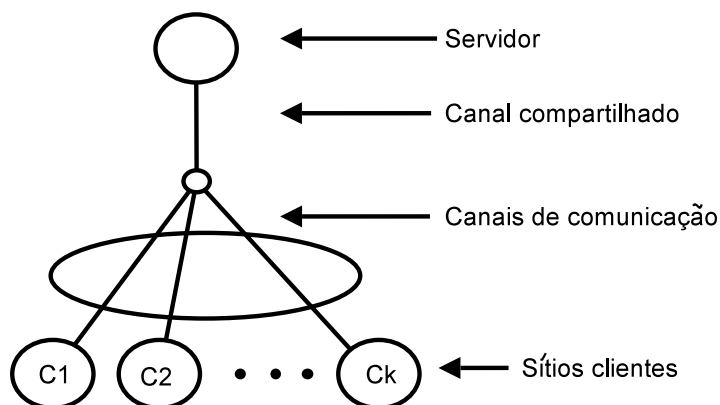


Figura 6.2: Modelo da topologia canônica

requisitos de banda de rede.

Na análise aqui efetuada, assume-se que o canal compartilhado está dentro da rede do servidor. Assim, a banda média de rede é dominada pela banda média requerida nos  $k$  canais disjuntos. Também se assume que a taxa de requisição  $N$  é homogeneamente distribuída entre todos os sítios clientes. A topologia selecionada e a distribuição de carga são o pior caso com respeito à escalabilidade da banda de rede com relação ao número de clientes [84] e, em particular, com respeito a como SP impacta a banda de rede.

A figura 6.3 mostra a economia na banda média de rede para ambas as variantes do SP, com relação ao *Bandwidth Skimming*, para uma topologia com  $k = 5$  sítios, uma carga IM típica, a mesma carga IA da figura 6.1, para diferentes dimensões do *buffer* local. Para calcular a banda média de rede, soma-se o número médio de fluxos que percorrem cada um dos  $k$  canais de comunicação. Como exemplo, se na figura 6.2 tivermos em média um fluxo para cada um dos sítios clientes, a banda média de rede será  $k$ . Assim, essa banda depende: (a) do número de fluxos de servidor percorrendo a rede e (b) do número de fluxos de rede criados para compartilhar os fluxos entre múltiplos sítios.

Para cargas IA, a grande economia de banda alcançada pela estratégia SP (figura 6.3(a)) sobrepuja a banda de rede, que é crescente devido ao compartilhamento de fluxos intersítios. Para a carga da figura 6.3(a) e  $k = 5$ , SP-Aggressive reduz a banda média de rede em até 36%, comparada com o *Bandwidth Skimming*. Maior (menor) economia é alcançada com menores (maiores) valores de  $k$ : para a mesma carga IA e  $N = 500$ , SP-Aggressive reduz a banda média de rede em até 40% para  $k = 2$  e em até 16% para  $k = 10$ . Assim, a otimização provê economia significativa de banda de rede mesmo para valores elevados de  $k$ , desde que a taxa de requisição originada de um mesmo sítio seja suficientemente elevada para permitir um número significativo de uniões de fluxos locais.

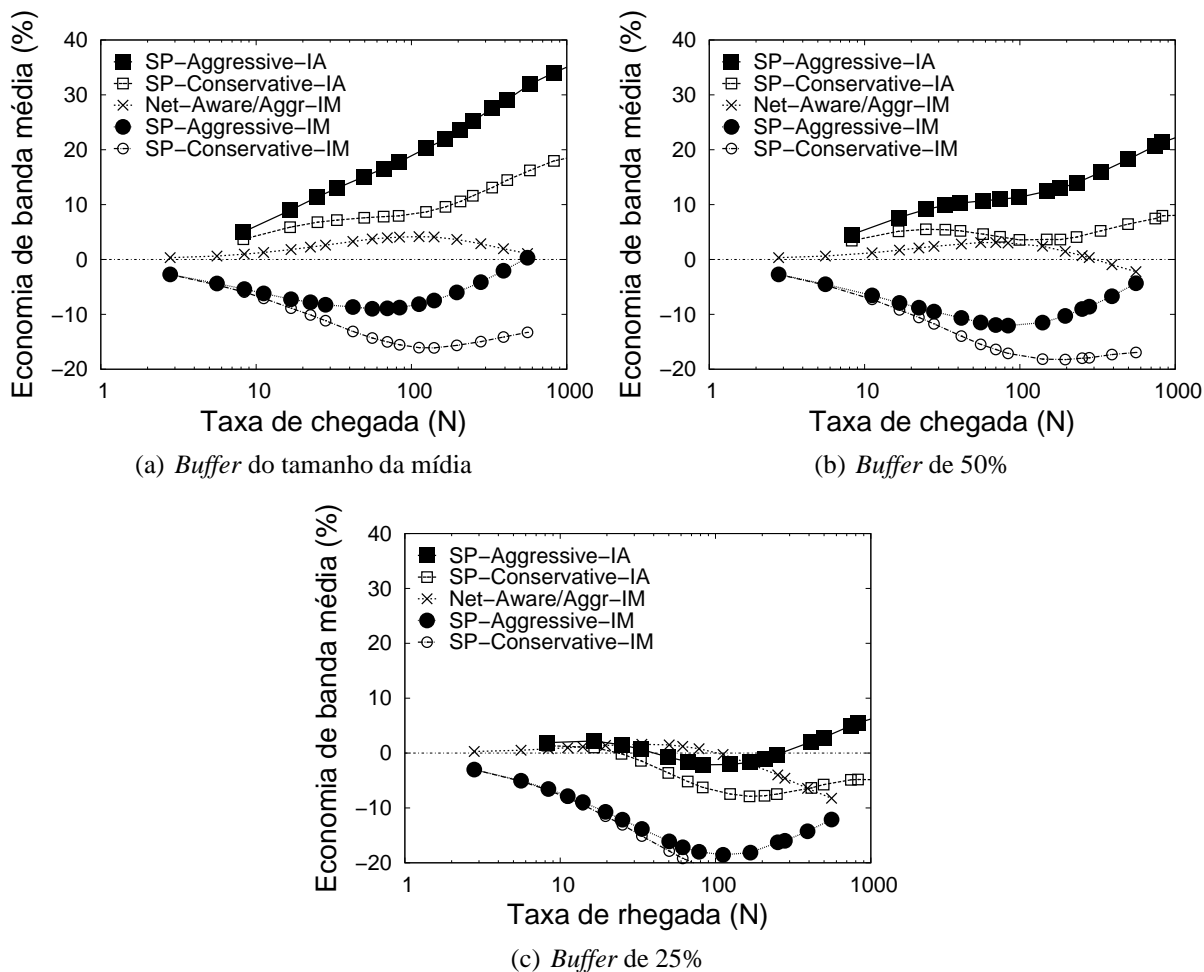


Figura 6.3: SILENT PREFETCH - Economia de banda média de rede com respeito ao *Bandwidth Skimming* com diferentes tamanhos de *buffer* ( $k=5$ )

Para cargas IM e IB, o crescimento no número de fluxos de rede domina e sobrepõe a modesta economia de banda de servidor alcançada com as otimizações SP. A banda média de rede requerida por SP-Aggressive chega a ser 9% superior à banda requerida pelo *Bandwidth Skimming*, para  $k = 5$ . À medida em que  $N$  cresce, todos os protocolos experimentam compartilhamento intersítios frequente e a diferença decresce.

