



COPPE/UFRJ

DISTRIBUIÇÃO DE VÍDEO SOBRE REDES PAR-A-PAR

Igor Monteiro Moraes

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica.

Orientador: Otto Carlos Muniz Bandeira
Duarte

Rio de Janeiro
Dezembro de 2009

DISTRIBUIÇÃO DE VÍDEO SOBRE REDES PAR-A-PAR

Igor Monteiro Moraes

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Otto Carlos Muniz Bandeira Duarte, Dr.Ing.

Prof. Dorgival Olavo Guedes Neto, Ph.D.

Prof. Luís Henrique Maciel Kosmalski Costa, Dr.

Prof. Marcelo Gonçalves Rubinstein, D.Sc.

Prof. Célio Vinicius Neves de Albuquerque, Ph.D.

RIO DE JANEIRO, RJ – BRASIL

DEZEMBRO DE 2009

Moraes, Igor Monteiro

Distribuição de Vídeo sobre Redes Par-a-Par/Igor Monteiro Moraes. – Rio de Janeiro: UFRJ/COPPE, 2009.

XII, 103 p.: il.; 29, 7cm.

Orientador: Otto Carlos Muniz Bandeira Duarte

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2009.

Referências Bibliográficas: p. 94 – 103.

1. Distribuição de Vídeo. 2. Sistemas Par-a-Par. 3. Avaliação Experimental. 4. Seleção de Pares. I. Duarte, Otto Carlos Muniz Bandeira. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Ao meu avô Paulo.

Agradecimentos

Agradeço aos meus pais e demais familiares pelo carinho, compreensão e incentivo ao longo de toda a minha vida. Agradeço à Luciana, uma menina muito especial que conheci apenas em 2007, mas que ficará ao meu lado por muito tempo. Muito obrigado pelo apoio e pela compreensão ao longo deste trabalho.

Ao professor Otto, meu orientador e responsável por grande parte da minha formação acadêmica e profissional, por sua amizade, conselhos e orientação. Aos professores Luís Henrique, Rubi, Rezende, Célio e Mauros, pela amizade, ensinamentos e conselhos.

Ao grande amigo e recém-professor Miguel, pelas contribuições a este trabalho e por toda nossa saudável convivência dentro e fora do laboratório.

Aos amigos desde a graduação e a iniciação científica, Bernardo, Doc, Marco, Pedro e Rafael, pelos momentos de diversão e pelos comentários sempre pertinentes sobre a realização deste trabalho.

A todos do GTA, em especial aos amigos Carlo, Carlos Henrique, Celso, Kleber, Lino, Marcel, Marcelo, Natalia, Rafael Santos e ao restante do pessoal do Rezende e da Iniciação Científica, pelas risadas e pelo incentivo durante o trabalho.

Agradeço em particular aos professores Dorgival Guedes, Luís Henrique Costa, Marcelo Rubinstein e Célio Albuquerque pela participação na banca examinadora.

Aos funcionários do Programa de Engenharia Elétrica da COPPE/UFRJ pela presteza no atendimento na secretaria do Programa.

A todos que me incentivaram, contribuindo de forma direta ou indireta, para a minha formação acadêmica e profissional.

Ao CNPq, à CAPES, à FINEP e ao FUNTTEL pelo financiamento da pesquisa.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

DISTRIBUIÇÃO DE VÍDEO SOBRE REDES PAR-A-PAR

Igor Monteiro Moraes

Dezembro/2009

Orientador: Otto Carlos Muniz Bandeira Duarte

Programa: Engenharia Elétrica

O uso de sistemas par-a-par é uma das soluções mais promissoras para a distribuição de vídeo na Internet, principalmente devido à escalabilidade desses sistemas. No entanto, a adoção de sistemas par-a-par para tal fim traz novos desafios. Um deles é avaliar experimentalmente o impacto da mobilidade dos usuários no desempenho de sistemas par-a-par de difusão de vídeo em uma rede sem fio. Mostra-se que tais sistemas se adaptam melhor ao cenário de mobilidade do que um aplicativo cliente-servidor independentemente do protocolo de transporte utilizado. A arquitetura distribuída dos sistemas par-a-par é o fator que mais contribui para tal adaptação. Outro desafio é aumentar a disponibilidade de pedaços de vídeo de interesse nos parceiros de um participante nos sistemas par-a-par de vídeo sob demanda, pois, nesses sistemas, os participantes controlam a reprodução do vídeo. Neste trabalho, é proposto um mecanismo de seleção de parceiros baseado no tempo de vida dos participantes. Um participante seleciona como seus parceiros outros participantes que tenham tempos de vida próximos ao seu e, com isso, aumenta-se a disponibilidade de pedaços de interesse entre os parceiros. O mecanismo proposto, em comparação com outros mecanismos, seleciona parceiros mais eficientes e não exige nenhuma mensagem de controle adicional, devido à interatividade dos usuários. Por isso, proporciona uma continuidade de reprodução do vídeo superior a 99% para a maioria dos padrões de interatividade analisados e, mesmo sendo o mais eficiente, exige tamanhos de *cache* menores do que os demais mecanismos.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

PEER-TO-PEER VIDEO STREAMING

Igor Monteiro Moraes

December/2009

Advisor: Otto Carlos Muniz Bandeira Duarte

Department: Electrical Engineering

Peer-to-peer systems are a promising solution for video streaming in the Internet, mainly because they are scalable. The adoption of these systems for video streaming brings new challenges. One of them is to experimentally evaluate the impact of user mobility on the performance of peer-to-peer (P2P) live-video systems over wireless networks. We show that these systems are more suitable for mobile environments than a client-server video application, regardless of the transport protocol employed by the P2P system. The P2P distributed architecture is the most important factor for this better performance. Another challenge is to increase the availability of chunks of interest between a peer and its partners in peer-to-peer video-on-demand systems because, in these systems, peers control the video playback. In this work, we propose a partner selection mechanism based on the lifetime of peers. A peer selects other peers with close lifetimes as partners in order to increase the availability of chunks of interest among partners. The proposed mechanism, in comparison with other mechanisms, selects more efficient partners and does not require additional control messages due to the users' interactivity. Thus, our mechanism provides a video playback continuity higher than 99% for the most interactivity patterns analyzed, which is the best performance, and requires smaller cache sizes than the other mechanisms.

Sumário

| | |
|---|------------|
| Lista de Figuras | x |
| Lista de Tabelas | xii |
| 1 Introdução | 1 |
| 1.1 Objetivos | 4 |
| 1.2 Organização do Texto | 6 |
| 2 Distribuição de Vídeo na Internet | 8 |
| 2.1 A Arquitetura da Internet e suas Limitações | 9 |
| 2.2 Técnicas e Propostas para Distribuição de Vídeo na Internet | 10 |
| 2.2.1 Comunicação Multidestinatória | 11 |
| 2.2.2 Codificação e Transmissão Adaptativas | 15 |
| 2.2.3 Redes de Distribuição de Conteúdo | 17 |
| 3 Distribuição de Vídeo sobre Redes Par-a-Par | 19 |
| 3.1 Fundamentos dos Sistemas Par-a-Par de Vídeo | 19 |
| 3.2 Arquiteturas de Distribuição | 21 |
| 3.2.1 Arquitetura de Distribuição em Árvore | 22 |
| 3.2.2 Arquitetura de Distribuição em Malha | 24 |
| 3.3 Sistemas de Difusão de Vídeo | 25 |
| 3.3.1 CoolStreaming/DONet | 26 |
| 3.4 Sistemas de Vídeo sob Demanda | 32 |
| 3.4.1 BulletMedia | 33 |
| 3.4.2 PPLive | 35 |

| | | |
|----------|---|-----------|
| 4 | Avaliação Experimental de Sistemas Par-a-Par de Vídeo em Redes Sem Fio | 37 |
| 4.1 | Trabalhos Relacionados | 38 |
| 4.2 | Redes em Malha Sem Fio | 39 |
| 4.2.1 | Endereçamento | 40 |
| 4.3 | Sistemas Par-a-Par de Vídeo Analisados | 42 |
| 4.4 | Resultados | 43 |
| 4.4.1 | Aplicativo Cliente-Servidor | 46 |
| 4.4.2 | Sistema Par-a-Par Baseado no TCP: PPStream | 47 |
| 4.4.3 | Sistema Par-a-Par Baseado no UDP: SopCast | 49 |
| 4.4.4 | Resumo dos Experimentos | 52 |
| 5 | Seleção de Pares em Sistemas Par-a-Par de Vídeo sob Demanda | 54 |
| 5.1 | Trabalhos Relacionados | 56 |
| 5.2 | Modelo do Sistema | 59 |
| 5.3 | O Mecanismo LIPS (<i>Lifetime-based Peer Selection</i>) | 61 |
| 5.4 | O Ambiente de Simulação | 66 |
| 5.5 | Resultados | 69 |
| 5.5.1 | Avaliação dos Mecanismos para os Diferentes Vídeos | 72 |
| 5.5.2 | Avaliação dos Mecanismos para os Diferentes Parâmetros do Sistema | 83 |
| 6 | Conclusões | 89 |
| | Referências Bibliográficas | 94 |

Lista de Figuras

| | | |
|-----|---|----|
| 2.1 | A heterogeneidade dos nós da Internet. | 11 |
| 2.2 | Um exemplo de uma rede sobreposta (<i>overlay</i>). | 13 |
| 2.3 | As diferenças entre a comunicação ponto-a-ponto e multidestinatória. | 14 |
| 2.4 | Um exemplo da codificação em camadas. | 16 |
| 2.5 | O funcionamento de uma rede de distribuição de conteúdo. | 18 |
| 3.1 | A arquitetura de distribuição em árvore. | 22 |
| 3.2 | A arquitetura de distribuição em malha. | 24 |
| 3.3 | A arquitetura de um nó participante do sistema CoolStreaming. | 27 |
| 3.4 | A entrada de um novo nó no sistema CoolStreaming. | 28 |
| 3.5 | O mapa de <i>buffer</i> no sistema CoolStreaming. | 31 |
| 4.1 | A trajetória do experimento. | 45 |
| 4.2 | Medições com o aplicativo cliente-servidor. | 47 |
| 4.3 | Medições com o PPStream. | 49 |
| 4.4 | Medições com o SopCast. | 51 |
| 5.1 | A taxa de acerto em função do tamanho do <i>cache</i> | 74 |
| 5.2 | O número de pedaços de interesse por parceiro em função do tamanho do <i>cache</i> | 76 |
| 5.3 | A sobrecarga da fonte em função do tamanho do <i>cache</i> | 80 |
| 5.4 | O índice de continuidade em função do tamanho do <i>cache</i> | 81 |
| 5.5 | O número de participantes que esperam por um pedaço de vídeo em função do tamanho do <i>cache</i> | 82 |
| 5.6 | O impacto do intervalo de atualização nos mecanismos de seleção de parceiros. | 84 |

| | | |
|------|--|----|
| 5.7 | O impacto do número máximo de parceiros nos mecanismos de seleção de parceiros. | 85 |
| 5.8 | O impacto do número máximo de parceiros da fonte nos mecanismos de seleção de parceiros. | 86 |
| 5.9 | O impacto do tamanho da janela de interesse nos mecanismos de seleção de parceiros. | 86 |
| 5.10 | O impacto dos parâmetros do anel expansivo no LIPS. | 87 |

Lista de Tabelas

| | | |
|-----|--|----|
| 4.1 | Principais características dos sistemas par-a-par analisados. | 43 |
| 4.2 | Resumo dos experimentos. | 53 |
| 5.1 | Frequência e tipo de interações dos participantes. | 68 |
| 5.2 | Distribuições que caracterizam o comportamento dos participantes. . | 69 |
| 5.3 | Parâmetros considerados em todas as simulações. | 71 |
| 5.4 | Parâmetros para a avaliação dos mecanismos para os diferentes vídeos. | 72 |
| 5.5 | Número de mensagens de controle adicionais para o LIPS e para a seleção baseada no tempo de reprodução. | 78 |

Capítulo 1

Introdução

A distribuição de vídeo é uma das aplicações de maior sucesso atualmente na Internet. Sítios que oferecem serviços de compartilhamento e distribuição de vídeos sob demanda são acessados por milhões de usuários diariamente. Somente o YouTube, o sítio mais famoso desse gênero, é acessado diariamente por cerca de 20 milhões de usuários [1, 2]. Além disso, emissoras de TV de todo o mundo já possuem serviços de difusão de TV na Internet. Nos Estados Unidos, a CBS já atingiu a marca de 260 mil usuários simultâneos durante a transmissão das finais da liga universitária de basquete em março de 2006 [3]. No Brasil, o portal Globo.com atingiu a marca de 72 mil usuários simultâneos durante a transmissão da semifinal entre Portugal e França na Copa do Mundo de 2006 [4].

As aplicações de distribuição de vídeo¹ na Internet podem ser classificadas em aplicações de vídeo sob demanda e aplicações de difusão. No vídeo sob demanda, um usuário seleciona o conteúdo que deseja receber e pode controlar a sua reprodução, como em um videocassete. O controle das ações de reprodução pode ser feito localmente pelo usuário, uma vez que o conteúdo recebido também é armazenado, ou pode ser solicitado diretamente à fonte. Para tanto, é necessário usar um protocolo como, por exemplo, o RTSP (*Real-Time Streaming Protocol*) para enviar mensagens de controle à fonte. Na difusão², uma fonte envia o mesmo conteúdo de vídeo para

¹Neste trabalho, a expressão reduzida “distribuição de vídeo” é usada para se referir à distribuição de fluxo contínuo de vídeo (*video streaming*). Na distribuição de fluxo contínuo, um usuário não precisa receber e armazenar todo o conteúdo de vídeo para reproduzi-lo. O vídeo é enviado em um fluxo contínuo e o seu conteúdo é reproduzido ao ser recebido pelo usuário.

²O termo difusão (*broadcast*) é usado como sinônimo de comunicação multidestinatária (*multi-*

um grupo de usuários, como ocorre em uma transmissão de TV. Um usuário recebe e reproduz o vídeo a partir do instante de sua requisição e não tem controle sobre as ações de reprodução.

Os maiores desafios da distribuição de vídeo na Internet são a escalabilidade, a garantia dos requisitos de qualidade de serviço das aplicações e a mobilidade dos usuários. Atualmente, as aplicações disponibilizam vídeos da ordem de centenas de kilobits por segundo e o número de usuários simultâneos é da ordem de centenas de milhares. Com isso, os recursos de banda passante exigidos de um distribuidor de vídeo são da ordem de centenas de gigabits por segundo, considerando o uso do modelo cliente-servidor e da comunicação ponto-a-ponto (*unicast*). Portanto, no modelo cliente-servidor, quanto mais usuários e quanto maior for a qualidade do vídeo, maiores são os custos dos distribuidores, o que torna esse modelo inadequado para a distribuição de vídeo na Internet. Além disso, a demanda por aplicações de distribuição de vídeo não está mais restrita apenas aos usuários de computadores pessoais. Com a popularização dos dispositivos móveis e o aumento da dependência da Internet, os usuários desejam acessar a rede a qualquer instante e em qualquer lugar, mesmo que estejam em movimento. Nesse cenário, as redes de acesso usam a transmissão sem fio, na qual as condições do meio são muito variáveis e existem restrições de banda passante. Além disso, os usuários trocam de ponto de acesso enquanto se movimentam. Essas características afetam o desempenho do modelo cliente-servidor nesse cenário, uma vez que o encerramento e o restabelecimento da conexão se tornam mais frequentes.

Diversas alternativas ao modelo cliente-servidor foram propostas. Uma das soluções mais promissoras atualmente é a distribuição de vídeo sobre redes par-a-par (*peer-to-peer* - P2P) [3, 5, 6]. Os sistemas par-a-par não exigem modificações no núcleo da rede, são escaláveis e reduzem os custos dos distribuidores de vídeo. Nesses sistemas, os próprios nós participantes³ constroem a arquitetura de distribuição do vídeo na camada de aplicação e, por isso, não exigem modificações nos roteadores da rede. Tal característica facilita a implantação desses sistemas. A escalabilidade

cast) para fazer a associação com a difusão convencional de rádio e TV. Formalmente, na difusão um mesmo conteúdo é enviado para todos os destinatários, enquanto que na comunicação multi-destinatária um mesmo conteúdo é enviado apenas para um grupo de destinatários.

³Ao longo deste trabalho os termos participantes, usuários, nós e pares são sinônimos.

e o custo reduzido para os distribuidores são garantidos pela cooperação entre os nós. Cada nó ao receber um vídeo pode encaminhar se solicitado esse mesmo vídeo para os demais nós pertencentes à arquitetura de distribuição e, conseqüentemente, compartilha seus recursos, como banda passante, espaço de armazenamento e processamento. Dessa forma, quanto mais nós participam do sistema de distribuição, maior é a capacidade total do sistema.

A difusão de vídeo ao vivo sobre sistemas par-a-par é atualmente um sucesso [3, 5, 6, 7]. Tal afirmativa pode ser comprovada pelo número de sistemas disponíveis comercialmente e propostos na literatura [8], bem como pelo número de usuários desses sistemas. Alguns sistemas possuem milhões de usuários e suportam centenas de milhares de usuários assistindo a um mesmo canal [9]. No entanto, atualmente, a maioria dos sistemas de vídeo par-a-par é proprietária, sem especificação ou código fonte aberto. Por isso, a avaliação desses sistemas na prática é um desafio, uma vez que muitas características relacionadas com a arquitetura interna do sistema não são conhecidas [10]. Além disso, existem diferentes aspectos práticos, provenientes de formas de utilização, configuração ou implementação de protocolos e aplicações, que influenciam o desempenho de sistemas par-a-par na Internet [7]. Uma forma de identificar as características desses sistemas e os aspectos práticos que os influenciam é através de avaliações experimentais. É possível determinar através de medições de tráfego, por exemplo, o protocolo de transporte usado por um sistema de vídeo par-a-par. Dependendo do protocolo usado, um sistema pode ser mais eficiente na entrega do vídeo, porém, a comunicação e a cooperação entre os participantes podem ser afetadas. Isso ocorre porque muitos roteadores da rede bloqueiam o tráfego de determinados protocolos [11]. Sendo assim, é importante avaliar experimentalmente o desempenho dos sistemas par-a-par em diferentes cenários para determinar a eficiência desses sistemas e identificar possíveis pontos a serem melhorados.

Outro desafio ainda maior é o desenvolvimento de sistemas par-a-par de vídeo sob demanda [2]. Embora a maioria dos sistemas de vídeo par-a-par implemente a difusão contínua de vídeo sem a possibilidade de interação do usuário, atualmente, o número de sistemas par-a-par que implementam serviços de vídeo sob demanda, que permitem parar/retroceder/avançar o vídeo, vem crescendo [1, 12, 13]. Na di-

fusão de vídeo, os participantes recebem e reproduzem o vídeo a partir do instante de sua requisição. Portanto, o tempo da requisição geralmente não corresponde ao início do vídeo. Além disso, na difusão, os participantes não podem controlar as ações de reprodução do vídeo. Portanto, os participantes têm interesse em um dado conteúdo durante um mesmo período de tempo e, conseqüentemente, o progresso da reprodução do vídeo é fortemente sincronizado entre os participantes. Os resultados dos experimentos realizados com o CoolStreaming, um dos sistemas pioneiros de vídeo par-a-par, mostram que o intervalo entre os trechos de um mesmo vídeo reproduzidos por diferentes participantes não é superior a um minuto [14]. Por isso, a troca de conteúdo de vídeo entre os participantes é facilitada, uma vez que todos têm interesse pelo mesmo trecho do vídeo ao mesmo tempo. Por outro lado, no vídeo sob demanda, cada participante pode estar assistindo a um determinado trecho do vídeo, dado que os participantes podem iniciar a reprodução do vídeo a qualquer instante e também podem interagir com o sistema durante a reprodução do vídeo. Dessa forma, o número de participantes que simultaneamente se interessam por um mesmo trecho de um vídeo tende a ser menor por causa da dessincronização da reprodução entre os participantes. Conseqüentemente, o número de pares dos quais se podem solicitar o conteúdo desse trecho também é menor. Por isso, é necessário desenvolver mecanismos específicos para aumentar a disponibilidade de conteúdo de interesse entre os pares em sistemas par-a-par de vídeo sob demanda.

1.1 Objetivos

Este trabalho tem dois objetivos principais: avaliar experimentalmente o desempenho de sistemas par-a-par de difusão de vídeo em uma rede sem fio e propor um novo mecanismo de seleção de parceiros específico para sistemas par-a-par de vídeo sob demanda.

Em primeiro lugar, avalia-se o impacto da mobilidade dos usuários no desempenho de alguns dos sistemas par-a-par de difusão de vídeo mais populares em uma rede sem fio experimental. A mobilidade é um dos principais motivos para a implantação das redes sem fio. Além disso, a difusão de TV faz parte do cotidiano das pessoas e, por isso, os usuários desejam ter acesso a aplicações de distribuição

de áudio e vídeo enquanto estão em movimento em áreas cobertas por redes sem fio. Nesse cenário, o desempenho dos sistemas par-a-par de vídeo percebido pelos usuários é afetado pela conectividade variável da rede, causada por diferentes fatores como a mudança de pontos de acesso (*handoff*), a variação da intensidade de interferências e a reconfiguração de protocolos. Para avaliar o efeito de tais fatores, experimentos práticos são realizados em um cenário de mobilidade no qual se busca reproduzir o comportamento de um usuário caminhando através de diferentes *gateways* enquanto assiste a um dado vídeo. O desempenho dos sistemas par-a-par é comparado com o desempenho de um aplicativo cliente-servidor. Os resultados dos experimentos mostram que os sistemas par-a-par de vídeo se adaptam melhor ao cenário de mobilidade do que o aplicativo cliente-servidor. Uma única mudança de *gateway* é suficiente para degradar o desempenho do cliente servidor. Além disso, mostra-se que a capacidade de adaptação dos sistemas par-a-par é diferente dependendo do tipo de protocolo de transporte usado.