A seguir, são avaliados os requisitos de banda média de servidor e de rede quando a escuta silenciosa é restrita aos fluxos enviados ao mesmo sítio cliente. Esse protocolo é aqui chamado *Network-Aware SP*. Note-se que não há preocupação em otimizar a banda de rede nas uniões efetuadas durante os períodos ativos, já que restringir tais uniões tem pouco impacto sobre os requisitos de banda de rede [84]. A figura 6.3(a) mostra resultados para *Network-Aware SP-Aggressive*, para a mesma carga IM e  $k = 5$ . Em comparação com o *SP-Aggressive*, que não otimiza o uso de rede, o protocolo cuidadoso com rede provê uma economia limitada de banda de rede (até 4%), ao custo de economia muito modesta de banda de servidor (4%, não mostrada

no gráfico). Assim, para cargas com baixa interatividade, o cuidado com a rede provê ganhos apenas marginais com relação ao *Bandwidth Skimming*. Já para cargas altamente interativas, a economia de banda de rede chega a 34%, e a economia de banda de servidor é ainda significativa (até 28%).

As figuras 6.3(b) e 6.3(c) permitem avaliar o impacto significativo da redução do *buffer* local. Com um *buffer* de dimensão igual a 50% da duração da mídia, a economia de banda de rede decai significativamente. Com um *buffer* de dimensão igual a 25% da duração da mídia, a economia obtida é pouco significativa.

## 6.3 Banda de rede para estratégias híbridas em topologia de rede real

### 6.3.1 Topologia da rede real

A topologia de rede usada foi levantada por meio de sondagens *traceroute* [46], acionadas a partir de 53 servidores na Internet que oferecem acesso público por meio de uma interface *web*. De abril a setembro de 2005 foram feitas seis coletas diárias das rotas entre esses 53 servidores. Removidos erros e rotas incompletas, restaram pelo menos 300 rotas entre cada par de servidores, num total de 4.915 nodos.

### 6.3.2 Banda de rede

Para a avaliação do impacto das estratégias híbridas sobre banda de rede, escolheu-se, arbitrariamente, um dentre os 53 nodos de sondagem para sediar o servidor. Esta escolha é possível porque os clientes são uniformemente distribuídos. Os clientes foram distribuídos uniformemente por entre os demais nodos podendo existir mais de um cliente em cada um deles, e porque a distância de qualquer nodo de sondagem aos outros nodos tem distribuição normal. Os fluxos sempre percorrem o menor caminho do servidor ao cliente. Similarmente ao que é feito na seção 6.2, a banda de rede é calculada como o somatório da duração dos fluxos em todos os enlaces que conectam os nodos. A banda é afetada pelo número de fluxos de servidor que atravessam a rede e pelo número de fluxos de rede criados para compartilhar os fluxos de servidor entre múltiplos nodos. Não são necessários fluxos de rede para compartilhar fluxos entre clientes de um mesmo nodo. Esse cenário representa o pior caso de distribuição de clientes, já que, quanto mais tendenciosa é a distribuição, menos fluxos de rede são necessários para atendê-los.



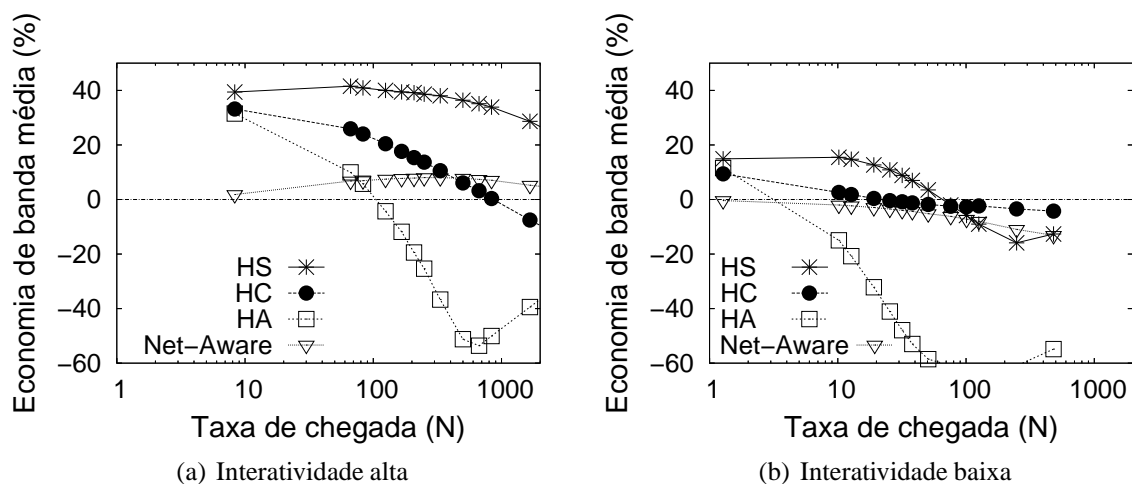


Figura 6.4: Economia de banda de rede para nível de interatividade alto

A avaliação foi focada nas estratégias híbridas, que permitem maiores economias de banda e em faixas mais amplas que as otimizações individuais. Como essas estratégias não se ocupam da economia de banda de rede, avaliou-se, como referência, a estratégia *Network-Aware*, similar à estratégia SP, mas que, em período inativo, somente escuta fluxos já disponíveis para outros clientes em seu nodo de rede (ver seção 6.2).

A figura 6.4 apresenta a economia de banda média de rede para as estratégias HS, HA, HC e *Net-Aware* para uma carga IA e uma IB típicas. Para cargas IA (figura 6.4(a)) a economia de banda de rede para HS chega a 40% sobre a banda de rede para o *Bandwidth Skimming* original. Para cargas IB (figura 6.4(b)), os ganhos são menos expressivos. A banda de rede gasta ainda é menor do que para o *Bandwidth Skimming*, porque a grande economia de banda do servidor resulta em menos fluxos sendo transmitidos através da rede. Para cargas altamente interativas, a economia de banda do servidor sobrepõe o aumento de banda causado pela escuta de fluxos em períodos inativos.

HA é pouco efetiva em termos de banda de rede (figura 6.4), porque incorpora a otimização gulosa GP, muito eficaz no uso dos períodos inativos para a escuta de fluxos (figura 5.6), resultando em menor economia de banda de rede. HC é menos agressiva que HA na escuta em períodos inativos e é mais efetiva em termos de banda de rede. HS, por incorporar KMB, fragmenta fortemente a área de armazenamento, o que reduz as oportunidades para escuta de fluxos em período inativo e leva a uma maior economia de banda de rede. *Net-Aware* não obtém economia significativa, já que, ao limitar a escuta de fluxos em período inativo, faz com que menos conteúdo fique disponível para armazenamento; a economia de banda de servidor não excede 5%, e a de rede não ultrapassa 8%.

**Conclusão:** Este capítulo identifica os principais compromissos envolvidos no impacto das estratégias desenvolvidas no capítulo anterior sobre a banda de rede e a banda de cliente. Um resultado surpreendente é que a redução de banda de servidor obtida é tão expressiva, que, especialmente para cargas de interatividade alta, ocorre redução de banda de rede, mesmo com o uso de estratégias de período inativo.

## ***7 Estratégia adaptativa para transmissão de mídia contínua com cargas interativas***

O capítulo 5 explora técnicas e estratégias voltadas à redução do impacto da interatividade na escalabilidade de protocolos de compartilhamento de banda. Inicia apresentando e avaliando técnicas que se servem de características dos protocolos e do comportamento do usuário. Na seqüência, explora estratégias que combinam técnicas individuais, como forma de, por um lado, maximizar a economia de banda obtida e, por outro lado, aumentar a faixa de dispersão na qual se consegue obter economia de banda.

Contudo, tanto as técnicas e as estratégias apresentadas neste trabalho quanto quaisquer outras que façam uso de área de armazenamento local ao cliente para reduzir retransmissão incorrem (em maior ou menor grau) no efeito degenerativo da fragmentação [67] (seção 5.3). Esse efeito pode limitar seriamente a economia obtida por uma técnica ou estratégia e, até mesmo, resultar em requisitos de banda de servidor que superem os do protocolo original. Esse comportamento diferenciado das várias técnicas e estratégias motiva o projeto de um protocolo adaptativo que monitore a carga e decida qual estratégia aplicar, ou mesmo se nenhuma estratégia deve ser aplicada, empregando-se apenas o protocolo *Bandwidth Skimming* original.

### **7.1 Protocolo adaptativo**

Esta seção apresenta um protocolo adaptativo que monitora a carga e ajusta-se às variações desta. Essa é a etapa final do trabalho de exploração do espaço de projeto de técnicas e estratégias voltadas para a interatividade efetuado nesta tese. A idéia fundamental do protocolo adaptativo é escolher, dentre as técnicas e as estratégias vistas no capítulo 5, a que melhor se adapte a cada carga ou, caso se esteja incorrendo no efeito degenerativo da fragmentação, desligar todas as técnicas e as estratégias e usar apenas o protocolo *Bandwidth Skimming* original.

É importante notar que, enquanto as técnicas e as estratégias até aqui apresentadas buscam explorar características das cargas e do protocolo base a fim de obter economia de banda de servidor, o protocolo adaptativo caminha em outra direção:

- em primeiro lugar, ele combina estratégias individuais aplicáveis a diferentes faixas de dispersão, como forma de ampliar significativamente a faixa de dispersões e os tipos de cargas em que se consegue obter economia de banda de servidor;
- em segundo lugar, ele avalia a banda de servidor e compara-a com uma estimativa de banda requerida pelo *Bandwidth Skimming*, como forma de evitar o efeito degenerativo da fragmentação de requisições, o qual limita seriamente a aplicabilidade das estratégias baseadas no uso de área de armazenamento local ao cliente.

Assim sendo, o protocolo adaptativo consegue economia significativa de banda média de servidor para uma faixa mais ampla de dispersões, e em especial para as dispersões mais baixas, nas quais o efeito degenerativo da fragmentação se manifesta fortemente, às custas de obter economia ligeiramente inferior à das melhores estratégias e técnicas individuais.

O projeto de um protocolo adaptativo poderia se tornar uma atividade muito complexa, dado o número de estratégias individuais (seis) e híbridas (três) dentre as quais deve escolher dinamicamente. No entanto, o trabalho efetuado no projeto de estratégias híbridas (seção 5.5) simplifica enormemente essa tarefa, uma vez que o projeto e a avaliação dessas estratégias já mostram quais as combinações de técnicas individuais obtêm maior economia de banda nas diferentes faixas de dispersão. Conforme visto na seção 5.5, HA (que aplica as técnicas LOC, KMB e GP) é a mais indicada quando o coeficiente de dispersão é alto ou médio, enquanto HC (que aplica as técnicas LOC e SP) é a mais indicada quando o coeficiente de dispersão se torna baixo. Além disso, dentro dessas condições de dispersão, essas estratégias são envelopes para as técnicas individuais que empregam. E, por fim, deve-se notar que HA e HC compartilham a técnica LOC (pouco sujeita ao efeito degenerativo da fragmentação), e diferem no uso de técnicas agressivas (KMB e GP para HA) e não-agressivas (SP para HC) no preenchimento da área local de armazenamento.

Dentro desse contexto, o protocolo adaptativo proposto adota o seguinte procedimento, que é descrito em mais detalhe no algoritmo 8:

1. escolhe-se um nível de coeficiente de dispersão como limiar de troca de estratégias (a escolha do limiar é mostrada na seção 7.2);

**Algoritmo 8** Protocolo Adaptativo

---

```

1:  $BW$ : Banda de servidor;
2:  $\Delta_{BS}$ : Coeficiente de dispersão na escalabilidade do Bandwidth Skimming;
3:  $C$ : Número de clientes ativos;
4:  $T_{HC}$ : Limiar de coeficiente de dispersão para o qual  $HC$  passa a ser usada;
5:  $T_{OFF}$ : Indica que nenhuma estratégia está em uso. Inicia-se com FALSE;
6:  $Proto$ : A estratégia em uso;
7:  $P_{BS}$ : A banda de servidor estimada para o Bandwidth Skimming;
8: if  $T_{OFF}$  then
9:    $Proto = CT$ ;
10: else
11:   if  $\Delta_{BS} < T_{HC}$  then
12:      $Proto = HC$ ;
13:     if  $BW < P_{BS}(\Delta_{BS}, C)$  then
14:        $Proto = CT$ ;
15:        $T_{OFF} = TRUE$ ;
16:     end if
17:   else
18:      $Proto = HA$ ;
19:     if  $BW < P_{BS}(\Delta_{BS}, C)$  then
20:        $Proto = CT$ ;
21:     end if
22:   end if
23: end if

```

---

2. o coeficiente de dispersão da carga é medido continuamente e usado como fundamento para decidir qual estratégia deve ser empregada (já que diferentes estratégias oferecem melhor economia em diferentes faixas de dispersão). Se estiver acima do limiar escolhido, adotar-se-á a estratégia HA, indicada para dispersão mais alta;
3. se o coeficiente de dispersão atingir o limiar escolhido ou ficar abaixo dele, adotar-se-á a estratégia HC, indicada para coeficiente baixo de dispersão;
4. em qualquer situação, usam-se o coeficiente de dispersão e o número de clientes ativos para estimar a banda média de servidor requerida pelo *Bandwidth Skimming* para aquela carga (seção 4.3). Se a banda estimada estiver abaixo da banda corrente, as estratégias HA e HC não deverão ser usadas, e passar-se-á a usar apenas o protocolo original.