Neste trabalho, é proposto também um mecanismo de seleção de parceiros específico para sistemas par-a-par de vídeo sob demanda. Os parceiros são os nós para os quais um participante envia e dos quais recebe o conteúdo de vídeo. Por isso, os mecanismos de seleção de parceiros podem ser usados para aumentar a disponibilidade do conteúdo de interesse nos parceiros e também para reduzir o tamanho do *cache* necessário em sistemas par-a-par de vídeo sob demanda. Atualmente, a maioria dos sistemas seleciona e atualiza o conjunto de parceiros de um nó aleatoriamente [14]. A seleção aleatória se justifica para os sistemas de difusão, pois todos os participantes estão interessados em um dado trecho do vídeo ao mesmo tempo. Por outro lado, no vídeo sob demanda, a seleção aleatória é menos eficiente, uma vez que os participantes entram no sistema em instantes diferentes e controlam a reprodução do vídeo. Sendo assim, a probabilidade de um participante recém-chegado ao sistema selecionar parceiros com interesses mútuos depende do tempo em que os demais participantes estão no sistema e do número de interações já feitas pelos participantes. O objetivo do mecanismo proposto neste trabalho, chamado de LIPS (*Lifetime-based Peer Selection*), é fazer com que o conjunto de parceiros selecionados por um participante tenha conteúdo do seu interesse para trocar. Para tanto, define-se uma métrica de seleção denominada tempo de vida (*lifetime*). O tempo de

vida de um participante é dado pelo tempo decorrido desde o início da reprodução do vídeo e, assim, indica há quanto tempo um participante está no sistema assistindo a um dado vídeo. O tempo de vida é diferente do tempo de reprodução. Esse último indica o ponto atual da reprodução e, portanto, é alterado a cada interação com a reprodução do vídeo. Define-se que um participante seleciona como parceiros outros participantes que possuam tempos de vida próximos ao seu. Com isso, a chance de se selecionar parceiros com interesse em um trecho de vídeo comum é maior do que na seleção aleatória, uma vez que os participantes selecionados iniciaram a reprodução do vídeo em trechos próximos. Se os parceiros estão reproduzindo as mesmas partes do vídeo, eles estão interessados no mesmo conteúdo de vídeo ao mesmo tempo. Consequentemente, o conteúdo de vídeo já recebido e reproduzido pode ser descartado e o tamanho do *cache* reduzido. A seleção de parceiros baseada no tempo de vida, entretanto, não garante a interseção de interesses porque os participantes podem interagir com o sistema e alterar o ponto de reprodução do vídeo a qualquer instante. Portanto, o comportamento dos participantes em sistemas de vídeo sob demanda influencia o desempenho dos mecanismos de seleção de parceiros. Sendo assim, a eficiência do mecanismo proposto é analisada para diferentes padrões de interatividade de usuários através de simulações. Em comparação com outros mecanismos, o LIPS seleciona parceiros mais eficientes e não exige nenhuma mensagem de controle adicional, em virtude da interatividade dos usuários. Por isso, o mecanismo proposto provê uma continuidade de reprodução do vídeo superior a 99% para a maioria dos padrões de interatividade analisados e, mesmo sendo mais eficiente do que os demais mecanismos, exige tamanhos de *cache* menores para obter tal desempenho. A continuidade é um dos parâmetros de qualidade mais importantes para as aplicações de distribuição de vídeo, uma vez que está associada diretamente à satisfação do usuário. Quando não há continuidade, um usuário fica com a tela do tocador de vídeo sem informação ou congelada.

1.2 Organização do Texto

Este trabalho está organizado da seguinte forma. No Capítulo 2, caracteriza-se o problema da distribuição de vídeo na Internet, apresentando-se as limitações da

atual arquitetura da rede. Nesse mesmo capítulo, algumas das técnicas empregadas e soluções propostas para a distribuição de vídeo, como a comunicação multidesnatária e as redes de distribuição de conteúdo, são brevemente descritas. Em seguida, no Capítulo 3, as principais características da distribuição de vídeo sobre redes par-a-par são apresentadas. Descrevem-se as diferentes arquiteturas de distribuição, ressaltando suas vantagens e desvantagens. Ainda nessa seção, exemplos de sistemas de difusão e de vídeo sob demanda são apresentados. No Capítulo 4, os resultados da avaliação experimental do impacto da mobilidade nos sistemas de difusão são apresentados. Descreve-se em detalhes o cenário utilizado, bem como a metodologia empregada nos experimentos. Discutem-se também os trabalhos relacionados à avaliação experimental de sistemas par-a-par de vídeo. No Capítulo 5, o mecanismo proposto LIPS é apresentado. O desempenho do LIPS é comparado com o desempenho da seleção aleatória e de outros dois mecanismos de seleção de parceiros para vídeos com características e padrões de interatividade diferentes. O procedimento para modelar o comportamento de usuários é descrito, assim como o simulador implementado. Discutem-se ainda os trabalhos relacionados aos sistemas par-a-par de vídeo sob demanda, especialmente, sobre os mecanismos de seleção de parceiros. Por fim, no Capítulo 6, são apresentadas as conclusões deste trabalho.

Capítulo 2

Distribuição de Vídeo na Internet*

A distribuição de vídeo na Internet traz novos desafios para o projeto de protocolos e aplicações. O primeiro desafio é garantir os requisitos de qualidade de serviço da aplicação de vídeo. O vídeo é uma mídia contínua que requer uma grande largura de banda passante, se comparada a outros tipos de mídia como dados e voz, e é sensível ao atraso e à variação do atraso (*jitter*). Apesar de ser uma mídia que apresenta alta redundância espacial e temporal, o vídeo também é sensível à taxa de perda de pacotes [15]. Quando codificado, as informações redundantes contidas no vídeo são reduzidas para diminuir o volume de informação e, conseqüentemente, o espaço ocupado em memória e o tempo de transmissão. O segundo desafio é garantir a escalabilidade da aplicação. Espera-se que o número de usuários de sistemas de distribuição de vídeo seja da ordem de milhões e, portanto, é necessário encaminhar o conteúdo de vídeo de forma eficiente para os receptores. O terceiro desafio é atender de forma satisfatória à heterogeneidade dos receptores do vídeo, visto que cada um pode possuir uma capacidade de processamento diferente e estar conectado à Internet por meio de redes de acesso com capacidades diferentes. Neste capítulo, são apresentadas algumas soluções propostas para lidar com tais desafios. Primeiramente, discutem-se as características da arquitetura atual da Internet que dificultam o desenvolvimento da distribuição de vídeo. Em seguida, apresentam-se algumas das técnicas empregadas para aumentar a eficiência da distribuição de vídeo

*Este capítulo é baseado no minicurso “Distribuição de Vídeo sobre Redes Par-a-Par: Arquiteturas, Mecanismos e Desafios” [8] apresentado no XXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC’08).

como os serviços de comunicação de grupo, as técnicas de codificação e transmissão adaptativas e as redes de distribuição de conteúdo.

2.1 A Arquitetura da Internet e suas Limitações

A Internet é uma rede baseada na técnica de comutação de pacotes. Ao contrário da técnica de comutação de circuitos, que pode ser utilizada nas redes de acesso à Internet, na comutação de pacotes não há reserva de recursos para os usuários. Isso permite o uso mais eficiente da rede uma vez que os recursos são compartilhados pelos usuários. Para a transmissão de dados, a comutação de pacotes é ideal, porém para a transmissão de mídias com restrições de tempo e perda de pacotes, como é o caso da distribuição de vídeo digital codificado, ela apresenta alguns problemas. Como existe compartilhamento, é necessária a adição de um cabeçalho em cada pacote. Isso geralmente aumenta a complexidade dos nós intermediários da rede, que precisam analisar o conteúdo do cabeçalho de cada pacote para encaminhá-lo corretamente. Além do atraso de propagação no meio, têm-se os atrasos de encapsulamento, de desencapsulamento e de processamento dos pacotes e o atraso de espera dos pacotes nas filas dos roteadores. Na Internet, o IP (*Internet Protocol*) é o protocolo responsável pelo encaminhamento dos pacotes. O IP oferece um serviço não confiável, sem estabelecimento prévio de conexão, em que os pacotes são encaminhados de acordo com a política de melhor esforço. Como não há conexão e a banda passante para cada usuário é alocada dinamicamente, de acordo com a disponibilidade da rede, não há garantia de vazão, atraso ou variação do atraso. Dessa forma, por não haver reserva de recursos para cada usuário da rede, nem diferenciação no tratamento dos pacotes encaminhados pelos roteadores, a qualidade de serviço exigida pela distribuição de vídeo não é atendida pela atual arquitetura da Internet. Portanto, é necessário o desenvolvimento de mecanismos para garantir tais requisitos.

Grande parte das aplicações da Internet se baseia no modelo cliente-servidor. Nesse modelo, é estabelecida uma comunicação ponto-a-ponto entre um cliente e o servidor, que ao receber requisições de um mesmo conteúdo de diferentes clientes, envia uma cópia do conteúdo requisitado para cada cliente. Para a distribuição de

vídeo, esse modelo não é adequado em grande escala. Uma das características da distribuição de vídeo é o envio de uma mesma informação para múltiplos receptores. Como o número de receptores potenciais é da ordem de milhões, o envio de diferentes cópias do mesmo vídeo através de múltiplos canais ponto-a-ponto entre o servidor e os receptores é inviável em termos de banda passante. Sendo assim, é necessário adotar técnicas de encaminhamento que garantam a escalabilidade de distribuição de vídeo.

Existe ainda o problema da heterogeneidade dos nós conectados a Internet e que são potenciais receptores do vídeo distribuído. Tal problema é ilustrado na Figura 2.1. A Internet é formada por redes com características distintas de banda passante. Os usuários podem estar conectados à rede através de uma linha convencional de telefonia, uma linha ADSL (*Asymmetric Digital Subscriber Line*), uma rede sem fio, etc. Deve-se considerar ainda que a banda passante da Internet é compartilhada e que os computadores de cada usuário possuem capacidades de processamento diferentes. Sendo assim, a diversidade na capacidade dos nós impede que uma única taxa de transmissão atenda satisfatoriamente a todos os receptores da distribuição de vídeo na Internet. Nesse caso, uma solução do tipo “melhor denominador comum” não é suficiente. Se a taxa de transmissão do vídeo for adaptada ao receptor de maior capacidade, haverá congestionamento nos enlaces dos receptores de menor capacidade e estouro no *buffer* desses receptores. Por outro lado, se a taxa de transmissão do vídeo for adaptada ao receptor de menor capacidade, os receptores de maior capacidade receberão um vídeo, cuja qualidade será menor do que a que eles podem receber. Portanto, é necessário que receptores com capacidades diferentes recebam vídeos com níveis de qualidade diferentes.

2.2 Técnicas e Propostas para Distribuição de Vídeo na Internet

Para contornar as limitações da atual arquitetura da Internet, algumas técnicas foram empregadas e mecanismos foram propostos para atender os requisitos da distribuição de vídeo. A seguir, algumas dessas técnicas e mecanismos são apresentadas.

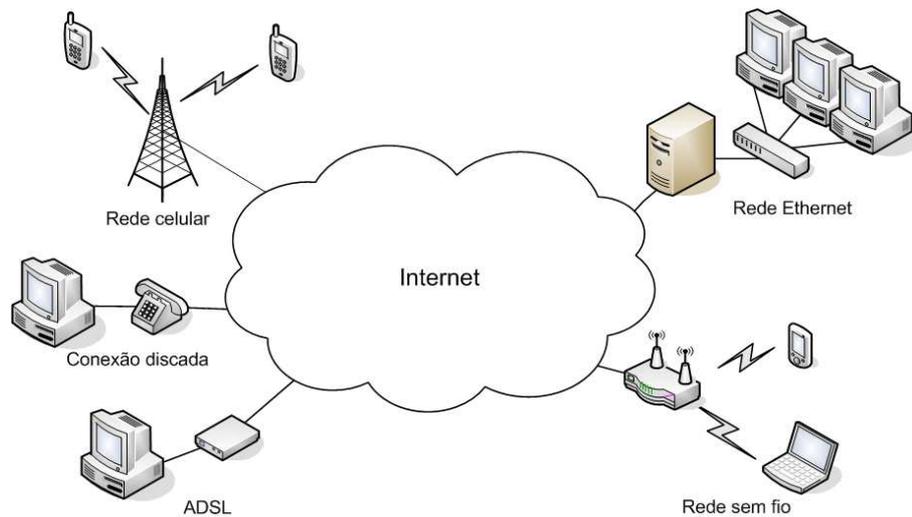


Figura 2.1: A heterogeneidade dos nós da Internet.

2.2.1 Comunicação Multidestinatória

A comunicação multidestinatória (*multicast*), também chamada de comunicação de grupo, tem como principal objetivo garantir a escalabilidade das aplicações, cuja característica é enviar uma mesma informação para múltiplos receptores. Esse é o caso da distribuição de vídeo. Busca-se aumentar a eficiência do encaminhamento de uma mensagem para evitar o consumo desnecessário de recursos da rede. Na comunicação multidestinatória, a fonte deve enviar apenas uma cópia da informação e a rede deve ser capaz de replicar essa informação somente quando necessário.

O serviço IP Multicast [16] implementa a comunicação multidestinatória na camada de rede. A ideia básica desse serviço é reunir um conjunto de estações em um grupo. Define-se um grupo como um conjunto de zero ou mais estações identificadas por um único endereço IP de destino. Dessa forma, ao enviar um datagrama para o endereço IP do grupo, todos os membros do grupo recebem o datagrama enviado. Proposto há quase 20 anos, o IP Multicast vem sendo implantado lentamente como serviço nativo na Internet, em virtude da complexidade para configurar e gerenciar o conjunto de protocolos de rede necessários à sua implementação [17]. A razão dessa complexidade deve-se ao fato do modelo de serviço IP Multicast definir um grupo como uma conversa muitos-para-muitos da qual qualquer estação pode participar sem exigência de autorização.

Em virtude da complexidade e, conseqüentemente, da lenta implantação do IP

Multicast na Internet, novas alternativas foram propostas para simplificar o serviço. Uma dessas alternativas é a comunicação multidestinatória com fonte específica (*Source-Specific Multicast* - SSM) [18, 19]. O serviço SSM usa o conceito de canal, proposto pelo protocolo EXPRESS (*EXPLICITly REquested Single Source multicast*) [20]. Um canal possui apenas uma fonte por grupo e é identificado por um par de endereços: o endereço da fonte e o endereço do grupo. Somente a fonte pode enviar datagramas para as estações pertencentes ao grupo. Consequentemente, a conversação é reduzida para um-para-muitos, o que simplifica a arquitetura do serviço.

Tanto o serviço IP Multicast quanto o SSM implementam a comunicação multidestinatória na camada de rede. Por isso, a implantação desses serviços requer modificações na infraestrutura de rede. Os roteadores têm de ser capazes de lidar com os protocolos de roteamento *multicast* e também devem manter estados por grupo, o que aumenta a complexidade e compromete a escalabilidade do serviço [21]. Uma alternativa para contornar tais problemas é implementar a comunicação multidestinatória na camada de aplicação, também conhecida por *multicast* aplicativo (*Application-Level Multicast* - ALM) [22, 23, 24]. A ideia é que o roteamento, o gerenciamento dos participantes e a replicação das informações sejam de responsabilidade das próprias estações que participam da comunicação. Todas essas funções são implementadas assumindo a existência apenas do serviço ponto-a-ponto. Para tanto, uma rede sobreposta (*overlay*) deve ser construída na camada de aplicação. Um exemplo de rede sobreposta é apresentado na Figura 2.2. Essa rede é composta de túneis ponto-a-ponto entre duas estações. Esses túneis são os enlaces da rede sobreposta. As próprias estações são responsáveis pelo encaminhamento dos pacotes. Dessa forma, como todos os pacotes são transmitidos de acordo com o serviço ponto-a-ponto, não são necessárias modificações na infraestrutura de rede nem o armazenamento de estados nos roteadores. Com isso, é possível acelerar a implantação de serviços de comunicação multidestinatória na Internet e torná-los mais escaláveis.

Existem duas possíveis formas de implementação da rede sobreposta na camada de aplicação. Na primeira, um conjunto de servidores ou *proxies*, geralmente chamados de nós âncoras, são posicionados em pontos específicos da rede [25, 26, 27].

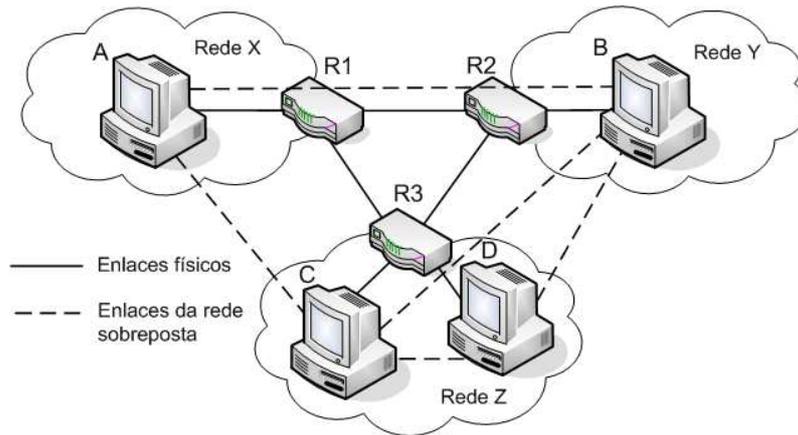


Figura 2.2: Um exemplo de uma rede sobreposta (*overlay*).

A rede sobreposta é construída e mantida apenas pelos nós âncoras. Toda estação que deseja participar da comunicação deve se conectar a um nó âncora específico. Em uma segunda abordagem, não existem nós dedicados à construção da rede sobreposta. A rede é construída e mantida por todos os participantes da comunicação de forma autônoma e cooperativa [28, 29, 14, 30]. Em troca da utilização do serviço, os nós compartilham seus recursos, o que caracteriza um sistema par-a-par. Portanto, os sistemas par-a-par podem ser usados para implementar a comunicação multidestinatória na camada de aplicação. Esse é o caso da distribuição de vídeo sobre redes par-a-par que é detalhada no Capítulo 3.

Embora eficiente, a comunicação multidestinatória na rede sobreposta da camada de aplicação não apresenta o mesmo desempenho da comunicação multidestinatória implementada na camada de rede. É difícil impedir que os túneis que formam os enlaces da rede sobreposta atravessem os mesmos enlaces físicos e, assim, não há como evitar a redundância de tráfego nesses enlaces. Além disso, o tráfego entre estações é encaminhado por outras estações que não são dedicadas a essa tarefa e que, portanto, podem acarretar em aumento da latência. A Figura 2.3 mostra as diferenças entre a comunicação ponto-a-ponto e a comunicação multidestinatória na camada de rede e na camada de aplicação. A topologia da rede do exemplo é composta por dois roteadores (R_1 e R_2) e por quatro estações (A , B , C , e D). Cada enlace possui um custo, indicado pelo valor sobreposto à linha que o representa. O custo do enlace entre os roteadores é maior do que o dos demais enlaces. No exemplo, a estação A deseja enviar uma mesma informação para as outras três estações.

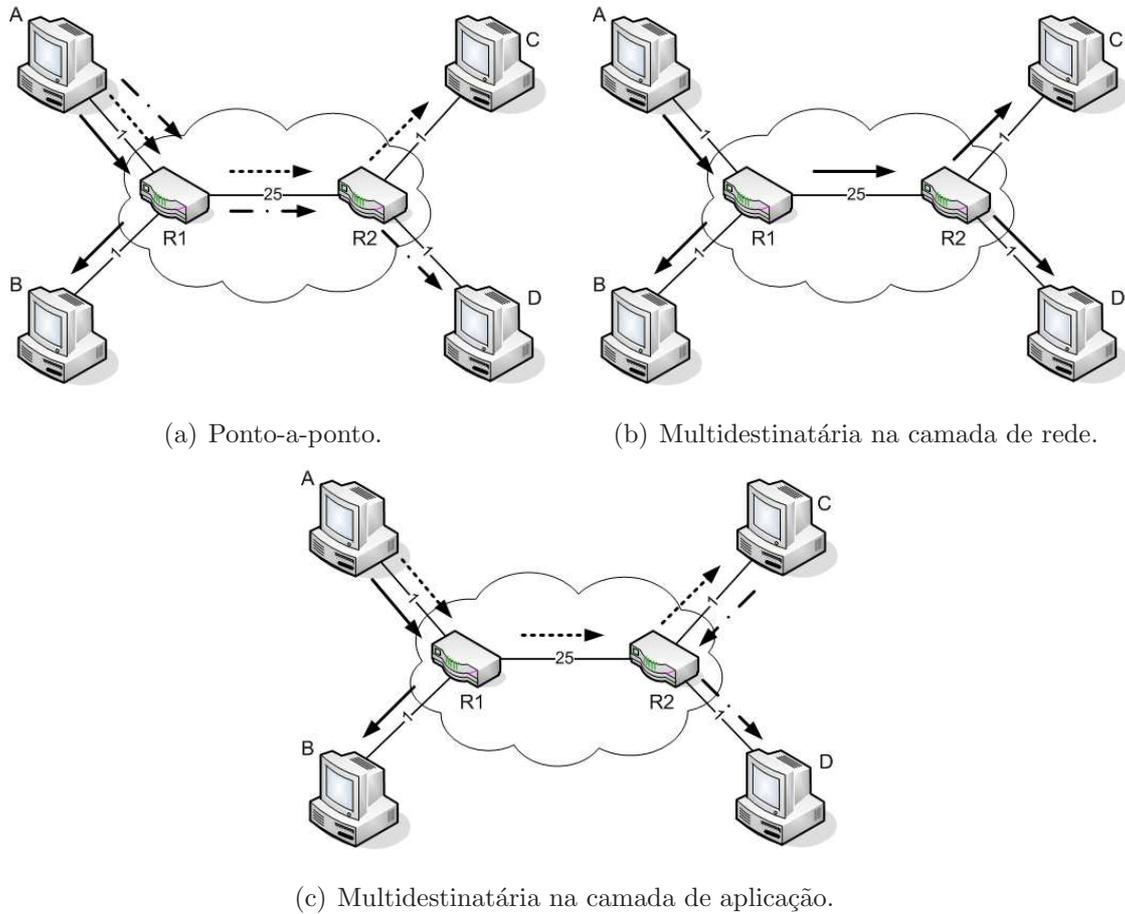


Figura 2.3: As diferenças entre a comunicação ponto-a-ponto e multidestinatória.

A Figura 2.3(a) mostra como essa informação é enviada por *A* usando o serviço ponto-a-ponto. Nota-se que existe uma grande redundância de tráfego nos enlaces próximos à fonte, bem como cópias da mesma informação em enlaces de maior custo. A Figura 2.3(b) mostra a árvore de distribuição construída por protocolo de roteamento multidestinatório. Nota-se que não existem transmissões redundantes. Apenas uma cópia da informação é transportada em cada enlace da rede. Além disso, nessa configuração, o atraso da informação recebida pelas estações é o mesmo, caso a informação tivesse sido enviada por *A* usando o serviço ponto-a-ponto. A Figura 2.3(c) mostra a árvore de distribuição construída por um sistema par-a-par na camada de aplicação. Observa-se que o número de cópias redundantes nos enlaces próximos à fonte é menor se comparado à comunicação ponto-a-ponto e que apenas uma cópia da informação foi transportada pelo enlace de maior custo. Notam-se ainda as diferenças de desempenho entre a comunicação multidestinatória na camada de aplicação e na de rede. Na camada de aplicação, ainda existe uma cópia

redundante nos enlaces $A-R_1$ e $C-R_2$ e o custo do caminho entre A e D aumentou. O custo desse caminho que era igual a 27 na comunicação multidesinatária na camada de rede passou a ser 29 na camada de aplicação. Constatase assim que existe um compromisso entre desempenho e facilidade de implantação, quando se compara as formas de implementação da comunicação multidesinatária.

2.2.2 Codificação e Transmissão Adaptativas

As aplicações de distribuição de vídeo devem se adaptar às condições da rede para atender satisfatoriamente aos seus diferentes receptores. Para isso, as aplicações devem responder aos congestionamentos nos enlaces da rede e também enviar aos receptores, que possuem diferentes capacidades de banda e processamento, vídeos de qualidades diferentes. Tais funcionalidades podem ser obtidas com a adoção de técnicas de codificação e transmissão de vídeo adaptativas [31]. A ideia básica dessas técnicas é adaptar a taxa de transmissão do vídeo às condições dos receptores.

Uma proposta para atender aos receptores heterogêneos e evitar a redundância da replicação é a codificação em camadas [32, 33]. Nessa técnica, ilustrada na Figura 2.4, um codificador comprime o vídeo em uma ou mais camadas hierárquicas com diferentes prioridades. A camada que contém as informações essenciais do fluxo de vídeo é chamada de camada base. Essa é a camada de mais alta prioridade. Todo receptor deve ser capaz de receber pelo menos essa camada de base. Camadas adicionais com prioridade decrescente podem ser codificadas para refinar a qualidade da camada base. Essas camadas de refinamento são somadas à camada base, ou a outras camadas de refinamento com maior prioridade, aumentando progressivamente a qualidade da informação de vídeo reconstruída. No exemplo, consideram-se duas camadas de refinamento além da camada base. O refinamento pode estar relacionado à taxa de transmissão, à resolução, à taxa de quadros ou à relação sinal-ruído com que o fluxo de vídeo chega ao receptor. No exemplo da Figura 2.4, a taxa do vídeo enviado é igual a 1 Mb/s. Assume-se que o vídeo está dividido em três camadas: a camada base com taxa de 128 kb/s e mais duas camadas de refinamento 1 e 2 com taxas de 384 kb/s e 512 kb/s, respectivamente. Dessa forma, os receptores A , B e C recebem fluxos de vídeo com qualidades diferentes, uma vez que recebem um número diferente de camadas. Assume-se a existência da comunicação

multidestinatória. Dessa forma, cada uma das camadas pode ser transmitida para um endereço multidestinatório e, assim, cada receptor pode associar-se a um ou mais grupos de acordo com as suas limitações de banda passante. Cada uma das camadas é transmitida sobre uma árvore de distribuição diferente. A hierarquia entre as camadas é o principal obstáculo para a adoção da codificação em camadas na prática. Para garantir que a camada base seja entregue sem erros, a rede tem que tratar diferenciadamente os pacotes de cada camada e também retransmitir os pacotes perdidos da camada base. Dependendo da configuração da rede e do atraso tolerado pela aplicação, não é possível realizar essas tarefas.

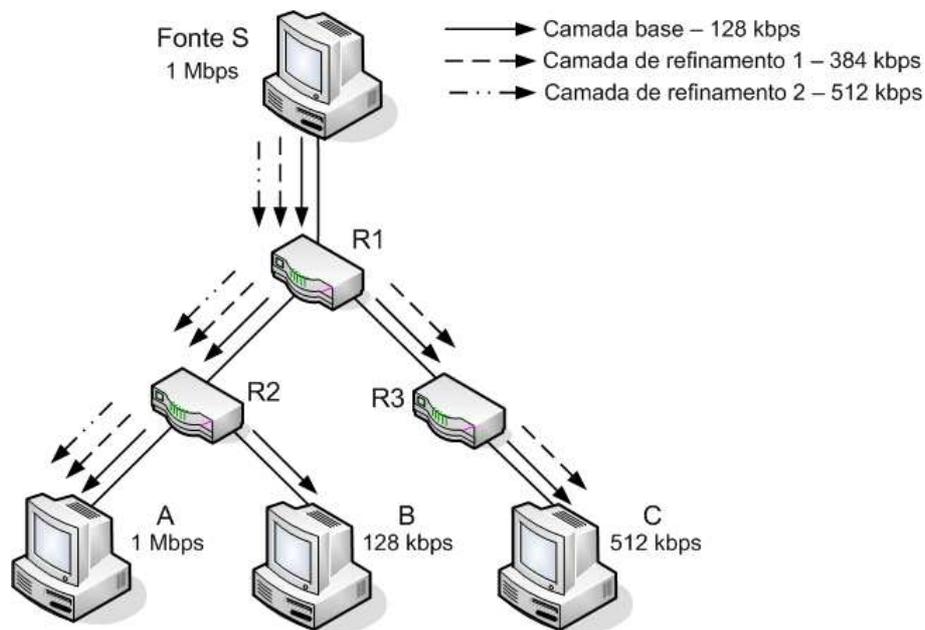


Figura 2.4: Um exemplo da codificação em camadas.

A codificação em múltiplos descritores (*Multiple Description Coding - MDC*) [34] busca simplificar a codificação em camadas. Na MDC, um fluxo de vídeo é codificado em diferentes subfluxos independentes, chamados de descritores. Cada descritor possui a mesma taxa e a mesma importância. Como não há hierarquia entre os descritores, todos os pacotes são tratados sem diferenciação e não é necessário retransmitir os pacotes perdidos. Os pacotes relativos a um descritor podem ser enviados em um mesmo caminho ou em caminhos disjuntos e, dessa forma, aumenta-se a robustez da aplicação. A qualidade do vídeo recebido depende do número de descritores recebidos em paralelo. Assim como na codificação em camadas, quanto mais

descritores recebidos, melhor é a qualidade do vídeo. Porém, cada descritor pode ser decodificado independentemente dos demais e é suficiente para gerar uma aproximação do conteúdo original. Isso facilita a manutenção da continuidade do vídeo. Atualmente, existem protótipos de codificadores MDC que conseguem gerar dois ou três descritores [6].

2.2.3 Redes de Distribuição de Conteúdo

O objetivo das redes de distribuição de conteúdo (*Content Distribution Networks* - CDNs) é aumentar a eficiência e a escalabilidade do modelo cliente-servidor para a distribuição de vídeo. Uma CDN é formada por um conjunto de nós interconectados através da Internet que cooperam de forma transparente para distribuir um conteúdo específico aos usuários [35]. Esses nós são servidores ou nós auxiliares que, geralmente, estão espalhados geograficamente e pertencem a *backbones* diferentes, como mostra a Figura 2.5. A ideia básica das CDNs é replicar o conteúdo do servidor de origem para servidores ou nós auxiliares mais próximos dos clientes. Espera-se que, dessa forma, o servidor entregue o conteúdo aos clientes com uma maior taxa de transferência e uma menor latência. Geralmente, quanto mais próximo de um cliente está o servidor, mais eficiente é a distribuição. Por isso, quanto mais servidores e nós auxiliares, melhor o desempenho de uma CDN. Entretanto, quanto maior o número e a capacidade dos nós de uma CDN, maior é o seu custo de implantação.

Uma rede de distribuição de conteúdo possui dois tipos de servidores: o servidor de origem e o servidor de réplica. O servidor de origem é o responsável pelo armazenamento, atribuição de identificadores e divulgação do conteúdo. O servidor de réplica, por sua vez, é responsável por encaminhar o conteúdo para um dado cliente. O funcionamento simplificado de uma rede de distribuição de conteúdo é ilustrado na Figura 2.5. Um cliente interessado em um dado conteúdo, no exemplo o nó A , envia uma requisição para o servidor de origem S (passo 1). Uma vez recebida a requisição de um conteúdo, o servidor de origem deve ser capaz de encaminhar essa requisição para o servidor de réplica mais próximo do cliente. No exemplo, o servidor de réplica X é escolhido por estar a um salto do nó A (passo 2). Nesse exemplo, assume-se que o conteúdo solicitado já está disponível no servidor de réplica X . Uma vez disponível, o conteúdo é entregue ao nó A pelo servidor

de réplica X (passo 3). Na prática, o encaminhamento da requisição, a escolha do servidor de réplica e a forma como o conteúdo do servidor de origem é replicado para os servidores de réplica são os principais desafios em uma CDN [35].

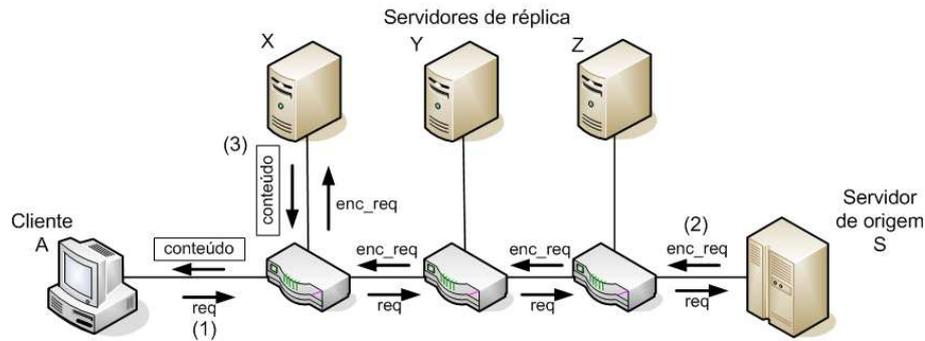


Figura 2.5: O funcionamento de uma rede de distribuição de conteúdo.

Existem diversos exemplos de redes de distribuição de conteúdo acadêmicas [36, 37] e comerciais [38, 39, 40]. A rede comercial mais popular é a Akamai [41]. Estima-se que essa rede possui mais de 19 mil servidores espalhados pela Internet. A Akamai foi usada para transmitir ao vivo o concerto Live Earth realizado simultaneamente em diferentes cidades do mundo e que teve a duração de 24 horas. Durante esse concerto, foram distribuídos mais de 15 milhões de fluxos de vídeo com um pico de 237 mil usuários simultâneos [42].

Capítulo 3

Distribuição de Vídeo sobre Redes Par-a-Par*

O uso de sistemas par-a-par é uma das soluções mais promissoras para tornar a distribuição de vídeo na Internet mais escalável e eficiente [3]. Esses sistemas se baseiam no compartilhamento de recursos entre seus participantes e não exigem modificações no núcleo da rede. Por isso, são escaláveis e têm custo reduzido. Neste capítulo, os fundamentos da distribuição de vídeo sobre redes par-a-par são apresentados. As diferentes arquiteturas de distribuição empregadas pelos sistemas par-a-par de vídeo são detalhadas. Além disso, as diferenças entre os sistemas de difusão e de vídeo sob demanda são apresentadas e, por fim, tais sistemas são exemplificados.

3.1 Fundamentos dos Sistemas Par-a-Par de Vídeo

Os sistemas par-a-par têm como principal característica a escalabilidade. Nesses sistemas, os nós participantes compartilham seus recursos e colaboram para o funcionamento e manutenção do sistema. Com isso, quanto maior o número de nós participantes, maior é a capacidade do sistema, visto que mais nós compartilham

*Este capítulo é baseado no minicurso “Distribuição de Vídeo sobre Redes Par-a-Par: Arquiteturas, Mecanismos e Desafios” [8] apresentado no XXVI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC’08).

seus recursos. Além disso, os próprios nós constroem a estrutura de comunicação entre os pares, sem que haja necessidade de modificações na infra-estrutura de rede. Essas características são fundamentais para o desenvolvimento da distribuição de vídeo em larga escala e, por isso, fazem dos sistemas par-a-par uma alternativa promissora para a implantação desse serviço na Internet.

O enorme sucesso dos sistemas par-a-par para o compartilhamento de arquivos também é um indicativo do potencial do uso de sistemas similares para distribuir vídeo na Internet. Porém, se comparada ao compartilhamento de arquivos, a distribuição de vídeo possui diferentes requisitos e introduz novos desafios. A maior diferença está relacionada ao tipo de mídia distribuída. Nos sistemas de compartilhamento de arquivos, trabalha-se com longas transferências de dados sem restrições de tempo. Nos sistemas de vídeo, ao contrário, a distribuição deve atender a requisitos estritos de banda passante e tempo para garantir a cadência e a qualidade do vídeo nos receptores. Além disso, os sistemas de compartilhamento de arquivos têm como objetivo construir arquiteturas de distribuição que tornem a indexação e a busca de arquivos mais eficientes. Na distribuição de vídeo, por outro lado, o objetivo é tornar a comunicação entre os pares mais eficiente. Outra diferença é o número de usuários simultâneos em cada tipo de sistema. Um programa de TV distribuído por um sistema de vídeo, por exemplo, atrai mais usuários simultâneos do que um arquivo específico disponibilizado por um sistema de compartilhamento de arquivos. Com mais usuários simultâneos e, conseqüentemente, mais recursos compartilhados, é possível atender os requisitos de qualidade de serviço do vídeo.

Um sistema de distribuição de vídeo par-a-par é composto por aplicativos e mecanismos responsáveis pela construção e gerenciamento da estrutura de distribuição e pelo encaminhamento do vídeo. Tais tarefas são realizadas de forma distribuída pelos próprios nós participantes, geralmente, sem o auxílio de uma entidade central. Um sistema de vídeo par-a-par funciona da seguinte forma. Os nós se auto-organizam e constroem uma rede sobreposta na camada de aplicação. Uma vez construída a rede, os nós interessados em um mesmo conteúdo de vídeo se organizam em uma arquitetura de distribuição. Os nós que compõem a arquitetura de distribuição recebem o fluxo de vídeo desejado e também encaminham esse fluxo para outros nós da arquitetura. Cria-se assim um ambiente colaborativo, no qual os nós contribuem

para o encaminhamento do vídeo com banda passante, processamento e espaço de armazenamento.

O desempenho de um sistema de distribuição de vídeo par-a-par está relacionado com a arquitetura de distribuição que esse sistema implementa. A arquitetura de distribuição define a forma como o conjunto de nós participantes se organiza para encaminhar o fluxo de vídeo, bem como os mecanismos e protocolos necessários para o gerenciamento dos participantes do sistema. Primeiramente, o encaminhamento deve ser eficiente em termos de banda passante e atraso para garantir os requisitos de qualidade de serviço do vídeo distribuído. Para também garantir a escalabilidade do sistema, a arquitetura deve suportar um número grande de nós participantes. Ainda, à medida que o número de pares aumenta, a sobrecarga de controle para a sua construção e manutenção não deve afetar o desempenho do sistema. O sistema de distribuição também tem de lidar com a dinâmica de entrada e saída de pares do sistema e com a heterogeneidade desses pares. Durante a entrada e saída de participantes, a coesão da estrutura de encaminhamento de dados deve ser mantida e perdas de informação causadas pela saída de nós do sistema devem ser minimizadas. Além disso, como os nós possuem capacidades diferentes de processamento e armazenamento e estão conectados à Internet através de redes de acesso de diferentes capacidades, deve-se garantir que os recursos exigidos de um nó participante não excedam a capacidade desse nó. Também é desejável que o sistema implemente mecanismos para evitar que um nó de menor capacidade afete o desempenho de todo o sistema, bem como mecanismos que distribuam vídeos de diferentes qualidades para pares com diferentes capacidades. A seguir, as arquiteturas propostas para os sistemas par-a-par de vídeo são detalhadas.

3.2 Arquiteturas de Distribuição

Basicamente, existem duas formas de arquiteturas de distribuição em sistemas de vídeo par-a-par: a arquitetura em árvore e a arquitetura em malha. Nessa seção, as arquiteturas em árvore e em malha são caracterizadas e suas vantagens e desvantagens são analisadas.

3.2.1 Arquitetura de Distribuição em Árvore

A ideia básica da arquitetura de distribuição em árvore é semelhante à ideia do serviço IP Multicast. Os nós se organizam para formar uma árvore de distribuição, por onde é encaminhado um dado fluxo de vídeo. A raiz da árvore é a fonte de vídeo e, geralmente, é construída uma árvore para cada fluxo de vídeo. Os nós pertencentes à árvore possuem relações de pai e filho, ou seja, quando um nó recebe um pacote de seu pai, ele encaminha uma cópia desse pacote para cada um de seus filhos, sem que haja uma requisição explícita por parte dos filhos. Dessa forma, o fluxo de vídeo é “empurrado” para os receptores. Por exemplo, na Figura 3.1, o nó *A* recebe o fluxo de vídeo diretamente da fonte *S* e encaminha para seus filhos *C* e *D*. Nenhum outro nó envia pacotes de vídeo para o nó *A*.

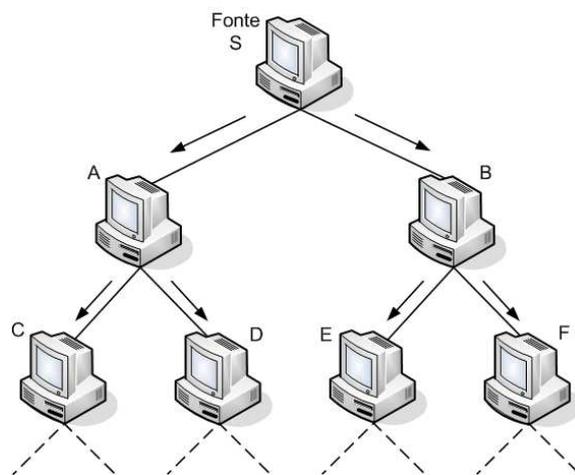


Figura 3.1: A arquitetura de distribuição em árvore.

As vantagens da arquitetura em árvore são a baixa latência e a baixa sobrecarga de controle quando a entrada e saída de pares não é frequente, pois uma vez construída a árvore somente os pacotes de vídeo são encaminhados sem a necessidade do envio de mensagens de controle. Entretanto, se a saída de nós é frequente, a sobrecarga de controle tende a ser alta, uma vez que a árvore deve ser reconstruída a cada uma dessas saídas. Quando um nó deixa a árvore, todos os seus nós descendentes deixam de receber o fluxo de vídeo, o que pode acarretar descontinuidades de recepção. Sendo assim, o tempo de reparo da árvore deve ser pequeno a fim de reduzir a perda de pacotes causada pela saída de um nó. Uma alternativa para amenizar tal problema é, durante a reconstrução da árvore, um nó pai armazenar

os pacotes de vídeo que deveriam ser encaminhados para o seu filho que deixou a árvore. Outra desvantagem da arquitetura em árvore é que os pacotes de um fluxo de vídeo seguem o mesmo caminho até um determinado receptor. Dessa forma, o algoritmo de construção da árvore deve balancear o número de filhos atribuídos a cada pai para evitar possíveis congestionamentos na rede. Outra característica da arquitetura em árvore que prejudica o seu desempenho é que a maioria dos nós participantes são folhas da árvore e, com isso, não contribuem com seus recursos no encaminhamento do fluxo de vídeo para outros nós, uma vez que não possuem filhos. A heterogeneidade dos receptores também pode degradar o desempenho do encaminhamento na arquitetura em árvore. Se um nó pai tem menor capacidade de banda passante do que seus filhos, conseqüentemente, esses filhos receberão um vídeo com uma qualidade mais baixa do que poderiam receber.