O algoritmo acima descrito é uma prova de conceito, que evidencia o ganho obtido com uma técnica adaptativa (ver seção 7.2). A elaboração de um protocolo adaptativo completo é uma tarefa complexa, que exige tratar dados coletados pelos diversos clientes, a fim de calcular o coeficiente de dispersão de uma carga. Ainda que o servidor possa fazer esse cálculo, a coleta de dados será feita, necessariamente, pelos clientes, uma vez que somente estes têm acesso às

requisições efetuadas pelos usuários. Além disso, o cálculo do coeficiente de dispersão deverá ser efetuado periodicamente, dentro de uma certa janela de tempo. O tempo de envio de dados dos clientes ao servidor e do servidor de volta aos clientes também deverá ser considerado. A abordagem a todas essas questões é deixada como trabalho futuro.

## 7.2 Resultados

Um aspecto fundamental do protocolo adaptativo é o coeficiente de dispersão escolhido como limiar para troca de estratégia de redução de impacto da interatividade a usar. Se o protocolo for particularmente sensível a pequenas variações nesse limiar, sua aplicabilidade torna-se reduzida. Para avaliar a sensibilidade do protocolo a essas variações, sua simulação foi efetuada com nove valores diferentes para o limiar de coeficiente de dispersão (0,1, 0,2, ..., 0,9). Para cada limiar, o erro foi calculado como sendo a diferença percentual entre a banda requerida pelo protocolo adaptativo e a banda requerida pelo protocolo, técnica ou estratégia híbrida que mais economize banda para a carga em questão. O erro representa, portanto, o quanto o protocolo adaptativo se distancia do melhor resultado possível, em termos percentuais. A figura 7.1 apresenta a distribuição dos erros cometidos pelo protocolo adaptativo, considerando todas as cargas sintéticas empregadas, para quatro diferentes valores do limiar de coeficiente de dispersão (0,1, 0,2, 0,3 e 0,5). Cada uma das curvas indica os resultados para um dos limiares. Considerando todas as cargas usadas, e limiares entre 0,1 e 0,5, o erro do protocolo adaptativo está abaixo de 5% para 96% das cargas e não excede 14% para nenhuma delas. Nessa figura também se pode observar que a distribuição dos erros é similar para os limiares de coeficiente de dispersão avaliados. Dessa forma, a escolha do limiar de coeficiente de dispersão (numa faixa ampla de valores) não afeta significativamente a escalabilidade do protocolo adaptativo nem a economia de banda por ele obtida.

A economia de banda obtida pelo protocolo adaptativo é mostrada na figura 7.2, que o compara com as duas estratégias que ele emprega, HA e HC. Lembrando que, conforme mostrado no Algoritmo 8, a troca de estratégias é feita de acordo com o coeficiente de dispersão corrente da carga, a figura apresenta economia de banda do protocolo adaptativo, da estratégia HA e da HC para duas cargas de vídeo educacional, para uma de vídeo de entretenimento e para uma de áudio (música). Em todos os quatro gráficos, o limiar usado foi o de 0,30. Na figura 7.2(a), o limiar escolhido força uma troca da estratégia HA para a estratégia HC um pouco antes do ponto ideal, que seria próximo ao coeficiente de dispersão 0,20, o que limita ligeiramente a economia de banda obtida pelo protocolo adaptativo. Em contrapartida, o protocolo adaptativo evita o efeito degenerativo da fragmentação de requisições e consegue obter economia de

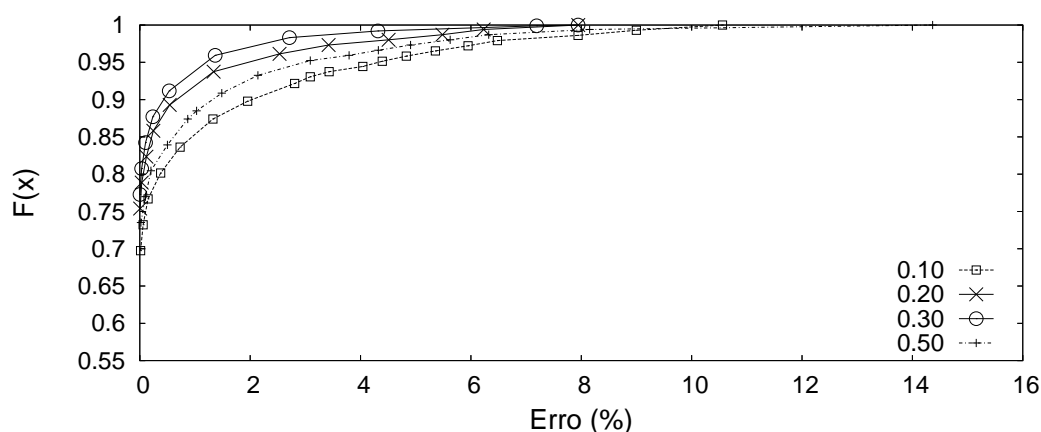


Figura 7.1: Erro do protocolo adaptativo para diferentes dispersões de referência

banda, mesmo para dispersão muito baixa. Na figura 7.2(b), a escolha do limiar recai sobre um ponto próximo ao ideal, e a economia obtida pelo protocolo adaptativo é próxima à ideal, ou seja, é próxima à que se obteria se o protocolo adaptativo alternasse de HA para HC exatamente no ponto em que a economia de banda obtida pelas duas estratégias se iguala. Na figura 7.2(c), tem-se um cenário onde a troca de HA para HC é tardia, ocorrendo após o ponto em que HA passa a apresentar penalidade em termos de banda média de servidor. Nessa situação, a estimativa de banda média de servidor requerida pelo protocolo *Bandwidth Skimming* original, feita na linha 19 do algoritmo 8, desconecta a estratégia HA e impede que o protocolo adaptativo incorra em penalidade de banda. Na figura 7.2(d), tem-se uma carga de áudio com poucas oportunidades para compartilhamento de banda, e, para dispersão mais baixa, tanto HA quanto HC incorrem em penalidades de banda. Nessa figura, a linha 13 e a 19 do algoritmo 8 impedem, então, que esse protocolo também incorra nessa penalidade. A estimativa de banda para o *Bandwidth Skimming* é otimista, e as estratégias HA e HC são desativadas para cargas em que alguma economia de banda seria possível (coeficiente de dispersão abaixo de 0,31). Em contrapartida, o efeito degenerativo da fragmentação de requisições é suprimido.

**Conclusão:** A estratégia adaptativa, apresentada neste capítulo, é uma prova de conceito de que é possível criar uma estratégia aplicável a um vasto número de cargas e taxas de chegadas (e, portando, a diversos coeficientes de dispersão). Várias questões precisam ser avaliadas para que, a partir do algoritmo aqui proposto, se chegue a um algoritmo completo. A abordagem a essas questões é deixada como trabalho futuro.