Uma proposta para tornar a arquitetura em árvore mais robusta à dinâmica de entrada e saída de nós e menos sensível à heterogeneidade dos receptores é o uso de múltiplas árvores [43, 44, 45]. A ideia básica é fazer com que receptores de diferentes capacidades recebam vídeos de diferentes qualidades. Para tanto, a fonte deve utilizar um codificador em camadas ou MDC, descritos na Seção 2.2.2. Ao utilizar tais codificadores, a fonte divide um determinado fluxo de vídeo em subfluxos. Cada subfluxo, então, é encaminhado através de uma árvore diferente. Cada nó, por sua vez, determina em quantas árvores deve se inscrever de acordo com a sua banda passante. Quanto maior o número de árvores em que um nó está inscrito, maior será a qualidade do vídeo recebido por esse nó. Um subfluxo é encaminhado em cada árvore da mesma forma que um fluxo. Um nó pai ao receber um pacote encaminha uma cópia desse pacote para os seus filhos. Durante o processo de construção da estrutura multiárvore, um dado nó é obrigatoriamente interno em uma e somente uma das árvores. Nas demais árvores, ele deve ser obrigatoriamente uma folha. O objetivo desse procedimento é minimizar os efeitos da saída de um nó e maximizar a utilização da banda passante compartilhada pelos nós. Caso um nó esteja inscrito em mais de uma árvore, a saída de um antecessor desse nó não provoca descontinuidades no vídeo recebido. Somente a qualidade do vídeo recebido será mais baixa. Além disso, todos os nós contribuem com banda passante para o encaminhamento do vídeo, visto que não podem ser folhas em todas as árvores.

3.2.2 Arquitetura de Distribuição em Malha

A ideia básica da arquitetura de distribuição de vídeo em malha é a mesma de sistemas de compartilhamento de arquivos como o BitTorrent [46]. Nesses sistemas, um arquivo é dividido em pedaços (*chunks*) que são espalhados pelos nós participantes. Na distribuição de vídeo, ao invés de um arquivo, divide-se um fluxo de vídeo em pedaços. Como os nós se organizam e interagem para receber e encaminhar um determinado conteúdo, assim como fazem as abelhas em uma colméia, o conjunto de nós participantes é geralmente chamado de enxame (*swarm*). Na Figura 3.2, por exemplo, o nó *F* recebe os pedaços de vídeo da fonte *S* de todos os seus parceiros – os nós *B*, *C*, *E*, *G* e *I* – e também encaminha esses pedaços para os parceiros. A fonte *S* é o único nó que só encaminha o fluxo de vídeo. Portanto, semelhante à arquitetura em árvore há apenas uma fonte de vídeo. Porém, na arquitetura em malha, cada nó pode receber vídeo de mais de um nó do sistema sem a organização hierárquica das árvores de distribuição.

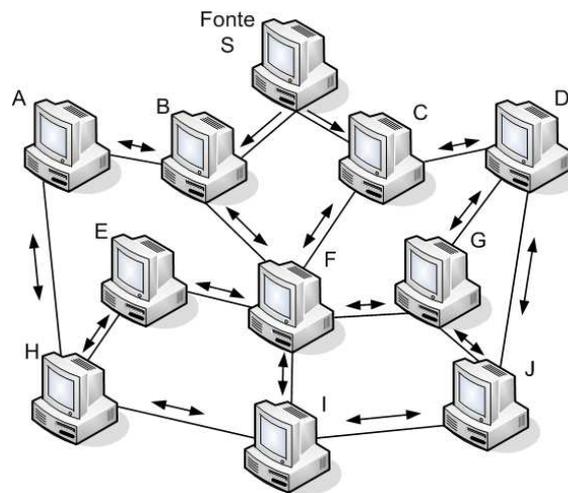


Figura 3.2: A arquitetura de distribuição em malha.

A principal diferença da arquitetura em malha para a arquitetura em árvore é que na primeira não se constrói e nem se mantém uma estrutura explícita para o encaminhamento do fluxo de vídeo. Argumenta-se que ao invés de reparar constantemente a estrutura de distribuição, em virtude da dinâmica de entrada e saída de participantes em um sistema par-a-par, é mais eficiente disseminar a disponibilidade dos pedaços de vídeo entre os nós para que cada nó solicite explicitamente os

pedaços que deseja. Para tanto, cada nó participante mantém um conjunto de nós, geralmente chamados de parceiros, com os quais periodicamente troca informações sobre os pedaços do vídeo que possui. Assim, um nó sabe quais parceiros possuem quais pedaços do vídeo e, dessa forma, requisita o envio dos pedaços desejados aos respectivos parceiros. Por isso, diz-se que na arquitetura em malha os receptores “puxam” os pedaços do vídeo. Vale ressaltar que os pedaços são enviados usando comunicações ponto-a-ponto na camada de aplicação entre um nó e seus parceiros.

A vantagem da arquitetura em malha é ser menos susceptível à entrada e à saída de nós do que a arquitetura em árvore. Primeiro, porque o conteúdo de vídeo é dividido em pedaços que estão disponíveis em diversos nós. Segundo, porque um nó recebe esses diferentes pedaços de diferentes parceiros e não somente de um nó pai, como ocorre em uma árvore de distribuição. Essas duas características reduzem a possibilidade de descontinuidades de recepção. Mesmo com a saída de um parceiro, um nó continua a receber os pedaços de vídeo, já que os pedaços que eram armazenados pelo nó que deixou o sistema podem ainda ser encontrados nos parceiros remanescentes. Por outro lado, a sobrecarga de controle, geralmente, é maior na arquitetura em malha do que na arquitetura em árvore, pois é necessário trocar as informações sobre a disponibilidade dos pedaços de vídeo entre os parceiros e enviar as requisições dos pedaços. Como não há uma estrutura explícita de distribuição na arquitetura em malha, os atrasos de inicialização e encaminhamento do vídeo tendem a ser maiores do que na arquitetura em árvore. Além disso, o desempenho dos sistemas baseados na arquitetura em malha depende do tamanho dos *buffers* nos nós, já que os pedaços de vídeo podem ser recebidos fora da ordem de reprodução e terão de ser armazenados. Também, quanto maiores os *buffers* dos nós, mais pedaços de vídeo eles podem armazenar. Isso aumenta a disponibilidade do vídeo na rede, favorecendo a aplicação.

3.3 Sistemas de Difusão de Vídeo

Nos sistemas de difusão, uma fonte envia o mesmo conteúdo de vídeo para um grupo de usuários, como ocorre em uma transmissão de TV. Um usuário recebe e reproduz o vídeo a partir do instante de sua requisição e não do início do vídeo.

Ele também não controla a reprodução do vídeo. Os sistemas de difusão empregam tanto a arquitetura em árvore quanto a em malha. O ESM/Narada [28] é um exemplo de sistema par-a-par de difusão de vídeo baseado na arquitetura em árvore. O SplitStream [43], por sua vez, emprega a arquitetura multiárvore. No entanto, a maioria e os mais populares sistemas par-a-par de difusão de vídeo empregam a arquitetura em malha. São exemplos de sistemas dessa arquitetura o SopCast [5], o PPLive [5] e o TVAnts [10]. O funcionamento desses sistemas é baseado no CoolStreaming/DONet [14], um dos sistemas pioneiros e que, por isso, é detalhado na Seção 3.3.1. Existem ainda sistemas híbridos, que buscam combinar as características das duas arquiteturas para criar sistemas mais eficientes e robustos. São exemplos desses sistemas o GridMedia [9, 47] e o CoolStreaming+ [48].

3.3.1 CoolStreaming/DONet

O CoolStreaming¹ (*Cooperative Overlay Streaming*) é a implementação do sistema DONet (*Data-driven Overlay Network*) [14]. O funcionamento do CoolStreaming se baseia na divisão de um determinado vídeo em pedaços. Cada participante troca periodicamente informações sobre a disponibilidade dos pedaços com um conjunto de nós, chamados de parceiros. Um nó solicita aos seus parceiros os pedaços de vídeo que não possui e disponibiliza os pedaços que possui. Para cada pedaço de vídeo, um nó pode ser tanto receptor quanto emissor. A exceção é a fonte do vídeo, que sempre é emissor, e por isso é chamada de nó origem. A fonte pode ser um servidor de vídeo dedicado ou simplesmente um nó participante que possui um conteúdo de vídeo para distribuir.

A principal característica do CoolStreaming é não manter uma estrutura de distribuição global para encaminhar um fluxo de vídeo. Um nó se conecta à rede sobreposta através de seus parceiros, como será detalhado adiante, e dessa forma a malha de distribuição é formada. Para cada fonte, forma-se uma malha de distribuição. Os nós também não possuem funções predefinidas, como os nós pais e nós filhos em uma árvore. Todo nó participante do sistema tem de encaminhar o conteúdo de vídeo para outros nós que desejam receber esse conteúdo. Diz-se que,

¹O módulo cliente do sistema CoolStreaming está disponível para os sistemas operacionais Windows e também na forma de *plugin* para ser embutido em navegadores *web* [49].

por não haver uma estrutura que restrinja a direção do encaminhamento do vídeo, a disponibilidade é que direciona o encaminhamento de um fluxo de vídeo.

Um nó participante do CoolStreaming deve ser capaz de manter uma lista, ao menos parcial, dos outros participantes ativos do sistema, estabelecer e manter parcerias com outros participantes e escalonar a transmissão dos pedaços de vídeo. Essas tarefas são, respectivamente, de responsabilidade do gerenciador de participantes, do gerenciador de parceiros e do escalonador, mostrados na Figura 3.3. A seguir, descreve-se como um novo nó participa da distribuição e como seus parceiros são selecionados. Discute-se também como a informação de disponibilidade de pedaços é mapeada e difundida entre os parceiros.

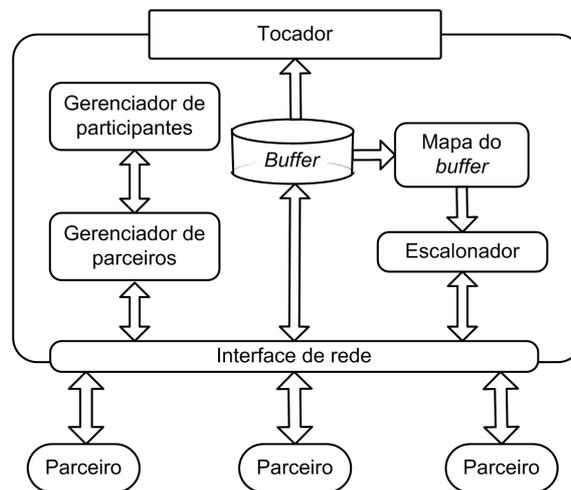


Figura 3.3: A arquitetura de um nó participante do sistema CoolStreaming.

Construção da Malha de Distribuição

Cada nó participante do CoolStreaming possui um identificador único, tal como o endereço IP, e mantém um *cache* de participantes, chamado de (*membership cache* - mCache), que contém uma lista parcial de identificadores dos nós ativos do sistema. A razão pela qual essa lista é parcial é explicada adiante. Para entrar no sistema, um nó deve primeiro contatar o nó origem, nesse caso, a fonte do vídeo. Ao ser contatada, a origem escolhe aleatoriamente um nó do seu mCache que será o nó adjunto. Em seguida, a origem redireciona o novo nó para o nó adjunto que, por sua vez, envia uma lista de possíveis parceiros ao novo nó. Uma vez recebida a lista, o novo nó contata os candidatos, estabelece as parcerias e passa a fazer parte da malha

de distribuição. O uso de nós adjuntos para enviar a lista de possíveis candidatos reduz a carga do nó de origem e torna mais uniforme a seleção de parceiros [14]. Para que o procedimento descrito seja viável, o nó origem tem de estar ativo durante a distribuição do vídeo e o seu endereço e/ou identificador tem de ser conhecido. O CoolStreaming não define a forma como o endereço da origem é obtido. Esse endereço, por exemplo, poderia ser divulgado em um sítio da Internet. A Figura 3.4 exemplifica a entrada de um nó no sistema. Nesse exemplo, o nó N deseja participar do sistema e, então, envia uma mensagem para a fonte S (passo 1). Assume-se que o identificador de S é conhecido pelo nó N . Ao receber a mensagem de N , a fonte S responde com o identificador do nó adjunto, nesse caso o nó D (passo 2). Com a resposta, o nó N envia uma outra mensagem para o nó adjunto D (passo 3), que responde a N com a lista de possíveis candidatos a parceiros, no caso A e B (passo 4). Em seguida, N envia uma mensagem para estabelecer parcerias com os nós A e B (passo 5). Como ambos respondem positivamente (passo 6), os enlaces entre os nós N e A e entre os nós N e B são criados (passo 7). A partir desse ponto, N se torna um participante do sistema.

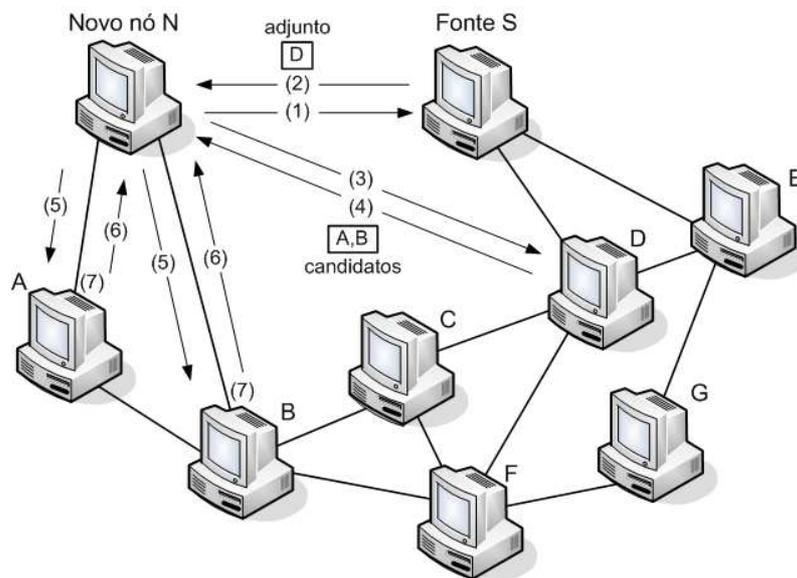


Figura 3.4: A entrada de um novo nó no sistema CoolStreaming.

Um ponto fundamental do funcionamento do CoolStreaming é a criação e a atualização do *cache* de participantes. Para acompanhar a dinâmica da malha, cada nó periodicamente envia a um conjunto de nós uma mensagem para anunciar

sua existência e, conseqüentemente, mostrar que é um participante ativo. Como esse conjunto de nós é definido será visto a seguir. Cada mensagem possui quatro campos: o número de seqüência da mensagem, o identificador do nó, o número atual de parceiros do nó e o tempo de vida, que indica o tempo de validade remanescente da mensagem. As mensagens são difundidas através do protocolo SCAM (*Scalable Gossip Membership*) [50]. Como na maioria dos protocolos baseados em fofocas (*gossip*), no SCAM um nó gera uma mensagem de existência e a envia para um conjunto aleatório de nós escolhidos do seu *cache* de participantes. Ao receberem essa mensagem, os nós escolhidos também sorteiam outros nós, armazenados nos seus *caches* de participantes, e encaminham a mensagem recebida para os nós sorteados. Dessa forma, a mensagem se espalha pela rede. Ao receber uma mensagem de existência com um número de seqüência não conhecido, o nó verifica o identificador de nó contido nessa mensagem. Caso o nó seja conhecido, a entrada do *cache* de participantes referente a esse nó é atualizada. Do contrário, se o nó não é conhecido, uma nova entrada é criada. Uma entrada do *cache* de participantes contém os quatro campos contidos na mensagem de existência e mais um campo no qual é armazenado o instante da última atualização da entrada. Diz-se que o mCache contém uma lista parcial dos participantes do sistema devido à aleatoriedade do algoritmo usado para difundir as mensagens de existência e por não haver troca do *cache* de participantes entre os nós do sistema.

Um nó periodicamente pode estabelecer novas parcerias com outros nós sorteados do seu *cache* de participantes. Esse procedimento tem duas finalidades. A primeira é auxiliar um nó a manter um número estável de parceiros, em virtude de saída ou falha de nós. A segunda é fazer com que um nó encontre parceiros de melhor qualidade. Para medir a qualidade de um parceiro, um nó define uma pontuação que varia de acordo com a quantidade de pedaços recebidos desse parceiro. Seja um nó i , parceiro de um nó j . A pontuação de j calculada por i é dada por $\max\{\bar{s}_{i,j}, \bar{s}_{j,i}\}$, onde $\bar{s}_{i,j}$ representa o número médio de pedaços que o nó i recebeu de j por unidade de tempo e $\bar{s}_{j,i}$ o número médio de pedaços que i enviou para j por unidade de tempo. Dessa forma, como um nó pode ser receptor e emissor, considera-se o valor máximo entre as duas direções. Intuitivamente, um parceiro com mais pedaços disponíveis e maior capacidade para enviar os pedaços terá uma maior pontuação

e, conseqüentemente, será um parceiro de maior qualidade. Sempre que um novo parceiro com maior pontuação é escolhido, o nó tem de retirar de sua lista o parceiro com a menor pontuação para manter o número de parceiros estável.

O número de parceiros influencia na sobrecarga de controle e na continuidade da recepção do vídeo. Quanto maior o número de parceiros, maior a sobrecarga de controle. Quanto maior é o número de parceiros, maior também é a continuidade do vídeo, pois cada nó possui mais opções de emissores. Entretanto, a partir de um determinado valor o ganho na continuidade é marginal. Nos experimentos com o CoolStreaming, esse valor para o número de parceiros é igual a quatro, considerando malhas de distribuição de diferentes tamanhos e que os participantes transmitem os pedaços de vídeo à mesma taxa [14].

Algoritmos de Escalonamento de Pedacos

O algoritmo de escalonamento de pedacos é fundamental para garantir os requisitos de banda passante e tempo da distribuição de vídeo. Assim como nos sistemas de compartilhamento de arquivos, os pedacos podem ser recebidos fora de ordem, mas devem ser recebidos antes do seu tempo de reprodução. Além disso, na difusão de vídeo, o progresso da reprodução do vídeo é fortemente sincronizado entre os participantes, pois os nós têm interesse em um dado conteúdo durante um mesmo período de tempo e também porque não podem controlar as ações de reprodução do vídeo. Os resultados dos experimentos realizados com o CoolStreaming mostram que o intervalo entre os trechos de um mesmo vídeo reproduzidos por diferentes participantes não é superior a um minuto [14]. Essa característica não é observada no compartilhamento de arquivos, uma vez que o interesse em um arquivo, geralmente, não está associado a um dado período de tempo, como ocorre, por exemplo, em transmissões de vídeo ao vivo. Portanto, cabe ao algoritmo de escalonamento determinar qual a maneira mais eficiente para um nó simultaneamente receber e encaminhar os pedacos de vídeo.

No CoolStreaming, um fluxo de vídeo é dividido em pedacos de tamanho uniforme. A disponibilidade desses pedacos no *buffer* de um nó é representada por um mapa de *buffer* (*Buffer Map* - BM). Como visto, os pedacos de vídeo só são úteis se forem recebidos antes do seu tempo de reprodução. Por isso, o mapa de *buffer*

é representado por uma janela deslizante, que é atualizada em função do tempo de reprodução, como mostra a Figura 3.5. Nesse exemplo, o *buffer* pode armazenar até 12 pedaços e o valor da janela deslizante é igual a 6 pedaços. Os retângulos hachurados indicam os pedaços disponíveis no *buffer*. Um bit assume o valor 1 se o pedaço correspondente está disponível. Do contrário, o bit assume o valor 0. No exemplo da Figura 3.5, o mapa de *buffer* codificado corresponde à palavra 101110.

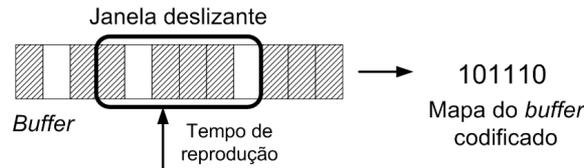


Figura 3.5: O mapa de *buffer* no sistema CoolStreaming.

Cada nó envia periodicamente o seu mapa de *buffer* para seus parceiros. Com isso, um nó conhece quais pedaços do vídeo seus parceiros possuem e, dessa forma, pode definir de qual parceiro e quando vai requisitar um dado pedaço. Essa é a função do escalonador de pedaços que deve ainda lidar com duas restrições: o tempo de reprodução de cada pedaço e a heterogeneidade dos receptores. Esse problema é uma variação do escalonamento de máquinas paralelas, que é um problema NP-completo [51]. Além disso, o algoritmo de escalonamento deve se adaptar rapidamente à dinâmica da malha de distribuição. Por isso, o escalonador usado no CoolStreaming define uma heurística simples e de resposta rápida às variações da malha.

A heurística de escalonamento de pedaços de vídeo funciona da seguinte forma. Inicialmente, um nó calcula o número de emissores potenciais para cada pedaço, isto é, os parceiros que possuem um dado pedaço armazenado em seus *buffers*. O algoritmo de escalonamento determina o emissor de cada pedaço de acordo com o número de emissores potenciais de cada pedaço e também com a capacidade de saída de cada possível emissor. Assume-se que um pedaço com menos emissores potenciais tem uma maior chance de ser recebido com atraso, podendo até mesmo chegar após o seu tempo de reprodução. Por isso, o algoritmo prioriza pedaços que possuam um menor número de emissores potenciais. Se houver mais de um possível emissor por pedaço, escolhe-se aquele com maior banda passante e que

possua disponibilidade para enviar o pedaço antes do seu tempo de reprodução. O algoritmo de escalonamento é executado periodicamente por cada nó e, dessa forma, se define a escala de pedaços a serem requisitados a um mesmo emissor. Uma vez definida a escala para um emissor, utiliza-se uma sequência de bits semelhante ao mapa de *buffer* para representá-la, na qual os bits em 1 representam os pedaços desejados. Cada escala é enviada para o seu respectivo emissor que, ao recebê-la, envia os pedaços requisitados ordenadamente.

Vale ressaltar que o nó origem é unicamente um emissor e sempre possui todos os pedaços do vídeo disponíveis. Para evitar a sobrecarga de requisições de seus parceiros, a origem implementa um algoritmo de escalonamento adaptativo. Se necessário, a origem difunde um mapa de *buffer* conservativo para reduzir a sua carga. A ideia é enviar para determinados parceiros um mapa que não contém todos os pedaços disponíveis, ou seja, com alguns bits zerados. Dessa forma, os parceiros que receberem esse mapa “incompleto” irão requisitar os pedaços “não-disponíveis” de outros parceiros e não mais da origem.

3.4 Sistemas de Vídeo sob Demanda

O desenvolvimento de sistemas par-a-par de vídeo sob demanda é recente [1, 12, 13, 2]. A maioria desses sistemas se baseia na arquitetura em malha. Assim, como na difusão, um fluxo de vídeo é dividido em pedaços. Entretanto, no vídeo sob demanda, cada participante pode estar assistindo a um dado trecho do vídeo, uma vez que é possível o usuário controlar a sua reprodução, pausando, retrocedendo e avançando. Dessa forma, o número de participantes que simultaneamente se interessa por um mesmo trecho de um vídeo tende a ser menor por causa da dessincronização da reprodução dos participantes e, conseqüentemente, o número de pares dos quais se podem solicitar pedaços desse trecho também é menor. Para aumentar a interseção de interesses entre os pares, os sistemas par-a-par de vídeo sob demanda empregam diferentes mecanismos. A seguir, são apresentados dois desses sistemas: o BulletMedia [12] e o PPLive [2].

3.4.1 BulletMedia

O BulletMedia é um sistema par-a-par de vídeo sob demanda que implementa as funções de avanço e retrocesso do vídeo [12]. A ideia básica desse sistema, baseado na arquitetura em malha, é que todo pedaço de um vídeo deve ser replicado pelos participantes mesmo que ele não seja necessário para o ponto de reprodução atual dos participantes ativos do sistema. Dessa forma, quando possuem recursos ociosos, os participantes solicitam e armazenam pedaços de vídeo que não são usados para a reprodução imediata e que talvez nem sejam usados por eles. Esse procedimento é chamado de armazenamento proativo de pedaços (*prefetching*). Ao aumentar a disponibilidade de pedaços no sistema, reduz-se o número de participantes que requisitam pedaços diretamente à fonte e também o tempo de espera para retomar a reprodução do vídeo após uma interação.

Os dois pontos-chaves do BulletMedia são o controle de quais pedaços do vídeo devem ser replicados e a descoberta eficiente dos pedaços armazenados pelos participantes. Para tanto, o BulletMedia constrói duas redes sobrepostas: uma rede em malha e uma rede estruturada baseada em tabelas *hash* distribuídas (*Distributed Hash Table - DHT*)². A rede sobreposta em malha é usada para encaminhar os pedaços necessários para o andamento normal da reprodução do vídeo. A rede sobreposta estruturada é usada para controlar a replicação dos pedaços. Ela também é usada pelos participantes para descobrirem a localização dos pedaços disponíveis em outros participantes e, conseqüentemente, afetar menos a continuidade de recepção após um avanço ou retrocesso.

Os procedimentos de entrada de participantes no sistema, de construção da rede em malha e de escalonamento de pedaços são similares aos do CoolStreaming, descritos na Seção 3.3.1. Além disso, no BulletMedia, cada participante possui um *buffer* de reprodução e um *cache* de conteúdo, ambos de tamanho fixo. O *buffer* de reprodução é usado para armazenar os pedaços que estão na iminência de serem requisitados pelo codificador de vídeo. O *cache* de conteúdo, por sua vez, armazena

²Diz-se que uma rede sobreposta é estruturada quando existe uma relação direta entre a sua topologia e o conteúdo armazenado nos nós da rede. Em virtude dessa relação, a localização de conteúdo em redes estruturadas é mais eficiente do que nas redes não-estruturadas, nas quais são necessários mecanismos adicionais para localizar o conteúdo desejado.

os pedaços já assistidos pelo participante, os que estão no *buffer* de reprodução e os que ainda serão encaminhados para esse *buffer*. O mapa de *buffer* do BulletMedia é a representação do *cache* de conteúdo. Assume-se que com uso do *cache* de conteúdo, ao longo do tempo, haverá sobreposições dos pontos de reprodução dos participantes, ou seja, eles estarão interessados nos mesmos pedaços do vídeo durante um mesmo intervalo de tempo. Isso nem sempre é eficiente, pois, como será visto adiante no Capítulo 5, os parceiros dos participantes são escolhidos de forma aleatória. Conseqüentemente, nem sempre os parceiros selecionados possuem pedaços de interesse disponíveis.

A rede estruturada baseada em DHT é construída para reduzir a dependência do sistema em relação à fonte. Para que a reprodução do vídeo seja iniciada, um número pré-estabelecido de pedaços deve estar armazenado no *buffer* de reprodução. Por isso, ao mudar o ponto de reprodução atual do vídeo, um participante deve descobrir e solicitar os pedaços que são necessários para recomençar a reprodução a partir desse novo ponto. Tais pedaços podem estar no *cache* de conteúdo do próprio participante ou nos dos seus parceiros. Do contrário, o participante deve solicitar os pedaços diretamente à fonte ou encontrar outro parceiro que possua tais pedaços. A DHT é usada justamente para acelerar o processo de busca de parceiros que possuem os pedaços necessários para o novo ponto de reprodução e, assim, reduzir a sobrecarga da fonte. A DHT armazena metadados sobre a localização dos pedaços nos participantes do sistema. Dessa forma, quando um participante altera seu ponto de reprodução, ele primeiro checa a disponibilidade dos pedaços de vídeo nos seus parceiros. Se eles não possuem os pedaços desejados, o participante envia uma requisição à DHT com os identificadores desses pedaços. A DHT, então, retorna para o participante um conjunto de nós associados aos pedaços desejados. O participante, por sua vez, escolhe aleatoriamente um nó desse conjunto e o adiciona ao seu conjunto de parceiros. Esse procedimento faz com que os participantes descubram novos parceiros mais facilmente e reduzam a dependência da fonte.

Um protótipo do BulletMedia foi implementado por Vratonjic *et al.* [12] e os resultados experimentais mostram a eficiência do mecanismo de armazenamento pró-ativo de pedaços. Um dado vídeo, cujo tamanho é 50 MB, é totalmente replicado na rede em cerca de 150 s mesmo com os usuários alterando o seu ponto de reprodução.

A capacidade de *upload* e *download* dos participantes é igual a 2 Mb/s. Os autores concluem, então, que o BulletMedia controla efetivamente a replicação de pedaços e é robusto às interações dos participantes.

3.4.2 PPLive

Atualmente, o PPLive é o sistema par-a-par mais popular para a difusão de vídeo [5]. Em janeiro de 2006, o PPLive atingiu a marca de mais de 200 mil usuários durante a distribuição de um programa popular da TV estatal chinesa. As taxas de envio variaram entre 400 e 800 kb/s, resultando em uma taxa agregada de 100 Gb/s [5].

O PPLive é composto, basicamente, por um conjunto de servidores, que são as fontes de vídeo; um conjunto de rastreadores (*trackers*), que auxiliam os participantes na busca por outros participantes interessados no mesmo vídeo; um servidor de inicialização (*bootstrap server*), para auxiliar os participantes na busca por um rastreador; e, o mais importante, o conjunto de participantes.

Recentemente, foi lançada a versão do PPLive com suporte ao vídeo sob demanda [2]. A alternativa usada pelo PPLive para reduzir os efeitos da dessincronização da reprodução entre os participantes é o uso de um *cache* de pedaços de vídeo, além do *buffer* de reprodução. Cada participante contribui com uma quantidade predeterminada e fixa de espaço em disco para servir de *cache* dos pedaços de vídeo recentemente recebidos. Assim, participantes que não estejam necessariamente assistindo à mesma parte do vídeo podem ter pedaços para trocar, uma vez que os pedaços já recebidos estão armazenados no *cache*. O PPLive, porém, adota uma estratégia de replicação de pedaços diferente da do BulletMedia. No PPLive, não há armazenamento pró-ativo, ou seja, um nó solicita somente os pedaços que são necessários para o seu ponto atual de reprodução. No entanto, quando há espaço disponível no *cache*, os participantes armazenam os pedaços de outros vídeos que já assistiu, além dos pedaços do vídeo que está assistindo. Essa abordagem é chamada de *cache* de múltiplos vídeos (*Multiple Video Cache - MVC*). Mostra-se que o MVC aumenta a escalabilidade dos sistemas par-a-par de vídeo sob demanda, pois reduz ainda mais a sobrecarga da fonte se comparado aos sistemas que armazenam somente os pedaços do vídeo em reprodução. Por outro lado, os sistemas com o

MVC sofrem mais com a saída dos pares, uma vez que esses pares podem participar de mais de um enxame com diferentes popularidades [52].

Para adicionar ou remover pedaços, ou até mesmo um vídeo inteiro, do *cache*, atribui-se um peso a cada vídeo. Dois fatores determinam esse peso: o número de pedaços desse vídeo que estão armazenados localmente e a necessidade do vídeo. A necessidade é dada pela taxa de disponibilidade por demanda (*Availability To Demand ratio* - ATD). Supondo que n participantes estão assistindo ao vídeo e que esse vídeo é armazenado em *cache* por c participantes, incluindo os que o assistem no momento, a taxa ATD é dada por c/n . Sendo assim, a necessidade de um vídeo diminui à medida que a sua ATD aumenta. No PPLive, define-se um limiar para a ATD de cada vídeo, calculado de acordo com a banda de subida (*upload*) disponibilizada pelos participantes e a taxa de transmissão da fonte [2]. Os rastreadores são os responsáveis pelo cálculo da ATD de cada vídeo.

No PPLive, a seleção dos parceiros é feita da seguinte forma. Quando um participante deseja iniciar a reprodução de um vídeo, ele entra em contato com um rastreador. O rastreador, então, o envia uma lista de participantes que armazenam os pedaços do vídeo desejado. Assim, um participante seleciona de forma aleatória seus parceiros entre os candidatos enviados pelo rastreador. Logo após iniciar a reprodução do vídeo, o participante informa ao rastreador que está replicando esse vídeo e, conseqüentemente, armazenando seus pedaços. Os parceiros trocam mapas de pedaços (*chunk bitmaps*) entre si para difundir a disponibilidade dos pedaços. Dessa forma, determina-se de qual parceiro deve se solicitar um pedaço desejado.

O PPLive define ainda mecanismos de escalonamento e de transmissão de pedaços, mecanismos de autenticação de pedaços para evitar ataques de poluição e mecanismos para lidar com dispositivos NAT (*Network Address Translation*) e *firewalls* [2], que geralmente prejudicam o desempenho dos sistemas par-a-par, como é visto no Capítulo 4.

Capítulo 4

Avaliação Experimental de Sistemas Par-a-Par de Vídeo em Redes Sem Fio

Um dos pontos mais críticos da distribuição de vídeo em redes móveis é o acesso sem fio de última milha. Existem diversas tecnologias de rede empregadas para tal finalidade. Cada uma possui diferentes características e requisitos, exigindo assim que os aplicativos de distribuição de vídeo se adaptem às redes sem fio. As redes em malha sem fio são uma solução promissora para prover ubiquidade de acesso à Internet com suporte à mobilidade dos usuários [53]. Neste capítulo, avalia-se experimentalmente o desempenho de alguns dos sistemas par-a-par de difusão de vídeo mais populares em uma rede em malha sem fio. Primeiramente, alguns trabalhos relacionados à avaliação experimental de sistemas par-a-par são discutidos. Em seguida, são identificados aspectos práticos relacionados à mobilidade dos usuários que afetam o desempenho dos sistemas par-a-par. Os sistemas par-a-par usados e o cenário dos experimentos são brevemente descritos. Por fim, os resultados dos experimentos são apresentados. Compara-se o desempenho dos sistemas par-a-par com um aplicativo cliente-servidor.

4.1 Trabalhos Relacionados

Existem diversos trabalhos sobre avaliações experimentais de sistema par-a-par de distribuição de vídeo [54, 55, 5, 10]. Entretanto, nenhum trata especificamente do desempenho desses sistemas em redes sem fio considerando a presença de usuários móveis.

Sentinelli *et al.* [5] avaliam o desempenho do SopCast, um dos sistemas par-a-par mais populares atualmente, em dois diferentes cenários. No primeiro, avalia-se tal sistema durante sua operação normal, ou seja, roda-se o aplicativo e seleciona-se um canal qualquer disponibilizado. Nesse cenário, determina-se o número de parceiros de cada participante, o atraso de inicialização dos canais, o tamanho do *buffer* de reprodução e o índice de continuidade de reprodução¹. Para um dado canal selecionado, conclui-se que cada participante possui de 2 a 5 parceiros e que o atraso de inicialização é da ordem de dezenas de segundos dependendo da rede de acesso do participante. Nota-se também que o tamanho do *buffer* de reprodução é da ordem de 2,0 MB e que o índice de continuidade é de aproximadamente 99% para um vídeo com taxa de transmissão de 380 kb/s. No segundo cenário, avalia-se o SopCast usando o PlanetLab, onde a fonte e todos os participantes que assistem ao seu conteúdo são controláveis. Mede-se o tráfego de subida (*upload*) e descida (*download*) e o atraso de inicialização. Mostra-se que as quantidades de tráfego de subida e de descida são aproximadamente iguais e que o atraso de reprodução nesse cenário também é da ordem de dezenas de segundos. Ao analisarem os resultados, os autores dizem que o maior obstáculo ao desenvolvimento dos sistemas par-a-par de vídeo na Internet é a conexão assimétrica característica da tecnologia ADSL usada, atualmente, pela maioria dos usuários residenciais.

Silverston e Formaux [10] medem o tráfego de rede gerado por quatro sistemas de vídeo par-a-par populares. As medições foram realizadas durante a copa do mundo de futebol em 2006. Os experimentos consistem da captura do tráfego gerado por participantes do sistema localizados em uma rede Ethernet e em uma rede ADSL que assistiam aos jogos de futebol usando um dos aplicativos. Os resultados desses experimentos mostram que cada um dos sistemas gera um padrão de tráfego dife-

¹O índice de continuidade é definido como a razão entre o número de pedaços de vídeo recebidos antes do tempo limite para a sua reprodução e o número total de pedaços de vídeo.

rente. Um resultado interessante mostra que a quantidade de tráfego enviada por um participante é maior do que a recebida. Para os quatro sistemas, o participante enviou aproximadamente 80% de todo o tráfego analisado e recebeu apenas 20%. É possível ainda identificar os tipos de protocolo de transporte usados por cada sistema. Essa é uma característica fundamental para avaliar os sistemas de vídeo par-a-par em redes móveis, como é visto adiante.

Alguns trabalhos avaliam a distribuição de vídeo em redes sem fio através de modelos analíticos, simulações e experimentos [56, 57, 58], mas não consideram o uso de sistemas par-a-par. Shihab *et al.* [57] comparam o desempenho da distribuição de vídeo em tecnologias de rede sem fio e sem novos fios usadas em redes domiciliares [59] através de simulações. A partir dessa análise, os autores propõem três arquiteturas de rede diferentes para atender os requisitos de custo, confiabilidade e escalabilidade da distribuição de vídeo.

4.2 Redes em Malha Sem Fio

As redes em malha sem fio (*Wireless Mesh Networks* - WMNs) [53] são uma alternativa de baixo custo para as redes de acesso. O objetivo dessas redes é aumentar o alcance da comunicação mantendo a conectividade da rede apesar da dinamicidade do meio e da mobilidade dos usuários. A principal característica das redes em malha sem fio é o uso de um *backbone* composto por roteadores sem fio tipicamente estacionários. A função desses roteadores é prover acesso a redes cabeadas de banda larga para nós que não estejam no alcance dos *gateways* para a Internet e interconectar redes locais isoladas. Os usuários usam os roteadores como pontos de acesso sem fio. Os roteadores do *backbone*, por sua vez, se comunicam através de múltiplos saltos para manter a conectividade da rede e para encaminhar o tráfego dos e para os usuários. Por isso, diz-se que as redes em malha sem fio são um híbrido entre os modos de comunicação *ad hoc* e infra-estruturado. Atualmente, as redes em malha sem fio estão sendo implantadas em cidades, *campi* universitários e em redes comunitárias [60].

Nas redes em malha sem fio, os usuários são potencialmente móveis e, como dito anteriormente, se conectam ao *backbone* através dos roteadores que atuam como

pontos de acesso. Durante o *handoff* que ocorre a cada mudança de ponto de acesso, a conectividade da rede pode ser interrompida e o desempenho dos protocolos degradados [61]. A duração do *handoff* depende da forma de endereçamento empregada pela rede, como discutido a seguir.

4.2.1 Endereçamento

O endereçamento é um grande desafio para as redes móveis. As mudanças e o uso de endereços válidos ou inválidos, também chamados de “públicos” ou “privados”, influenciam o desempenho dos protocolos, podendo provocar perdas de conectividade e a degradação total de aplicações.

Endereços IP válidos e inválidos

De acordo com dados coletados em 2005, mais de 50% dos usuários de banda larga do Reino Unido e dos Estados Unidos estão conectados à Internet com endereços inválidos, através de dispositivos que fazem tradução de endereços (*Network Address Translation* - NAT) [11]. Usando NAT, vários computadores localizados em um domicílio ou escritório podem compartilhar o acesso à Internet utilizando um único endereço IP válido. O dispositivo NAT, por exemplo, um roteador com uma conexão ADSL e portas Ethernet, “traduz” os endereços dos pacotes de saída, através da substituição do endereço de origem (um endereço IP inválido não-roteável) e do número de porta escritos pela estação pelos endereço IP e número de porta do dispositivo NAT (o roteador). Esse procedimento é chamado de mapeamento de portas.

O principal problema relacionado ao uso do NAT é que usuários conectados à Internet através de um dispositivo NAT podem se conectar a outros usuários com endereço IP válido, mas em princípio não podem receber conexões, uma vez que o seu endereço inválido não é conhecido pelos demais usuários da Internet. Por essa razão esse endereço inválido e, portanto, não-roteável, é também chamado de privado. Além disso, quando um usuário que está atrás de um dispositivo NAT se move para um novo local que é servido por outro dispositivo NAT, as conexões TCP são quebradas, devido à mudança de endereço IP e ao tempo exigido pela reconfiguração do mapeamento de portas. Conseqüentemente, as aplicações podem

sofrer com perdas de segmentos e desconexões.

Nos sistemas par-a-par, os participantes precisam ser bidirecionalmente alcançáveis. Como frequentemente o NAT impede participantes de estabelecerem conexões entre si, o desempenho desses sistemas é afetado. Pelo mesmo motivo, o uso de *firewalls* também prejudica o desempenho desses sistemas. Atualmente, a maioria dos sistemas par-a-par depende de configuração manual das portas para que seja possível a comunicação entre pares conectados através de dispositivos NAT. No entanto, algumas técnicas foram propostas para fazer com que sistemas par-a-par possam funcionar através de dispositivos NAT [62, 63, 64]. Entretanto, nenhuma dessas técnicas é ideal para um cenário de mobilidade.

Endereços IP dinâmicos e fixos

As aplicações na Internet assumem que o endereço IP de um dado nó não muda durante longos períodos de tempo. Essa suposição, entretanto, não é válida para redes móveis, pois os usuários podem mudar de endereços à medida que se movem de uma rede para outra. Sendo assim, conexões estabelecidas não sobrevivem às mudanças de endereço, pois assumem endereços IP de origem e destino fixos. Caso o usuário móvel use um IP dinâmico, a aplicação pode experimentar perdas de conectividade durante o processo de aquisição de novos endereços IP. Por outro lado, se o usuário móvel possui um endereço IP fixo, seu endereço deve ser mantido através de túneis para dar suporte às conexões ativas usando IP móvel, por exemplo.

Em redes em malha sem fio, uma alternativa ao tunelamento é executar um protocolo de roteamento *ad hoc* em cada usuário móvel. Tais redes usam protocolos de roteamento para encaminhar dados. Cada nó mantém uma tabela de roteamento para encaminhar corretamente os pacotes. Se um usuário, que também está rodando um protocolo de roteamento, se move pelo *backbone*, a topologia da rede muda e, assim, a tabela de roteamento deve ser atualizada. Se o usuário possui um endereço IP fixo, o *handoff* é influenciado apenas pelo tempo de convergência do protocolo de roteamento da rede sem fio. Por outro lado, se o usuário móvel usa um endereço IP dinâmico, o tempo para se adquirir um novo endereço IP ou para se criar um túnel deve ser acrescentado ao tempo de convergência.