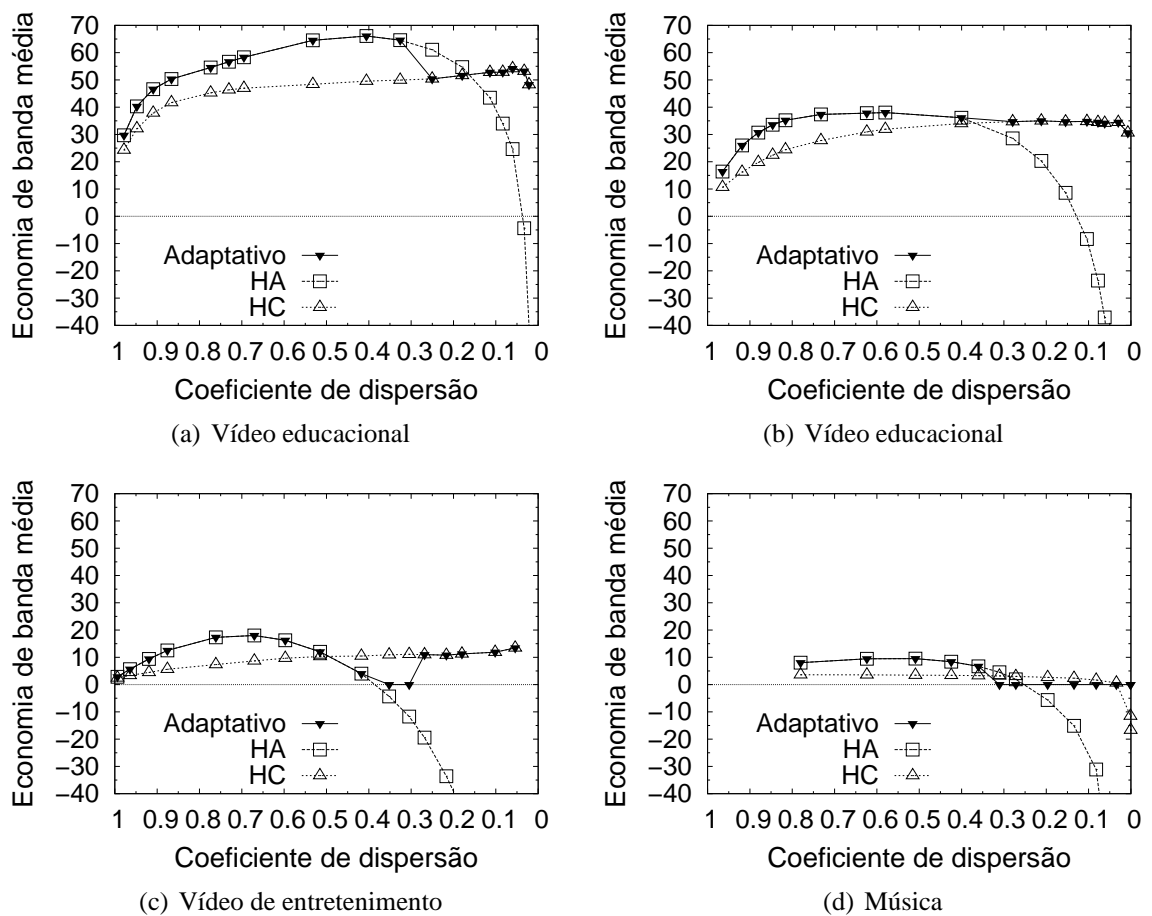


Figura 7.2: Economia de banda para o protocolo adaptativo (limiar de dispersão 0.30)



## 8 *Conclusões e trabalhos futuros*

Este trabalho investiga o impacto da interatividade na escalabilidade de protocolos de compartilhamento de banda para mídia contínua e propõe novas estratégias para melhorar essa escalabilidade. Para essa investigação, foi criado um modelo probabilístico para representar cargas de mídia contínua interativa. Esse modelo, associado a uma variedade de cargas interativas reais, foi usado para criar um rico conjunto de cargas sintéticas realistas. Esse conjunto de cargas permitiu efetivar uma exploração do espaço de projeto de protocolos de mídia contínua significativamente mais ampla que a realizada por trabalhos anteriores.

A dispersão foi identificada como fator de impacto primário das cargas interativas na escalabilidade de protocolos de mídia contínua. A dispersão foi usada para agrupar as cargas aqui empregadas em três perfis de interatividade: IA (interatividade alta), IM (interatividade média) e IB (interatividade baixa). Foi também usada como um arcabouço que facilitou significativamente a análise de escalabilidade de protocolos.

A escalabilidade do *Bandwidth Skimming* para cargas interativas foi comparada com resultado analítico prévio [74]. Foi possível confirmar, para as cargas estudadas, que esse resultado é adequado para cargas IB, verificando-se entretanto que, para cargas IA e IM, a expressão 4.1 permite calcular a escalabilidade com maior precisão. Ainda com relação às cargas IB, pôde-se determinar um valor de uma constante que permite aplicar o resultado prévio [74] a estudos de dimensionamento de capacidade de servidores.

A escalabilidade do *Bandwidth Skimming* e a do *Patching* foram comparadas com cargas interativas. Com isso, selecionou-se o *Bandwidth Skimming* como protocolo base para as estratégias aqui propostas. Os resultados obtidos mostram que as estratégias propostas aprimoram significativamente a escalabilidade do *Bandwidth Skimming*, o mais efetivo dos dois protocolos correntes analisados, para uma ampla variedade de cargas interativas e taxas de chegadas. A melhor estratégia proposta reduz a banda média de servidor requerida pelo *Bandwidth Skimming* em até 66%, com espaço em *buffer* de cliente de dimensão igual à duração da mídia. Uma segunda estratégia oferece economia mais limitada, de até 53%, com espaço ilimitado em *buf-*

fer de cliente, e de até 28% com o espaço limitado a 25% da duração da mídia. Além disso, as melhorias mais significativas obtidas pelas estratégias propostas ocorrem para cargas altamente interativas, para as quais os protocolos atuais têm sua pior escalabilidade. Foram mostrados resultados preliminares sobre os compromissos relativos à banda de cliente e à banda de rede, os quais permitem identificar os principais compromissos envolvidos na escalabilidade dos protocolos e estratégias avaliados.

O impacto da dispersão na escalabilidade de protocolos foi avaliado quantitativamente. Foram obtidas expressões para o impacto na escalabilidade do *Patching* e na do *Bandwidth Skimming*. A escalabilidade do *Bandwidth Skimming* foi avaliada, por meio de simulações, como uma função do impacto da dispersão, sendo esse um resultado relevante que permite estimar a banda média requerida pelo protocolo a partir de dados que podem ser obtidos diretamente da carga.

Nesta tese, identificou-se que qualquer protocolo de compartilhamento de banda que faça uso de *buffer* local ao cliente para reduzir a retransmissão de conteúdo está sujeito, em maior ou menor grau, ao efeito degenerativo da fragmentação de requisições. Este efeito, resultante da fragmentação do conteúdo do *buffer*, pode suprimir a economia de banda obtida ou mesmo levar a um requisito de banda superior ao do protocolo original. Esse efeito degenerativo pode limitar seriamente a aplicabilidade de qualquer estratégia de otimização a qualquer protocolo de compartilhamento de banda. A fim de permitir a aplicação das estratégias propostas a uma ampla variedade de cargas, contornando o efeito da fragmentação, foi apresentado um protocolo adaptativo que monitora a carga, mede o impacto da dispersão e decide qual estratégia aplicar ou mesmo se nenhuma deve ser aplicada. Esse protocolo foi usado com sucesso nos diversos perfis de interatividade presentes nas cargas empregadas, incluindo as de vídeo educacional de interatividade alta, as de vídeos de entretenimento de interatividade média e as de áudio de interatividade baixa, fato que o diferencia de todas as outras estratégias apresentadas.

Diversos pontos relevantes são evidenciados por esta tese e abrem espaço para uma série de trabalhos futuros. Esta tese faz a avaliação da escalabilidade com base na banda média requerida. Uma abordagem alternativa, deixada como trabalho futuro, é a avaliação com base na distribuição de banda. Também é deixada como trabalho futuro a avaliação, para cargas interativas, do dimensionamento de servidores com o uso de requisitos de banda média, como o já efetuado em [75] para cargas seqüenciais. Outro ponto a ser explorado é a comparação, para cargas interativas, dos protocolos de compartilhamento de banda com os protocolos de difusão periódica.

O protocolo adaptativo mostrou-se bem sucedido em operar com os variados perfis de inte-

ratividade e, ainda assim, contornar o efeito degenerativo da fragmentação. Contudo, ele exige que os diversos clientes cooperem para calcular a dispersão da carga. Além disso, o cálculo da dispersão deve ser efetuado em um certo período de tempo, a fim de que possa ser usado para estimar a banda requerida. O desenvolvimento desses pontos poderá ser alvo de trabalhos futuros.

A fragmentação, que foi tratada aqui com o uso do protocolo adaptativo, sugere que diferentes políticas de obtenção de conteúdo para colocação no *buffer* e também políticas para uso do conteúdo armazenado podem ter impacto sobre a escalabilidade. Trabalhos futuros poderão abordar esses aspectos.

A dispersão é um conceito chave para esta tese. Novos estudos poderão ser desenvolvidos com relação a ela entre os quais se incluem a elaboração de métricas para outros protocolos e a avaliação da escalabilidade do *Patching* em função da dispersão (nos moldes do que foi feito para o *Bandwidth Skimming*). Também poderá ser desenvolvida uma abordagem analítica para a medição da dispersão, o que ampliaria a aplicabilidade desse conceito.

## *Referências Bibliográficas*

- [1] E. L. Abram-Profeta and K. G. Shin. Providing unrestricted vcr functions in multicast video-on-demand servers. In *ICMCS '98: Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, page 66, Washington, DC, USA, 1998. IEEE Computer Society.
- [2] S. Acharya, B. Smith, and P. Parnes. Characterizing User Access to Videos on the World Wide Web. In *Proc. MMCN*, San Jose, CA, Janeiro 2000.
- [3] C. Aggarwal, J. Wolf, and P. Yu. On Optimal Piggyback Merging Policies for Video-on-Demand Systems. In *Proc. ACM SIGMETRICS*, pages 200–209, Orlando, FL, Maio 1996.
- [4] A. Allen. *Probability, Statistics, and Queueing Theory with Computer Science Applications*. Academic Press Limited, 1990.
- [5] J. Almeida, J. Krueger, D. Eager, and M. Vernon. Analysis of Educational Media Server Workloads. In *Proc. NOSSDAV*, Port Jefferson, NY, Junho 2001.
- [6] K. C. Almeroth and M. H. Ammar. On the performance of a multicast delivery video-on-demand service with discontinuous VCR actions. In *IEEE International Conference on Communications*, volume 3, pages 1631–1635, 1995.
- [7] B. C. B. Alves, R. M. M. Leão, and E. de S. e Silva. Caracterizando Variáveis de Interatividade dos Alunos do Curso de Computação do CEDERJ, Baseado no Servidor Multimídia RIO. In *Proc. XXVII SBC 2007, Anais do XXVII Congresso da Sociedade Brasileira de Computação*, Rio de Janeiro, Brasil, Julho 2007.
- [8] S. Banerjee, B. Bhattacharjee, and C. Kommareddy. Scalable application layer multicast. In *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 205–217, New York, NY, USA, 2002. ACM.
- [9] S. Banerjee, S. Lee, R. Braud, B. Bhattacharjee, and A. Srinivasan. Scalable resilient media streaming. In *NOSSDAV '04: Proceedings of the 14th international workshop on Network and operating systems support for digital audio and video*, pages 4–9, New York, NY, USA, 2004. ACM.
- [10] R.O. Banker, J.B. Huppertz, M.T. Hayashi, D.B. Lett, V.E. Godlewski, and M.W. Raley. Method of providing video on demand with VCR like functions, October 18 1994. US Patent 5,357,276.
- [11] A. Bar-Noy and R. Ladner. Competitive On-Line Stream Merging Algorithms for Media-on-Demand. *Journal of Algorithms*, 48(1):59–90, 2003.

- [12] M. Bradshaw, B. Wang, S. Sen, L. Gao, J. Kurose, P. Shenoy, and D. Towsley. Periodic Broadcast and Patching Services - Implementation, Measurement and Analysis in an Internet Streaming Video Testbed. *Multimedia Systems*, 9(1):78–93, 2003.
- [13] L. Filho C. Rodrigues and R. Leão. Buffering para Otimização de Sistemas Interativos de Vod. In *Proc. Simpósio Brasileiro de Redes de Computadores*, pages 941–954, Belém, PA, Maio 2007.
- [14] Y. Cai, K. Hua, and K. Vu. Optimizing Patching Performance. In *Proc. Conf. on Multimedia Computing and Networking*, San Jose, CA, Janeiro 1999.
- [15] Y. Cai and K. A. Hua. Sharing multicast videos using patching streams. *Multimedia Tools Appl.*, 21(2):125–146, 2003.
- [16] S. W. Carter and D. D. E. Long. Video-on-demand server efficiency through stream tapping. In *IC3N '97: Proceedings of the 6th International Conference on Computer Communications and Networks*, page 200, Washington, DC, USA, 1997. IEEE Computer Society.
- [17] S. W. Carter, D. D. E. Long, and J. Pâris. An efficient implementation of interactive video-on-demand. In *MASCOTS '00: Proceedings of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, page 172, Washington, DC, USA, 2000. IEEE Computer Society.
- [18] H.J. Chen, A. Krishnamurthy, T.D.C. Little, and D. Venkatesh. A scalable video-on-demand service for the provision of vcr-like functions. In *ICMCS '95: Proceedings of the International Conference on Multimedia Computing and Systems*, page 65, Washington, DC, USA, 1995. IEEE Computer Society.
- [19] L. Cherkasova and M. Gupta. Characterizing locality, evolution, and life span of accesses in enterprise media server workloads. In *NOSSDAV '02: Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video*, pages 33–42, New York, NY, USA, 2002. ACM Press.
- [20] L. Cherkasova and M. Gupta. Analysis of enterprise media server workloads: access patterns, locality, content evolution, and rates of change. *IEEE/ACM Trans. Netw.*, 12(5):781–794, 2004.
- [21] M. Chesire, A. Wolman, G. Voelker, and H. Levy. Measurement and Analysis of a Streaming Media Workload. In *Proc. Symp. on Internet Technologies and Systems*, San Francisco, CA, Março 2001.
- [22] Y. Chu, S. Rao, S. Seshan, and H. Zhang. Enabling conferencing applications on the internet using an overlay multicast architecture. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 55–67, New York, NY, USA, 2001. ACM.
- [23] E. Coffman, P. Momcilovic, and P. Jelenkovic. The Dyadic Stream Merging Algorithm. *Journal of Algorithms*, 43(1):120–137, 2002.
- [24] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2001.