4.3 Sistemas Par-a-Par de Vídeo Analisados

Foram escolhidos quatro dos sistemas par-a-par de vídeo mais populares. São eles o PPLive, o SopCast, o PPStream e o TVAnts². Todos são baseados na arquitetura em malha, implementam a difusão de vídeo e são proprietários, ou seja, não possuem especificação e o código-fonte não está disponível. Atualmente, o PPLive é o sistema mais popular, entretanto, o SopCast a cada dia tem mais usuários [5]. A razão dessa popularidade é que o SopCast permite que qualquer usuário registre e distribua o seu próprio canal de TV na Internet. Com isso qualquer um pode se tornar um produtor de conteúdo.

A escolha do protocolo de transporte é fundamental para o desempenho dos sistemas de vídeo par-a-par. Dependendo da escolha, um sistema pode ser mais eficiente na entrega de conteúdo de vídeo, porém a comunicação e a cooperação entre os participantes podem ser afetadas. Além disso, a escolha afeta a adaptação dos sistemas à mobilidade dos usuários. Os sistemas par-a-par de vídeo se dividem entre o uso do UDP e do TCP para o envio de mensagens de controle e comunicação entre os participantes [10]. O processamento requerido pelo UDP é menor do que o do TCP, uma vez que o UDP não estabelece conexão e nem implementa mecanismos de retransmissão e de controle de congestionamento e de fluxo. Por isso, o UDP é o protocolo de transporte mais indicado para aplicações multimídias. No entanto, por não possuir controle de fluxo, a maioria dos dispositivos de *firewall* e NAT bloqueia o tráfego UDP para evitar possíveis congestionamentos na rede e a sobrecarga de processamento nos roteadores. Por isso, muitos sistemas de vídeo par-a-par usam o TCP como protocolo de transporte para atravessarem dispositivos de *firewall* e NAT. Ao usar o TCP para distribuição de vídeo, um sistema pode experimentar variações na taxa efetiva do vídeo recebido por seus participantes em virtude da dinâmica do mecanismo de controle de congestionamento do TCP. Em função da arquitetura fechada da maioria dos sistemas, medições foram realizadas de modo a identificar os protocolos de transporte utilizados tanto para o tráfego de vídeo quanto o tráfego de controle e confirmaram os resultados encontrados na literatura [5, 10]. O PPLive usa o TCP para a transmissão do vídeo e datagramas UDP para o controle. O SopCast

²Esses sistemas estão disponíveis, respectivamente, em <http://www.pplive.com/en>, <http://www.sopcast.org>, <http://www.ppstream.cn> e <http://tvants.en.softonic.com>.

usa essencialmente o UDP tanto para a transmissão do vídeo quanto para o controle. Somente algumas poucas conexões TCP são observadas na inicialização do sistema. O PPStream usa apenas o TCP, enquanto o TVAnts usa tanto o TCP quanto o UDP para transmissão e sinalização. As principais características dos aplicativos analisados estão na Tabela 4.1.

Tabela 4.1: Principais características dos sistemas par-a-par analisados.

| Sistemas | Estrutura de distribuição | Protocolo de Transporte | Código-fonte | Canal próprio |
|-----------------|---------------------------|---------------------------|--------------|---------------|
| SopCast | malha | UDP/poucos segmentos TCP | proprietário | sim |
| PPLive | malha | TCP/poucos datagramas UDP | proprietário | não |
| TVAnts | malha | TCP/UDP | proprietário | não |
| PPStream | malha | TCP | proprietário | não |

Espera-se que os sistemas fortemente baseados no uso do TCP enfrentem problemas no cenário de mobilidade dos experimentos, descrito a seguir. A natureza orientada à conexão do TCP em conjunto com seu mecanismo de controle de congestionamento torna a adaptação desse protocolo não-trivial para redes sem fio [65]. Porém, a natureza da arquitetura par-a-par para a distribuição de vídeo é baseada em um grande número de conexões TCP simultâneas e de curta duração. Essa característica facilita a adaptação dos sistemas baseados em TCP, como é visto a seguir.

4.4 Resultados

Nesta seção, os resultados dos experimentos são apresentados. Primeiramente, o desempenho do aplicativo cliente servidor é avaliado enquanto o cliente se move. Nesse mesmo cenário, o comportamento dos diferentes sistemas par-a-par de vídeo é analisado. As medidas foram feitas em uma rede de testes *indoor* localizada no laboratório do Grupo de Teleinformática e Automação (GTA) da COPPE/UFRJ. A rede de testes é composta de dois computadores pessoais, um computador portátil e cinco roteadores sem fio. Os computadores pessoais estão equipados com interfaces sem fio PCI IEEE 802.11b/g, antenas omnidirecionais e interfaces Fast Ethernet. Os

dois computadores pessoais são os *gateways* da rede e provêm acesso à Internet para o cliente móvel. Esses computadores rodam o sistema operacional Debian Linux e o computador portátil roda o Windows XP, pois a maioria dos sistemas par-a-par de vídeo não são implementados para o Linux. Para os experimentos com o aplicativo cliente-servidor, outro computador pessoal, rodando o Windows Vista e conectado por uma rede cabeada aos *gateways* da rede sem fio, é usado como servidor de vídeo. Os roteadores sem fio são da marca/modelo LinkSys WRT54G e rodam o sistema operacional Linux OpenWRT. Esses roteadores e os dois *gateways* formam o *backbone* da rede em malha.

Os passos do experimento são os seguintes. Em primeiro lugar, executam-se farejadores de pacotes (*packet sniffers*) no computador portátil e nos computadores pessoais. Em seguida, inicia-se um dos sistemas par-a-par e seleciona-se um canal ou inicia-se o aplicativo cliente servidor. Para avaliar o impacto da mobilidade do usuário, a análise é limitada aos eventos que ocorrem dentro da rede em malha sem fio. Para tanto, compara-se o tráfego que chega aos *gateways* com o que chega ao computador portátil. Assim que o *buffer* de reprodução do sistema par-a-par esteja cheio, essa fase de inicialização se encerra. Logo em seguida, o usuário carregando o computador portátil começa a andar do ponto A ao ponto D e retorna ao ponto A, como mostra a Figura 4.1. Esse cenário de mobilidade busca reproduzir o comportamento de um usuário andando através de diferentes *gateways*. Espera-se que nesse cenário, o usuário mude de *gateway* pelo menos duas vezes, do GW_1 para o GW_2 e, então, novamente para GW_1 no caminho de volta. Assim que o usuário retorna ao ponto A, chega-se à fase de encerramento, na qual o aplicativo de vídeo e os farejadores são encerrados.

Nos experimentos, todos os nós da rede de testes usam o padrão IEEE 802.11g no modo *ad hoc* e rodam o protocolo OLSR (*Optimized Link-State Routing*) configurado com a métrica de roteamento ETX (*Expected Transmission Count*). Os *gateways* anunciam rotas-padrão usando mensagens HNA (*Host and Network Association*) implementadas pelo OLSR. Dessa forma, o usuário escolhe o *gateway* que utiliza automaticamente, de acordo com o caminho computado pelo protocolo de roteamento. Na rede de testes, o computador portátil usa um endereço IP privado. Por isso, os *gateways* usam NAT para permitir que usuários acessem a Internet.

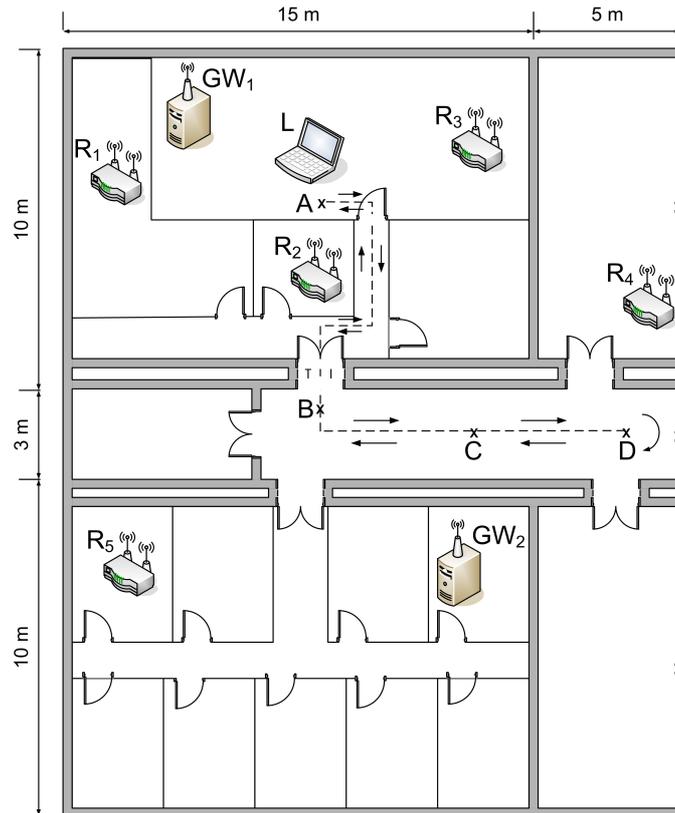


Figura 4.1: A trajetória do experimento.

O endereço IP do computador portátil é fixo e, por isso, não são considerados, na análise, possíveis atrasos provocados pelo DHCP.

O servidor de vídeo usado no experimento da arquitetura cliente-servidor é o aplicativo VLC. A requisição do vídeo é feita usando o HTTP (*HyperText Transfer Protocol*). O usuário móvel também usa o VLC para receber o vídeo. O vídeo servido foi previamente gravado de um canal do SopCast, um dos sistemas par-a-par analisados, para garantir que os vídeos usados nos experimentos tenham taxas de transmissão similares.

Foram usados quatro sistemas par-a-par de difusão de vídeo nos experimentos: o SopCast (versão 2.0.4), o PPStream (versão 2.2.25), o PPLive (versão 1.9.21) e o TVAnts (versão 1.0.0.59). Para cada sistema, foram feitas cinco rodadas de medidas usando canais populares: a TV Macumb@ para o SopCast, o Hunan TV para o PPStream e o PPLive e o CCTV-1 para o TVAnts. Esses canais possuem taxas de transmissão entre 350 e 400 kb/s.

Além das métricas de desempenho da rede, é feita uma análise subjetiva para

verificar a qualidade do vídeo recebido pelo usuário móvel. Primeiramente, o fluxo de vídeo recebido pelo usuário é gravado. Esse vídeo é chamado de amostra do usuário. O fluxo de vídeo recebido por um outro usuário localizado na mesma rede cabeada dos dois *gateways* também é gravado. Esse vídeo é chamado de amostra de referência. As amostras do usuário e de referência são mostradas para 30 espectadores. Cada espectador atribui uma nota de 1 a 5 para cada amostra de acordo com a qualidade do vídeo percebida. A nota 5 é a melhor.

Como mencionado anteriormente, de acordo com o protocolo de transporte usado, os sistemas se comportam de forma diferente na presença de usuários móveis e de *gateways* usando NAT. A seguir, são apresentados os resultados para o StpCast e o PPStream, que representam, respectivamente, os sistemas que usam UDP e TCP. Os resultados obtidos para todos os sistemas estão resumidos na Tabela 4.2 da Seção 4.4.4. São analisados os tráfegos de descida e de subida.

4.4.1 Aplicativo Cliente-Servidor

A Figura 4.2 mostra o desempenho do aplicativo cliente-servidor quando o cliente está parado ou se movendo. Os pontos de A a D na Figura 4.2(b) mostram os instantes nos quais o usuário passa pelos marcos da trajetória da Figura 4.1. Os pontos pretos localizados sobre o eixo X indicam os instantes de mudança de *gateway*. Nos dois cenários, o cliente estabelece somente uma conexão TCP com o servidor. Quando o cliente está parado, essa conexão permanece ativa durante todo o experimento, como mostra a Figura 4.2(a). Por outro lado, quando o cliente está em movimento, o número de segmentos TCP recebidos diminui rapidamente logo após a primeira mudança de *gateway*. Após aproximadamente 20 s, a conexão TCP é encerrada, como indicado na Figura 4.2(b), e a recepção do vídeo cessa. Para esse caso, os espectadores deram nota 3,68 (com desvio padrão igual a $\sigma = 0.75$) para a amostra de referência e 1,35 ($\sigma = 0.55$) para a amostra do usuário. Isso mostra que o aplicativo cliente-servidor não se adapta ao cenário de mobilidade. Ao confiar em apenas uma conexão, a recepção do vídeo é interrompida, pois o servidor encerra a conexão. Por outro lado, um sistema par-a-par estabelece continuamente novas conexões de curta duração e mantém múltiplas conexões ativas e, por isso, é mais adaptado ao cenário de mobilidade, como demonstra a análise das próximas seções.

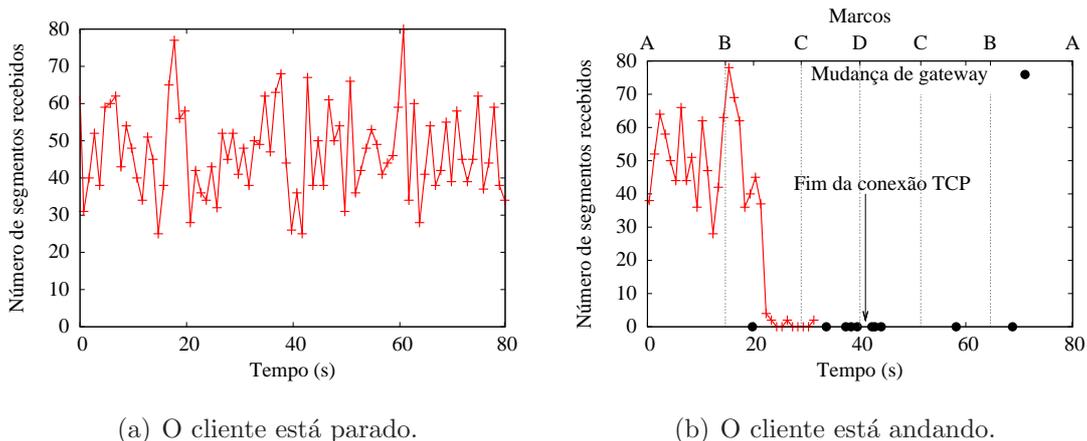
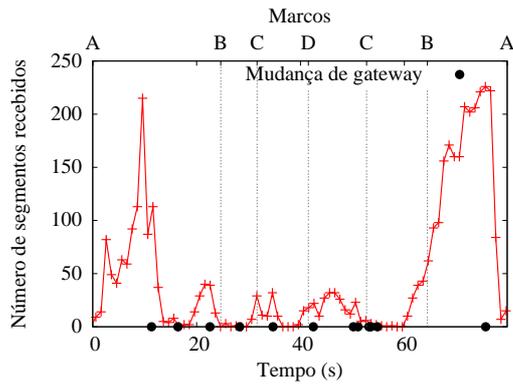


Figura 4.2: Medições com o aplicativo cliente-servidor.

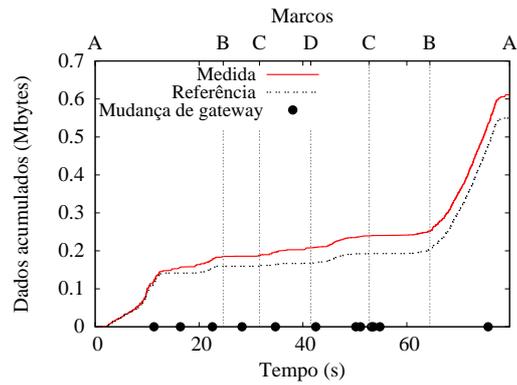
4.4.2 Sistema Par-a-Par Baseado no TCP: PPStream

A Figura 4.3 apresenta os resultados de uma rodada de medição com o PPStream. Tal sistema utiliza apenas o TCP como protocolo de transporte. Todas as figuras, exceto a Figura 4.3(d), mostram o desempenho do sistema durante a movimentação do cliente do ponto A para D e o retorno até A. O PPStream é o sistema par-a-par mais sensível às mudanças de *gateway*. A Figura 4.3(a) mostra o número de segmentos TCP recebidos por segundo durante o experimento. Observa-se que existe um período no qual o protocolo de roteamento está se adaptando ao movimento do cliente. À medida que as rotas são recomputadas, ocorrem mais de duas mudanças de *gateways*, apesar do cliente se movimentar em um cenário com apenas dois *gateways*. Nota-se também que o número de segmentos TCP recebidos diminui abruptamente e em alguns momentos é igual a zero. A Figura 4.3(b) mostra o número cumulativo de bytes referentes aos segmentos TCP recebidos pelo cliente enquanto se move através dos *gateways* da rede de testes. A curva de referência representa o número cumulativo de bytes transmitidos pelos *gateways* sem considerar as retransmissões. Por sua vez, a curva medida representa o número de bytes recebidos pelo cliente considerando as retransmissões e, por isso, está acima da curva de referência. Dessa forma, quanto maior o número de falhas de transmissão na rede sem fio, maior a distância entre as curvas. Os platôs ocorrem quando as conexões TCP estão estagnadas. A Figura 4.3(c) ilustra o início (cruzes) e o fim (triângulos) de cada conexão TCP. Como as fases de inicialização e encerramento são omitidas nessa figura, algumas conexões já estão abertas antes do início da movimentação

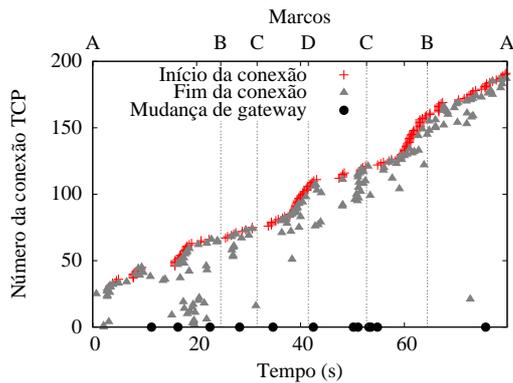
do cliente e outras não se encerram antes do término da movimentação. O fato da maioria das conexões TCP serem de curta duração contribui para a maior robustez dos sistemas par-a-par, nos quais os participantes podem entrar e sair do sistema a qualquer instante, em relação aos aplicativos cliente-servidor. Esse comportamento é melhor ilustrado pela Figura 4.3(d), que mostra a função distribuição cumulativa (*Cumulative Distribution Function* - CDF) da duração das conexões TCP incluindo as fases de inicialização e encerramento. Mostra-se que cerca de 90% das conexões duram menos do que 30 segundos. O uso de conexões de curta duração também ajuda os sistemas par-a-par a lidarem com a mobilidade dos usuários. Os sistemas par-a-par armazenam o conteúdo de vídeo recebido em um *buffer* e só reproduzem o vídeo quando esse *buffer* está em um dado nível pré-estabelecido. O conteúdo de vídeo armazenado no *buffer* vem de inúmeras conexões. Porém, apenas poucas dessas conexões estão ativas durante um período de tempo. Conseqüentemente, poucas conexões são afetadas pela mudança de *gateway*, e, por isso, a recepção do vídeo não é severamente afetada. No caso dos experimentos realizados, o intervalo em que há diminuição abrupta do número de segmentos recebidos não é suficiente para reduzir o nível do *buffer* a um ponto que afete significativamente a recepção do vídeo. A audiência, nesse caso, classificou a amostra de referência com nota 4,08 ($\sigma = 0,68$) e a amostra do usuário com nota 2,45 ($\sigma = 0,77$). É importante notar também que durante as mudanças de *gateway*, a taxa de abertura de novas conexões diminui. A Figura 4.3(e) mostra o intervalo de tempo entre os instantes de recepção dos segmentos TCP recebidos pelo cliente. Nota-se que durante as mudanças de *gateway*, o número de segmentos recebidos diminui e o tempo entre recepções sucessivas aumenta. Nesse caso, além das mudanças de *gateway*, o mecanismo de controle de congestionamento do TCP também contribui para reduzir a taxa de transmissão das fontes, uma vez que podem ocorrer estouros dos temporizadores e perdas de reconhecimentos positivos (ACKs) em virtude de eventuais perdas de conectividade causadas pela adaptação do protocolo de roteamento. Como os participantes também enviam pedaços do vídeo para seus parceiros, analisa-se também o tráfego de subida. A Figura 4.3(f) mostra que o tráfego TCP também sofre com falhas de transmissão na direção de subida. Isso é provocado principalmente pelo uso do NAT, como será visto a seguir.



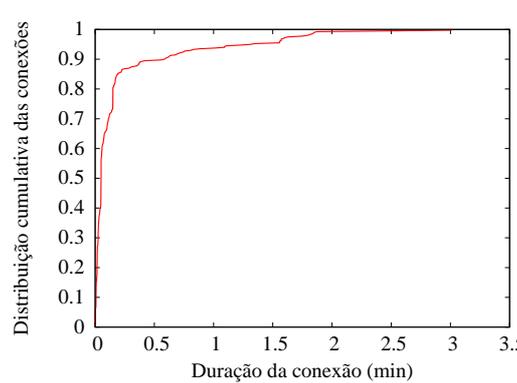
(a) Segmentos TCP recebidos.