- [25] C. Costa, I. Cunha, A. Borges, C. Ramos, M. Rocha, J. Almeida, and B. Ribeiro-Neto. Analyzing Client Interactive Behavior on Streaming Media Servers. In *Proc. 13th WWW Conf.*, New York, Maio 2004.
- [26] C. Costa, C. Ramos, I. Cunha, and JM Almeida. GENIUS: A Generator of Interactive User Media Sessions. *Workload Characterization, 2004. WWC-7. 2004 IEEE International Workshop on*, pages 29–36, 2004.
- [27] A. Dan, P. Shahabuddin, D. Sitaram, and D. Towsley. Channel allocation under batching and vcr control in video-on-demand systems. *J. Parallel Distrib. Comput.*, 30(2):168–179, 1995.
- [28] A. Dan, D. Sitaram, and P. Shahabuddin. Scheduling Policies for an On-Demand Video Server with Batching. In *MULTIMEDIA '94: Proceedings of the second ACM international conference on Multimedia*, pages 15–23, New York, NY, USA, 1994. ACM Press.
- [29] J. K. Dey-Sircar, J. D. Salehi, J. F. Kurose, and D. Towsley. Providing vcr capabilities in large-scale video servers. In *MULTIMEDIA '94: Proceedings of the second ACM international conference on Multimedia*, pages 25–32, New York, NY, USA, 1994. ACM Press.
- [30] D. Eager and M. Vernon. Dynamic Skyscraper Broadcasts for Video-on-Demand. In *Int'l. Conf. Workshop on Advances in Multimedia Information Systems*, Istanbul, Turkey, Setembro 1998.
- [31] D. Eager, M. Vernon, and J. Zahorjan. Optimal and Efficient Merging Schedules for Video-on-Demand Servers. In *Proc. ACM MULTIMEDIA*, Orlando, FL, Novembro 1999.
- [32] D. Eager, M. Vernon, and J. Zahorjan. Bandwidth Skimming: A Technique for Cost-Effective Video on Demand. In *Proc. MMCN*, San Jose, CA, Janeiro 2000.
- [33] D. Eager, M. Vernon, and J. Zahorjan. Minimizing Bandwidth Requirements for On-Demand Data Delivery. *IEEE Trans. on Knowledge and Data Engineering*, 13(5):742–757, 2001.
- [34] eTeach. <http://eteach.cs.wisc.edu/index.html>.
- [35] S. Fahmy and M. Kwon. Characterizing overlay multicast networks and their costs. *IEEE/ACM Trans. Netw.*, 15(2):373–386, 2007.
- [36] Z. Fei, M. H. Ammar, I. Kamel, and S. Mukherjee. Providing interactive functions through active client-buffer management in partitioned video multicast vod systems. In *NGC '99: Proceedings of the First International COST264 Workshop on Networked Group Communication*, pages 152–169, London, UK, 1999. Springer-Verlag.
- [37] W. Feng, F. Jahanian, and S. Sechrest. Providing vcr functionality in a constant quality video-on-demand transportation service. In *ICMCS '96: Proceedings of the 1996 International Conference on Multimedia Computing and Systems (ICMCS '96)*, page 0127, Washington, DC, USA, 1996. IEEE Computer Society.
- [38] L. Gao and D. Towsley. Supplying Instantaneous Video-on-Demand Services Using Controlled Multicast. In *Proc. IEEE Multimedia Computing Systems*, Junho 1999.

- [39] L. Golubchik, J. C. S. Lui, and R. Muntz. Reducing i/o demand in video-on-demand storage servers. In *SIGMETRICS '95/PERFORMANCE '95: Proceedings of the 1995 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, pages 25–36, New York, NY, USA, 1995. ACM Press.
- [40] L. Guo, S. Chen, Z. Xiao, and X. Zhang. Analysis of multimedia workloads with implications for internet streaming. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 519–528, New York, NY, USA, 2005. ACM Press.
- [41] N. Harel, V. Vellanki, A. Chervenak, G. Abowd, and U. Ramachandran. Workload of a media-enhanced classroom server. *Proceedings of Workshop on Workload Characterization*, 1999.
- [42] A. Hu. Video-on-Demand broadcasting protocols: A comprehensive study. In *Infocom*, 2001.
- [43] K. Hua, Y. Cai, and S. Sheu. Patching: A Multicast Technique for True Video-on-Demand Services. In *Proc. ACM MULTIMEDIA*, Bristol, UK, Setembro 1998.
- [44] K. Hua and S. Sheu. An Efficient Periodic Broadcast Technique for Digital Video Libraries. *Multimedia Tools and Applications*, 10(2/3):157–177, 2000.
- [45] K. A. Hua and D. A. Tran. Range multicast for video on demand. *Multimedia Tools Appl.*, 27(3):367–391, 2005.
- [46] V. Jacobson. Traceroute tool. <ftp://ftp.ee.lbl.gov/traceroute.tar.gz>, 2005.
- [47] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and Jr. J. W. O'Toole. Overcast: reliable multicasting with on overlay network. In *OSDI'00: Proceedings of the 4th conference on Symposium on Operating System Design & Implementation*, pages 14–14, Berkeley, CA, USA, 2000. USENIX Association.
- [48] S. Jin and A. Bestavros. Gismo: a generator of internet streaming media objects and workloads. *SIGMETRICS Perform. Eval. Rev.*, 29(3):2–10, 2001.
- [49] S. Jin and A. Bestavros. Scalability of Multicast Delivery for Non-sequential Streaming Access. In *Proc. SIGMETRICS*, Marina Del Rey, Junho 2002.
- [50] S.W. Lau, J.C.S. Lui, and L. Golubchik. Merging video streams in a multimedia storage server: complexity and heuristics. *Multimedia Systems*, 6(1):29–42, 1998.
- [51] A. Laursen, J. Olkin, and M. Porter. Oracle media server: providing consumer based interactive access to multimedia data. In *SIGMOD '94: Proceedings of the 1994 ACM SIGMOD international conference on Management of data*, pages 470–477, New York, NY, USA, 1994. ACM Press.
- [52] K. K. W. Law, J. C. S. Lui, and Leana Golubchik. Efficient support for interactive I/ in multi-resolution vod systems. *The VLDB Journal*, 8(2):133–153, 1999.
- [53] Y. Leung and T. K. C. Chan. Design of an interactive video-on-demand system. *IEEE Trans. on Multimedia*, 5(1):130–140, March 2003.