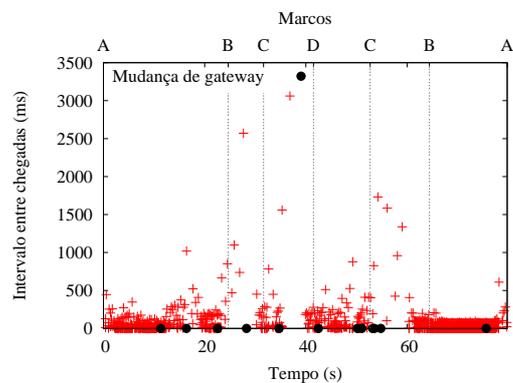
(b) Número cumulativo de bytes TCP recebidos.



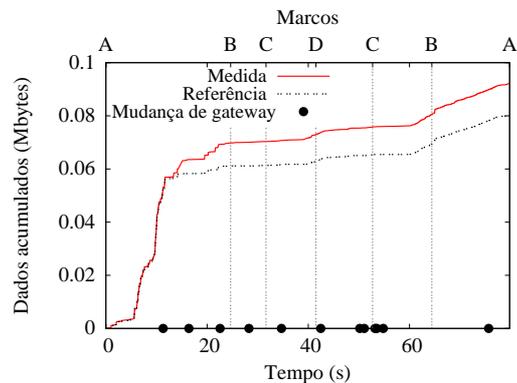
(c) Duração das conexões TCP.



(d) CDF da duração das conexões TCP.



(e) Intervalo entre os segmentos TCP recebidos.



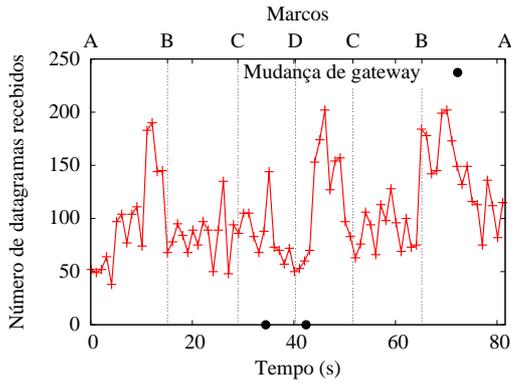
(f) Número cumulativo de bytes TCP enviados.

Figura 4.3: Medições com o PPStream.

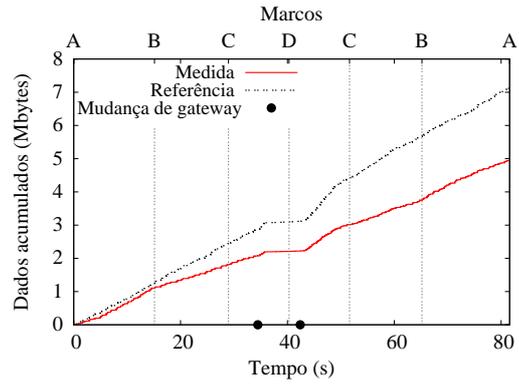
4.4.3 Sistema Par-a-Par Baseado no UDP: SopCast

A Figura 4.4 apresenta os resultados para uma rodada de medição com o SopCast, cuja maioria do tráfego é composta de datagramas UDP. Em geral, os sistemas que usam o UDP apresentaram menos perdas em virtude da mudança de *gateways* do que os sistemas que usam TCP. Apesar de existirem perdas de datagramas quando o *gateway* muda, os dados recebidos não atingem o nível zero, como pode ser visto

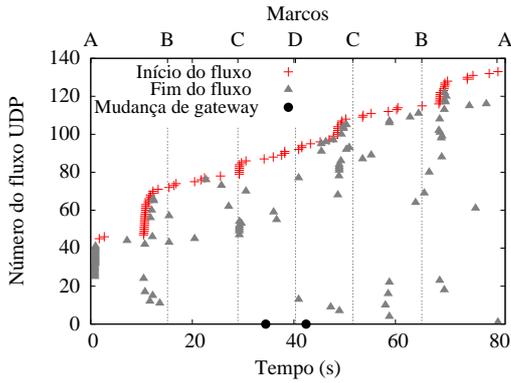
na Figura 4.4(a). Tal fato é reforçado pela avaliação subjetiva do vídeo. Os espectadores deram nota 3,77 ($\sigma = 0,62$) para a amostra de referência e 3,24 ($\sigma = 0,62$) para a amostra do usuário. Esse comportamento também é refletido no número cumulativo de datagramas UDP recebidos, que é ilustrado pela Figura 4.4(b). Nessa figura, a curva de referência mostra o número cumulativo de bytes enviados pelos dois *gateways* e a curva medida o número de bytes recebidos pelo cliente. Nesse caso, a curva medida está abaixo da de referência porque não há retransmissões para os datagramas perdidos. Nota-se que essas curvas não apresentam platôs extensos como a Figura 4.3(b). A Figura 4.4(c) ilustra o início e o fim de diferentes fluxos UDP recebidos pelo cliente móvel e a Figura 4.4(d) mostra a função distribuição cumulativa da duração dos fluxos UDP. Assim como as conexões TCP no PPStream, a maioria dos fluxos UDP é de curta duração, o que significa que poucos fluxos são afetados pelas mudanças de *gateway*. Cerca de 80% dos fluxos duram menos do que 30 segundos. Outra observação importante é que o intervalo entre os datagramas UDP recebidos é menos sensível às mudanças de *gateways*, como pode ser comprovado pela comparação das Figuras 4.3(e) e 4.4(e). Diferentemente do que ocorrem com os sistemas que usam TCP, as mudanças de *gateway* não resultam em perdas no tráfego de subida nos sistema que usam majoritariamente o UDP, como é visto na Figura 4.4(f). Isso ocorre porque, independentemente do uso de NAT, o endereço IP do participante de destino se mantém inalterado mesmo que o cliente móvel mude o *gateway* ao qual está associado. Dessa forma, se os datagramas enviados pelo cliente móvel são recebidos por pelo menos um *gateway*, a taxa de entrega não é afetada. Por outro lado, na direção de descida, os participantes localizados na Internet têm de mudar o endereço IP de destino de acordo com o *gateway* ao qual o cliente se associa durante a movimentação. Nesse caso, o endereço IP de destino é igual ao endereço do *gateway*, que usa NAT. Essa operação pode levar algum tempo e, frequentemente, resulta em falhas de transmissão independentemente do protocolo de transporte usado. Como o tráfego TCP é bidirecional, tal protocolo também é afetado por esse problema na direção de subida. Portanto, a combinação de mudanças de *gateway*, NAT e TCP sempre resulta em falhas de transmissão quando os segmentos são enviados para o *gateway* ao qual o cliente móvel estava previamente associado.



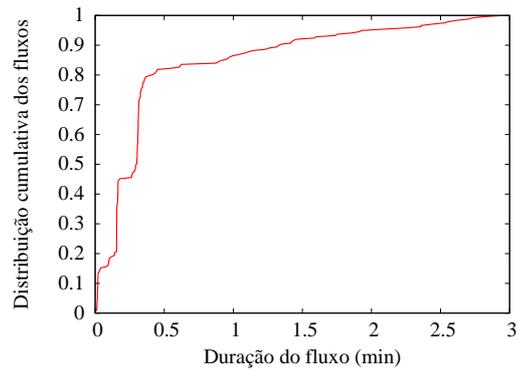
(a) Datagramas UDP recebidos.



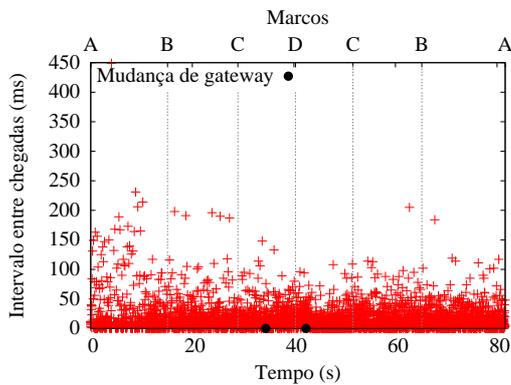
(b) Número cumulativo de bytes UDP recebidos.



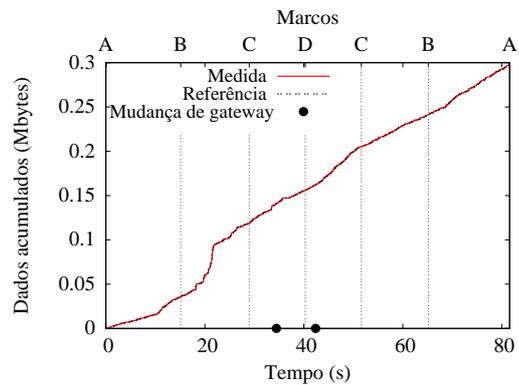
(c) Duração dos fluxos UDP.



(d) CDF da duração dos fluxos UDP.



(e) Intervalo entre os datagramas UDP recebidos.



(f) Número cumulativo de bytes UDP enviados.

Figura 4.4: Medições com o SopCast.

À primeira vista, pode parecer que o melhor desempenho do SopCast se dá exclusivamente pelo menor número de mudanças de *gateway* quando as Figuras 4.3 e 4.4 são comparadas. No entanto, quando são analisadas todas as rodadas dos experimentos, é possível notar que, em média, o número de mudanças de *gateway* para os experimentos com o PPStream e o SopCast, apresentados na Tabela 4.2, são próximos e que, mesmo assim, o desempenho do SopCast é superior ao do PPStream.

4.4.4 Resumo dos Experimentos

A Tabela 4.2 resume as medições obtidas com o SopCast, o PPStream, o PPLive e o TVAnts. Cada medida foi repetida cinco vezes para cada aplicação. Mostra-se a média e o desvio padrão de algumas medidas de interesse. A taxa de entrega do tráfego UDP é definida como a razão entre o número de datagramas UDP transmitidos e o número de datagramas recebidos pelo cliente móvel. Com as fontes TCP, o cálculo da taxa de entrega não é direto porque um segmento é retransmitido até ser entregue com sucesso ou até ocorrer um estouro do temporizador. Sendo assim, uma perda de segmento deveria acontecer apenas após múltiplas retransmissões. Além disso, as retransmissões também podem ser causadas por perdas de reconhecimentos positivos (ACKs). Como no cenário analisado deseja-se medir as falhas de transmissão, o percentual do tráfego TCP que não é retransmitido é definido pela razão entre o número de primeiras tentativas de transmissão de um segmento e o número total de tentativas, incluindo as possíveis retransmissões.

Nota-se que tanto para os sistemas que usam TCP quanto para os que usam UDP, o tráfego de descida é mais afetado do que o tráfego de subida. Como visto na seção anterior, na direção de descida os participantes que estão na Internet têm de mudar o endereço IP de destino à medida que o cliente se movimenta e muda o seu *gateway*. Por outro lado, na direção de subida o endereço IP do participante de destino não se altera mesmo que o cliente móvel mude o *gateway* ao qual está associado. É importante ressaltar também que na direção de subida, os sistemas que usam TCP também sofrem com o problema da mudança de endereço de destino, uma vez que o tráfego do TCP é bidirecional e o cliente está atrás de um dispositivo NAT. Observa-se também que a taxa de entrega dos datagramas UDP na direção de descida é mais afetada do que a dos segmentos TCP, pois as fontes não usam mecanismos de controle de congestionamento como ocorre com o TCP. Por outro lado, na direção de subida, a taxa de entrega é alta, o que confirma os resultados mostrados nas Seções 4.4.2 e 4.4.3. É importante notar ainda que o número de mudanças de *gateway* é similar para todos os sistemas. Essa variável depende apenas do protocolo de roteamento usado, que no caso, é o mesmo para todos os sistemas. Por fim, pode-se concluir que taxa de abertura de novas conexões ou fluxos é o fator que mais contribui para a robustez dos sistemas par-a-par em cenários de mobilidade.

Tabela 4.2: Resumo dos experimentos.

| | Sistemas | | | |
|--|--------------------------|---------------------------|-------------------------|-------------------------|
| | SopCast | PPLive | TVAnts | PPStream |
| Protocolo de transporte | UDP/poucos segmentos TCP | TCP/poucos datagramas UDP | TCP/UDP | TCP |
| Duração do experimento (s) | 80,80 ($\sigma=1,56$) | 79,62 ($\sigma=0,52$) | 78,25 ($\sigma=3,75$) | 78,27 ($\sigma=5,02$) |
| Número de mudanças de <i>gateway</i> | 6,40 ($\sigma=2,58$) | 10,00 ($\sigma=2,06$) | 8,00 ($\sigma=1,67$) | 8,00 ($\sigma=3,58$) |
| Taxa de entrega UDP (%) - descida | 79,53 ($\sigma=16,44$) | 66,18 ($\sigma=2,87$) | 66,46 ($\sigma=1,59$) | - |
| Taxa de entrega UDP (%) - subida | 99,99 ($\sigma=0,01$) | 99,80 ($\sigma=0,33$) | 98,82 ($\sigma=0,97$) | - |
| Taxa de novos fluxos UDP por segundo | 0,84 ($\sigma=0,02$) | 1,07 ($\sigma=0,01$) | 0,31 ($\sigma=0,01$) | - |
| Taxa de não-retransmissão TCP (%) - descida | 62,50 ($\sigma=41,46$) | 93,92 ($\sigma=2,07$) | 92,48 ($\sigma=4,40$) | 97,47 ($\sigma=0,70$) |
| Taxa de não-retransmissão TCP (%) - subida | 100,00 ($\sigma=0,00$) | 97,31 ($\sigma=1,23$) | 95,63 ($\sigma=2,03$) | 95,73 ($\sigma=2,70$) |
| Taxa de novas conexões TCP por segundo | 0,03 ($\sigma=0,01$) | 1,90 ($\sigma=0,01$) | 0,51 ($\sigma=0,01$) | 1,44 ($\sigma=0,05$) |

Capítulo 5

Seleção de Pares em Sistemas

Par-a-Par de Vídeo sob Demanda

Nos sistemas par-a-par de distribuição de vídeo sob demanda, os participantes controlam a reprodução do vídeo e, por isso, podem estar assistindo a partes diferentes desse vídeo em um mesmo intervalo de tempo¹. Sendo assim, o número de participantes que simultaneamente se interessa por um mesmo trecho de um vídeo tende a ser menor do que nos sistemas de difusão, em virtude da dessincronização da reprodução dos participantes. Conseqüentemente, o número de parceiros dos quais se podem solicitar pedaços desse trecho também é menor. Para aumentar a disponibilidade de pedaços de vídeo, os sistemas par-a-par de vídeo sob demanda, desenvolvidos recentemente, armazenam os pedaços de vídeo em um *cache* [3, 12]. Nesses sistemas, cada participante reserva uma quantidade de espaço em disco, fixa e predeterminada, para servir de *cache* dos pedaços de vídeo recebidos recentemente. Assim, os participantes que não estejam necessariamente assistindo ao mesmo trecho do vídeo podem ter pedaços para trocar, uma vez que todos ou uma parte dos pedaços de vídeo recebidos por um participante estão armazenados no seu *cache*. No entanto, o uso do *cache* para lidar com a dessincronização da reprodução dos parceiros possui alguns compromissos de custo e desempenho. O tamanho do *cache* de cada participante é limitado e, portanto, algoritmos de substituição de *cache* são necessários [66]. Quanto maior a qualidade do vídeo, maior a quantidade de espaço em disco necessária para armazenar os pedaços recebidos por unidade de tempo. A

¹Em 2008, o PPLive registrou a marca de 150 mil usuários e 500 canais simultâneos [2].

TV de alta definição (*High-Definition Television* - HDTV) se tornou popular e, com isso, os usuários desejam vídeos de alta qualidade mesmo na Internet. No entanto, para armazenar 30 minutos de um vídeo de alta qualidade transmitido a uma taxa de 5 Mb/s é necessário um espaço em disco da ordem de 1 GB. Além disso, a demanda por aplicações de distribuição de vídeo não está mais restrita aos usuários de computadores pessoais. Ela é cada vez maior entre usuários dotados de dispositivos móveis, que possuem severas restrições de espaço em disco [7]. Portanto, é necessário desenvolver mecanismos para reduzir o espaço em disco usado para armazenar os pedaços de vídeo recebidos.

Os mecanismos de seleção de pares podem ser usados para aumentar a disponibilidade de pedaços de interesse entre um participante e seus parceiros. Tais mecanismos também podem ser empregados para reduzir o tamanho do *cache* usado pelos sistemas par-a-par de vídeo sob demanda. Atualmente, a maioria dos sistemas seleciona e atualiza os parceiros de um participante de forma aleatória [14, 12, 2]. A seleção aleatória de parceiros se justifica para a difusão do vídeo, porque todos os participantes do sistema estão interessados em um dado trecho do vídeo no mesmo intervalo de tempo. Por outro lado, no vídeo sob demanda, a seleção aleatória é menos eficiente, uma vez que os participantes entram no sistema em instantes diferentes e controlam a reprodução do vídeo e, como consequência, os pedaços de vídeo que interessam a um participante podem ser diferentes de um participante para outro. A probabilidade de um participante recém-chegado ao sistema selecionar parceiros com interesses mútuos depende do tempo em que os demais participantes estão no sistema e do número de interações já feitas pelos participantes.

Neste trabalho, é proposto um mecanismo de seleção de pares específico para sistemas par-a-par de vídeo sob demanda. O objetivo do mecanismo LIPS (*Lifetime-based Peer Selection*) é aumentar a probabilidade de estabelecimento de parcerias entre participantes com mais pedaços de interesse para trocarem entre si. O processo de seleção é baseado no tempo de vida (*lifetime*) dos participantes, que é definido com o tempo decorrido desde o início da reprodução do vídeo. Um participante seleciona como parceiros outros participantes que possuam tempos de vida próximos ao seu. Com isso, a chance de se escolher parceiros com interesse em um trecho de vídeo comum é maior do que na seleção aleatória, uma vez que os participantes

selecionados iniciaram a reprodução do vídeo em trechos próximos. Se os parceiros estão assistindo à mesma parte do vídeo, eles estão interessados nos mesmos pedaços ao mesmo tempo. Consequentemente, os pedaços recebidos anteriormente podem ser descartados e, assim, o tamanho do *cache* pode ser reduzido. A seleção baseada no tempo de vida, entretanto, não garante a disponibilidade de pedaços de interesse entre os parceiros. Mesmo tendo iniciado a reprodução do vídeo em trechos próximos, os participantes podem interagir com o sistema e alterar o ponto de reprodução do vídeo a qualquer instante. Portanto, o padrão de comportamento dos usuários em serviços de vídeo sob demanda afeta o desempenho dos mecanismos de seleção de pares.

Neste capítulo, o mecanismo proposto LIPS é apresentado em detalhes. O desempenho do LIPS é avaliado através de simulações para diferentes padrões de comportamento dos usuários em sistemas de distribuição de vídeo sob demanda. Compara-se o LIPS com outros três mecanismos: a seleção aleatória, a seleção baseada na capacidade dos participantes e a seleção baseada no tempo de reprodução. Os modelos do sistema e de comportamento de usuários adotados neste trabalho e o simulador implementado para avaliar o desempenho dos mecanismos também são descritos. Discutem-se ainda alguns trabalhos relacionados aos sistemas par-a-par de vídeo sob demanda, especialmente, sobre os mecanismos de seleção de parceiros.

5.1 Trabalhos Relacionados

Sistemas par-a-par que implementam serviços de vídeo sob demanda enfrentam grandes desafios [2]. Existem trabalhos que propõem novos sistemas específicos para tal finalidade [2, 12, 13]. Dois deles foram apresentados na Seção 3.4: o BulletMedia [12] e o PPLive [2]. Ambos definem um *cache* de pedaços além do *buffer* de reprodução. O *cache* é usado para aumentar a disponibilidade de pedaços de interesses entre os parceiros. Basicamente, o *cache* armazena pedaços já assistidos pelo participante. Sendo assim, quanto mais pedaços armazenados, maior a probabilidade de um participante ter pedaços que são de interesse dos seus parceiros. A desvantagem dessa abordagem é que quanto maior a qualidade e o tamanho do vídeo, maior o tamanho do *cache* necessário. Uma abordagem para reduzir o tamanho do *cache* é

adotar mecanismos de seleção de parceiros que considerem fatores característicos dos sistemas de vídeo sob demanda. Com esses mecanismos, é possível formar parcerias entre participantes que assistem a trechos próximos do vídeo. Conseqüentemente, só é preciso armazenar pedaços desses trechos e não de todo o vídeo. Por isso, a disponibilidade de pedaços de interesse é extremamente dependente do tamanho do *cache*. Tanto o BulletMedia quanto o PPLive selecionam os parceiros de um participante aleatoriamente e, por isso, o desempenho desses sistemas está diretamente relacionado ao tamanho do *cache* que utilizam.

Existem muitas propostas de mecanismos de seleção de pares para sistemas de compartilhamento de arquivos e de difusão [67, 68]. Na sua maioria, tais propostas buscam aumentar a eficiência do encaminhamento do conteúdo de vídeo na estrutura de distribuição. Por isso, o processo de seleção de pares é baseado em métricas de desempenho relacionadas normalmente à capacidade de transmissão, à continuidade de reprodução e à manutenção da qualidade do vídeo [69].