- [54] W. Liao and V. O. K. Li. The split and merge protocol for interactive video-on-demand. *IEEE MultiMedia*, 4(4):51–62, 1997.
- [55] H. Ma, K. Shin, and W. Wu. Best-effort Patching for Multicast True VoD Service. *Multimedia Tools Applications*, 26(1):101–122, 2005.
- [56] H. Ma and K. G. Shin. Multicast video-on-demand services. *SIGCOMM Comput. Commun. Rev.*, 32(1):31–43, 2002.
- [57] M. Maia, M. Rocha, Í. Cunha, J. Almeida, and S. Campos. Network bandwidth requirements for optimized streaming media transmission to interactive users. In *WebMedia '06: Proceedings of the 12th Brazilian symposium on Multimedia and the web*, pages 283–292, New York, NY, USA, 2006. ACM Press.
- [58] L. R. Mendes, Y. Iano, V. I. B. Sablón, and R. T. Pietro. Subsistema de Compressão e Codificação do Sinal de Vídeo dos Padrões HDTV (Parte II). *Telecomunicações - Revista Científica Periódica*, 03(01):19–27, 2000.
- [59] B. Netto. Patching Interativo: Um Novo Método de Compartilhamento de Recursos para Transmissão de Vídeo com Alta Interatividade. Master's thesis, UFRJ, Brasil, 2004.
- [60] B. Netto, J. A. Azevedo, E. de Souza e Silva, and R. M. M. Leão. Servidor Multimídia RIO em Ensino à Distância. In *Proc. 6th International Free Software Forum*, volume I, pages 91–95, Porto Alegre, Brasil, Junho 2005.
- [61] B. Özden, R. Rastogi, and A. Silberschatz. On the storage and retrieval of continuous media data. In *CIKM '94: Proceedings of the third international conference on Information and knowledge management*, pages 322–328, New York, NY, USA, 1994. ACM Press.
- [62] B. Özden, R. Rastogi, and A. Silberschatz. Research issues in multimedia storage servers. *ACM Comput. Surv.*, 27(4):617–620, 1995.
- [63] J. Padhye and J. Kurose. An Empirical Study of Client Interactions with a Continuous-Media Courseware Server. *IEEE Internet Computing*, 3(2):65–73, 2003.
- [64] J.F. Paris. An interactive broadcasting protocol for video-on-demand. *Performance, Computing, and Communications, 2001. IEEE International Conference on.*, pages 347–353, 2001.
- [65] RIPPLES/MANIC. <http://manic.cs.umass.edu>.
- [66] M. Rocha, M. Maia, J. Almeida, and S. Campos. Escalabilidade de Protocolos com Compartilhamento de Banda para Cargas de Mídia Contínua Realistas. In *Proc. SBC WPerformance*, São Leopoldo, RS, Brasil, Jul. 2005.
- [67] M. Rocha, M. Maia, Ítalo Cunha, J. Almeida, and S. Campos. Scalable Media Streaming to Interactive Users. In *Proc. ACM MULTIMEDIA*, Singapore, Singapore, Nov. 2005.
- [68] C. Rodrigues and R. Leão. Novas Técnicas de Compartilhamento de Banda para Servidores de Vídeo Sob Demanda com Interatividade. In *Proc. Simpósio Brasileiro de Redes de Computadores*, Fortaleza, CE, Brasil, Maio 2005.



- [69] C. Rodrigues and R. Leão. Técnicas para Sistemas de Vídeo sob Demanda Escaláveis. In *Proc. XXIV Simpósio Brasileiro de Redes de Computadores (SBRC 2006)*, pages 247–262, Curitiba, PR, Brasil, Maio 2006.
- [70] J. R. Santos, R. R. Muntz, and B. Ribeiro-Neto. Comparing random data allocation and data striping in multimedia servers. In *SIGMETRICS '00: Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 44–55, New York, NY, USA, 2000. ACM Press.
- [71] J. J. Schultz and T. Znati. An efficient scheme for chaining with client-centric buffer reservation for multi-media streaming. In *ANSS '03: Proceedings of the 36th annual symposium on Simulation*, page 31, Washington, DC, USA, 2003. IEEE Computer Society.
- [72] S. Sen, J. Rexford, and D. Towsley. Proxy Prefix Caching for Multimedia Streams. In *Proc. IEEE INFOCOM*, New York, NY, Março 1999.
- [73] S. Sheu, K.A. Hua, and W. Tavanapong. Chaining: A generalized batching technique for video-on-demand systems. In *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pages 110 – 117, June 1997.
- [74] H. Tan, D. Eager, and M. Vernon. Delimiting the Range of Effectiveness of Scalable On-Demand Streaming. In *Proc. Performance*, Rome, Setembro 2002.
- [75] H. Tan, D. Eager, M. Vernon, and H. Guo. Quality of Service Evaluations of Multicast Streaming Protocols. In *Proc. SIGMETRICS*, Marina Del Rey, Junho 2002.
- [76] W. Tang, Y. Fu, L. Cherkasova, and Amin Vahdat. Medisyn: a synthetic streaming media service workload generator. In *NOSSDAV '03: Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video*, pages 12–21, New York, NY, USA, 2003. ACM Press.
- [77] M. A. Tantaoui, K. A. Hua, and S. Sheu. Interaction with broadcast video. In *MULTIMEDIA '02: Proceedings of the tenth ACM international conference on Multimedia*, pages 29–38, New York, NY, USA, 2002. ACM Press.
- [78] D. Tomimura. Caracterização e Modelagem do Comportamento de Usuários Acessando um Vídeo de Ensino à Distância. Master's thesis, UFRJ, Brasil, 2006.
- [79] D. Tomimura, R. M. M. Leão, E. de Souza e Silva, and F. F. S. Filho. Caracterização do comportamento de usuários acessando um vídeo de ensino à distância. In *SBC2006 - V Workshop em Desempenho de Sistemas Computacionais e de Comunicação*, pages 34–53, 2006.
- [80] Universo Online. <http://www.uol.com.br>.
- [81] B. Vickers, C. Albuquerque, and T. Suda. Source-adaptive Multilayered Multicast Algorithms for Real-time Video Distribution. *IEEE/ACM Transactions on Networking*, 8(6):720–733, 2000.
- [82] N. J. Vieira. *Introdução aos Fundamentos da Computação*. Thomson Learning, 2006.

- 
- [83] H. Yu, D. Zheng, B. Y. Zhao, and W. Zheng. Understanding user behavior in large-scale video-on-demand systems. In *EuroSys '06: Proceedings of the 2006 EuroSys conference*, pages 333–344, New York, NY, USA, 2006. ACM Press.
- [84] Y. Zhao, D. Eager, and M. Vernon. Network Bandwidth Requirements for Scalable On-Demand Streaming. In *Proc. INFOCOM*, New York, Junho 2002.
- [85] G. Zipf. *Human Behavior and the Principle of Least-Effort*. Addison-Wesley, Cambridge, MA, 1949.