Rejaie e Stafford [70] e Zhang *et al.* [14] alegam que seleção aleatória dos pares é suficiente para atender os requisitos desejados pela distribuição de vídeo. Por outro lado, Liang e Nahrstedt [68] consideram que a seleção aleatória e a visão parcial dos participantes do sistema não são apropriadas para o estabelecimento de parcerias que atendam os requisitos de qualidade de serviço exigidos pela distribuição de vídeo. Para garantir esses requisitos, Liang e Nahrstedt propõem a separação de participantes do sistema em grupos de acordo com as características de desempenho, como a banda passante. Dessa forma, um participante seleciona como seus parceiros somente membros do seu grupo. Para Liu *et al.*, a presença de pares em condições satisfatórias para atender a transmissão de vídeo está associada à organização dos nós fisicamente na rede [67]. Segundo os autores, nós fisicamente próximos têm maiores chances de se tornarem melhores “encaminhadores” de vídeo. Entretanto, em redes sobrepostas, como as construídas pelos sistemas par-a-par, um nó vizinho pode não estar fisicamente próximo. Logo, organizar a rede sobreposta de forma a otimizar as distâncias físicas entre os nós vizinhos é o desafio.

O mecanismo proposto neste trabalho difere dos demais por ser específico para a distribuição de vídeo sob demanda. Com o LIPS, reduz-se o principal problema do vídeo sob demanda, que é a disponibilidade de pedaços de interesse entre os parceiros

durante a reprodução do vídeo. Esse objetivo é priorizado em relação ao aumento da eficiência da distribuição ao se escolher parceiros, por exemplo, com maior banda passante disponível e menor atraso. Considera-se que é mais importante se ter pedaços de vídeo para trocar usando um enlace que não seja ótimo do que se ter um enlace ótimo, mas com poucos pedaços para trocar. Essa afirmação é comprovada ao se comparar o desempenho do LIPS e da seleção baseada na capacidade dos participantes, apresentada na Seção 5.5.

Outro ponto a ser discutido é a comparação entre o uso do tempo de vida e do tempo de reprodução como métricas de seleção de parceiros. Enquanto o tempo de vida expressa o intervalo de tempo decorrido entre a entrada do participante no sistema e o tempo atual, o tempo de reprodução indica o ponto do vídeo que o participante está reproduzindo naquele momento, que é expresso pelo intervalo de tempo do início do vídeo até o ponto de reprodução atual. Assim, o tempo de vida leva em conta apenas o momento em que o participante entrou no sistema enquanto o tempo de reprodução leva em conta toda e qualquer interação do participante com a reprodução do vídeo.

Existem propostas de sistemas de vídeo baseados na arquitetura em árvore que definem as relações de pai e filho de acordo com o tempo de reprodução dos participantes [71, 72]. A ideia básica é que os filhos estejam em pontos de reprodução anteriores aos dos pais. No entanto, dependendo do número e do tipo das interações, a frequência de reconstrução da árvore pode ser alta, o que aumenta a sobrecarga de controle e influencia a continuidade de recepção do vídeo. Não foi encontrado na literatura nenhum sistema baseado na arquitetura em malha que use o tempo de reprodução como métrica de seleção de parceiros. Neste trabalho, o tempo de vida foi escolhido para ser usado pelo mecanismo proposto LIPS para evitar a sobrecarga de mensagens de controle.

Apesar do tempo de reprodução ser uma métrica mais fiel, o uso dessa métrica para selecionar os parceiros requer que a fonte e os participantes estejam atualizados sobre o tempo de reprodução atual de todos os participantes do sistema. Portanto, para toda interação de qualquer um dos participantes, a informação sobre o ponto de reprodução de um participante após a interação deve ser difundida pela rede aumentando a sobrecarga de mensagens de controle. Uma das formas de difundir essa

informação é enviar mensagens de atualização para a fonte. Assim, um participante que deseja estabelecer parcerias deve sempre consultar a fonte para se informar sobre os pontos de reprodução atualizados de cada participante do sistema. Esse procedimento é detalhado na Seção 5.4. Por outro lado, ao usar o tempo de vida, uma mensagem de atualização só é necessária quando um novo participante entra no sistema e o tempo de vida é facilmente atualizado pela fonte e pelos demais participantes do sistema de forma totalmente independente e autônoma. É necessário apenas que eles definam uma unidade de tempo comum e incrementem um contador, como é visto na Seção 5.3. Existe, portanto, um compromisso entre o aumento da disponibilidade de pedaços de interesse nos parceiros e a sobrecarga de controle. Para avaliar esse compromisso, a seleção baseada no tempo de reprodução é implementada e comparada com o LIPS. Os resultados são apresentados na Seção 5.5.

5.2 Modelo do Sistema

Neste trabalho, assume-se um sistema par-a-par de vídeo sob demanda que funciona da seguinte maneira. Todos os participantes do sistema são cooperativos e estão interessados no mesmo vídeo, que é dividido em pedaços. Inicialmente, todos os pedaços são armazenados somente pela fonte. Um participante deve contatar a fonte para entrar no sistema. O endereço da fonte é descoberto por um mecanismo de inicialização independente. A fonte ou um nó designado envia para o participante que deseja entrar no sistema um subconjunto de participantes ativos, chamados de candidatos. Esse subconjunto é determinado de acordo com o mecanismo de seleção de parceiros empregado pelo sistema. O participante, então, envia uma requisição de estabelecimento de parcerias para cada candidato. Caso o participante receba de volta um reconhecimento positivo de pelo menos um dos candidatos, considera-se que o participante acaba de entrar no sistema. Os candidatos que responderam com reconhecimentos positivos são considerados parceiros do novo participante. Os parceiros são os vizinhos de um participante na rede sobreposta e um participante somente troca mapas de *buffer* e pedaços de vídeo com seus parceiros.

Cada participante possui um *cache*, além do *buffer* de reprodução, ambos de tamanho predeterminado e fixo. O *buffer* de reprodução é usado para armazenar

os pedaços que estão na iminência de serem enviados para o tocador de vídeo. O *cache*, por sua vez, armazena todos ou um subconjunto dos pedaços recebidos pelo participante, incluindo os que estão no *buffer* de reprodução e os que já foram ou que ainda serão encaminhados para esse *buffer*. Além disso, os participantes armazenam somente pedaços do vídeo que estão assistindo e ao entrarem no sistema não possuem nenhum pedaço previamente armazenado no *cache*, exceto a fonte. Define-se também que o mapa de *buffer* representa a disponibilidade de pedaços no *cache* de cada participante.

Considera-se também o mecanismo de seleção de pedaços proposto por Shah e Pâris [73]. Esse mecanismo define quais pedaços de vídeo devem ser requisitados de acordo com o tempo atual de reprodução do vídeo. Para tanto, define-se uma janela de pedaços de interesse, que contém os próximos W pedaços de vídeo a serem requisitados por um participante. Com esse mecanismo, os participantes não desperdiçam tempo solicitando e recebendo pedaços de vídeo que não pertençam a sua janela de interesse, que contém os pedaços de vídeo mais urgentes. Sempre que o primeiro pedaço de vídeo da janela é recebido, o início da janela é deslocado até a posição do próximo pedaço que ainda não foi recebido. O tamanho da janela de interesse é definido em função do tamanho do *buffer* de reprodução, de forma que seja possível para um participante receber todos os pedaços de vídeo da janela nesse intervalo de tempo. Logo, deve-se satisfazer a seguinte relação, considerando que W é o tamanho da janela em número de pedaços:

$$W = \frac{db}{c}, \quad (5.1)$$

onde d é o tamanho do *buffer* de reprodução, b é a taxa de transmissão do vídeo e c o tamanho de um pedaço de vídeo. O tamanho do *buffer* de reprodução representa a diferença entre o tempo de reprodução de um pedaço e o tempo em que esse pedaço foi encaminhado para o *buffer*².

Assume-se que os pedaços da janela de interesse são escalonados de acordo com o algoritmo de seleção do mais-raro-primeiro (*rarest-first algorithm*), proposto por Zhang *et al.* para o sistema CoolStreaming [14]. Esse algoritmo calcula, primeiramente, o número de fornecedores potenciais para cada pedaço contido na janela de

²Por simplificação, ao longo do texto, considera-se que o tempo de reprodução do vídeo e o tempo em que um dado pedaço é encaminhado para o *buffer* de reprodução são sinônimos.

interesse. Os fornecedores de um pedaço de interesse são os parceiros de um participante que possuem tal pedaço armazenado em seu *cache* de pedaços. Considera-se que a probabilidade de um pedaço com menos fornecedores potenciais ser recebido após o seu tempo de reprodução é maior do que a de pedaços com mais fornecedores. Por isso, o escalonador determina o fornecedor de cada pedaço de interesse começando por aqueles com apenas um fornecedor potencial, em seguida os com dois e assim por diante. Para os pedaços com mais de um fornecedor potencial, seleciona-se o fornecedor com a maior capacidade em termos de banda passante.

5.3 O Mecanismo LIPS (*Lifetime-based Peer Selection*)

O mecanismo de seleção de parceiros proposto, chamado de LIPS (*Lifetime-based Peer Selection*), é específico para sistemas par-a-par de vídeo sob demanda [74, 75]. O objetivo do LIPS é formar parcerias entre participantes que tenham pedaços de interesse³ para trocar. Como já mencionado anteriormente, tal tarefa se torna mais difícil em sistemas de vídeo sob demanda, quando comparada à difusão de vídeo ao vivo, pois os participantes podem interagir com o vídeo pausando, avançando ou retrocedendo a sua reprodução. Dessa forma, o número de participantes que simultaneamente se interessa por um pedaço de vídeo específico é variável e, conseqüentemente, o número de parceiros dos quais se podem solicitar esse pedaço específico tende a ser menor do que na difusão de vídeo.

O processo de seleção de pares no LIPS é baseado no tempo de vida dos participantes. Define-se o tempo de vida de um participante como o tempo decorrido desde o início da reprodução do vídeo até o tempo atual. Portanto, o tempo de vida indica há quanto tempo um usuário está no sistema assistindo a um determinado vídeo. Ele difere do tempo de reprodução, uma vez que ele é similar a um relógio de unidades de tempo e não se altera devido a uma interação. Ao usar o tempo de vida para selecionar os parceiros, aumenta-se a probabilidade de estabelecer parcerias entre participantes interessados nos mesmos pedaços de vídeo em um dado intervalo de tempo, como ocorre na difusão de vídeo. A suposição fundamental

³Os pedaços de interesse são os pedaços de vídeo referentes à atual janela de interesse.

do LIPS é que participantes que entram no sistema em instantes próximos assistem a trechos próximos do vídeo e, provavelmente, estão interessados nos mesmos pedaços de vídeo. Portanto, esses participantes são selecionados como parceiros. Além disso, se os parceiros estão assistindo à mesma parte do vídeo, eles estão interessados nos mesmos pedaços de vídeo ao mesmo tempo. Conseqüentemente, os pedaços recebidos anteriormente podem ser descartados, porque ambos os participantes já assistiram/encaminharam esse trecho do vídeo. Dessa forma, o LIPS reduz o tamanho do *cache* exigido dos participantes do sistema.

O número de pedaços de interesse disponíveis nos parceiros também depende da interatividade dos participantes. Apesar de terem controle total sobre a reprodução do vídeo a qualquer instante, estudos mostram que os participantes de sistemas de vídeo sob demanda seguem padrões de interatividade determinados pelas características do vídeo [76, 77, 78, 79, 80]. A duração e o conteúdo do vídeo, por exemplo, são fatores que influenciam a frequência e os tipos de interações feitas pelos participantes. Portanto, a interatividade dos usuários afeta o desempenho dos mecanismos de seleção de pares e, por isso, o desempenho do LIPS é avaliado para diferentes padrões de interatividade obtidos a partir de dados reais, como é visto na Seção 5.5.

O LIPS funciona da seguinte forma. O tempo de vida de um participante é dado por um contador que é incrementado em uma unidade a cada unidade de tempo. Por isso, o LIPS não requer a sincronização do relógio dos pares. O mecanismo proposto requer apenas que os participantes do sistema definam a mesma unidade de tempo para incrementar os contadores.

O LIPS funciona da seguinte forma. O tempo de vida de um participante é dado por um contador que é incrementado em uma unidade a cada segundo. Por isso, o LIPS não requer a sincronização do relógio dos pares. A escolha do segundo como unidade de tempo se justifica porque a maioria dos vídeos tem duração da ordem de dezenas de minutos e, assim, o escorregamento dos relógios, que atualmente é da ordem de poucos milissegundos por dia, tem pouca influência no desempenho do LIPS. Cada participante é responsável por atualizar e divulgar o seu tempo de vida para os outros participantes. Entretanto, o LIPS não exige a introdução de novas mensagens no sistema. A difusão dos tempos de vida é feita através das mensagens de existência (*membership messages*), presentes nos sistemas par-a-par de distri-

buição de vídeo. Tais mensagens são enviadas pelos participantes para seus vizinhos para anunciar sua atividade no sistema e, conseqüentemente, para que os demais pares atualizem suas listas de participantes [8]. Dessa forma, é necessário apenas introduzir um novo campo nas mensagens de existência e uma nova entrada na lista de participantes para informar e armazenar o tempo de vida de cada um deles. O campo tempo de vida de uma entrada da lista de participantes pode ser atualizado tanto pelo próprio nó responsável pela lista quanto pela chegada de uma nova mensagem de existência. Um novo participante entra no sistema como já descrito na Seção 5.2. Com o LIPS, o nó responsável pelo envio da lista de candidatos, a fonte ou o nó designado, assume que o tempo de vida de um novo participante é zero e envia, como candidatos a parceiros, os L últimos participantes a entrarem no sistema. Assume-se que a recepção da primeira resposta positiva para o estabelecimento de parcerias dispara o contador do tempo de vida e, simultaneamente, a reprodução do vídeo é iniciada. Periodicamente, o processo de atualização do conjunto de parceiros é disparado. Esse processo é necessário, porque o número máximo de parceiros pode ainda não ter sido atingido ou porque alguns dos parceiros podem ter falhado ou deixado o sistema. Além disso, um participante e seus parceiros podem mudar seus pontos de reprodução e, conseqüentemente, podem deixar de ter pedaços de interesse para trocar e, nesse caso, é interessante atualizar o conjunto de parceiros para substituí-los.

O Algoritmo 1 é periodicamente executado por cada nó n para atualizar o seu conjunto de parceiros. As entradas do algoritmo, no instante de atualização t , são o conjunto de participantes ativos no sistema, $\mathcal{A}(t)$, e o conjunto atual de parceiros do nó n , $\mathcal{P}_n(t)$. O algoritmo possui dois procedimentos principais. O primeiro procedimento calcula o número de parceiros a serem selecionados no instante t , representado pela variável *novosParceiros*. O segundo procedimento seleciona os parceiros de n de acordo com o tempo de vida dos participantes ativos no sistema.

O primeiro passo do procedimento para determinar o número de parceiros a serem selecionados é identificar se um dos parceiros p deixou o sistema ou falhou. Para isso, basta verificar as entradas ativas da lista de participantes do nó. Caso não haja nenhuma entrada relativa a um dos parceiros, ele é removido do conjunto $\mathcal{P}_n(t)$ e a variável *novosParceiros* é incrementada em uma unidade. Essa variável indica

o número de nós a serem selecionados ao término da atualização. Nesse ponto, se o tamanho de $\mathcal{P}_n(t)$ for menor do que o tamanho máximo do conjunto, definido por `NUM_MAX_PARCEIROS`, calcula-se o número de novos parceiros a serem selecionados. Do contrário, calcula-se a utilidade de cada parceiro para o nó. Tal função retorna uma pontuação para os parceiros de acordo com o número médio de pedaços enviados e recebidos durante o intervalo de tempo de um período de atualização. Se um nó tem uma utilidade abaixo do limiar U ele é removido. Um número pequeno de pedaços de vídeo recebidos e enviados pode indicar que os parceiros não possuem mais interesse no mesmo trecho, devido a uma interação, ou que o enlace entre os pares está congestionado. Se algum parceiro foi removido do conjunto, novamente a variável *novosParceiros* é incrementada.

O passo seguinte inicia o procedimento de seleção de parceiros de acordo com seus tempos de vida. Para tanto, utiliza-se a técnica de anel expansivo. Dois parâmetros são definidos: o fator de expansão do anel α e o limite do anel R . A partir desses parâmetros, é possível determinar se o tempo de vida de um nó i , $l(t)_i$, é suficientemente próximo do tempo de vida de n , $l(t)_n$, de acordo com a seguinte equação:

$$|l(t)_n - l(t)_i| < \alpha R. \quad (5.2)$$

Os primeiros parceiros selecionados são aqueles dentro do limite inicial R do anel. A cada passo do laço, se o número de parceiros a serem selecionados ainda não foi atingido ou se o parâmetro α ainda está dentro do limiar A especificado, o parâmetro α é incrementado e o limite do anel é expandido. O processo é repetido até um dos critérios de parada ser atingido. Se o algoritmo para porque *novosParceiros* = 0, o conjunto terá o maior tamanho possível no momento e a diferença entre os tempos de vida de n e dos parceiros estará dentro do limite desejado. Se o algoritmo para porque α é igual a A , a diferença máxima desejada entre os tempos de vida de n e de seus candidatos a parceiros foi atingida. Além disso, o tamanho do conjunto de parceiros pode não ter atingido o tamanho máximo possível. Nessa situação, dependendo dos valores de A e R , a diferença entre o tempo de vida de n e dos seus candidatos a parceiros pode ser grande e, provavelmente, tais participantes não terão pedaços de interesse para trocarem entre si. Por isso, adiciona-se a fonte ao conjunto

Algoritmo 1 Atualização do conjunto de parceiros.

Requer: $t = atualizar$, $\mathcal{A}(t) > 0$ e $n \in \mathcal{A}(t)$

$novosParceiros \leftarrow 0$

para todo nó $p \in \mathcal{P}_n(t)$ **faça**

se $p \notin \mathcal{A}(t)$ **então**

 remover ($p, \mathcal{P}_n(t)$)

$novosParceiros \leftarrow novosParceiros + 1$

fim do se

fim do para

se tamanho de $\mathcal{P}_n(t) < \text{NUM_MAX_PARCEIROS}$ **então**

$novosParceiros \leftarrow novosParceiros + \text{calcNovosParceiros}(\mathcal{A}(t), \text{maxParceiros})$

senão

para todo nó $p \in \mathcal{P}_n(t)$ **faça**

se utilidade(p) $< U$ **então**

 remover ($p, \mathcal{P}_n(t)$)

$novosParceiros \leftarrow novosParceiros + 1$

fim do se

fim do para

fim do se

$\alpha \leftarrow 1$

enquanto $novosParceiros > 0$ ou $\alpha < A$ **faça**

para todo nó $i \in \mathcal{A}(t)$ e $i \notin \mathcal{P}_n(t)$ e $i \neq n$ **faça**

se $|l(t)_n - l(t)_i| < \alpha R$ e tamanho de $\mathcal{P}_n(t) < \text{NUM_MAX_PARCEIROS}$ **então**

 adicionar ($i, \mathcal{P}_n(t)$)

$novosParceiros \leftarrow novosParceiros - 1$

fim do se

fim do para

$\alpha \leftarrow \alpha + 1$

se $\alpha = A$ **então**

 adicionar ($fonte, \mathcal{P}_n(t)$)

fim do se

fim do enquanto

de parceiros de n e, assim, o nó n poderá solicitar pedaços de vídeo diretamente à fonte.

Vale ressaltar que o LIPS é um mecanismo de seleção de parceiros que pode ser usado por qualquer sistema par-a-par de vídeo sob demanda baseado na arquitetura em malha, nos quais os participantes estabeleçam parcerias e troquem pedaços do vídeo.

5.4 O Ambiente de Simulação

Para avaliar o desempenho do mecanismo proposto e compará-lo com os demais mecanismos, foi desenvolvido um simulador específico, escrito em linguagem C++, que é composto por dois módulos. O primeiro módulo é responsável pela geração da carga sintética para representar o comportamento interativo dos participantes durante a reprodução do vídeo. O segundo módulo implementa os mecanismos de seleção de parceiros e o escalonamento e a troca de pedaços de vídeo. Dessa forma, simula-se o padrão de interatividade dos participantes e observa-se a evolução da reprodução, a qualidade do vídeo recebido pelos participantes e a disponibilidade dos pedaços de interesse nos parceiros.

O padrão de interatividade dos participantes afeta o desempenho dos sistemas par-a-par de vídeo sob demanda, principalmente os mecanismos de seleção de pares. Segundo esse padrão, os participantes alteram o ponto de reprodução do vídeo e, conseqüentemente, alteram o trecho do vídeo no qual têm interesse. O tempo de entrada de um participante no sistema, a frequência, o tipo e a duração das interações são parâmetros necessários para se determinar o comportamento e gerar cargas sintéticas próximas da realidade.

Neste trabalho, utilizam-se as caracterizações de comportamento de usuários em serviços de vídeo sob demanda apresentadas por Costa *et al.* [78] e Huang *et al.* [2]. A caracterização de Costa *et al.* foi escolhida por analisar diferentes tipos de vídeo, de diferentes tamanhos e popularidades e a de Huang *et al.* por apresentar dados referentes ao PPLive, um sistema par-a-par de vídeo sob demanda comercial. As duas caracterizações trabalham com arquivos de carga com dados reais. A frequência de cada tipo de interação e as distribuições e seus respectivos parâmetros, extraídas dos trabalhos citados anteriormente, são apresentadas nas Tabelas 5.1 e 5.2.

Costa *et al.* definem padrões de comportamento para duas categorias de vídeo:

vídeos de entretenimento e vídeos educacionais. Os vídeos de entretenimento são parametrizados de acordo com arquivos de carga dos servidores da TV UOL. A TV UOL é o serviço de vídeo sob demanda do UOL, o maior provedor de acesso à Internet da América Latina. Quando os experimentos foram realizados, o serviço tinha entre 34 e 520 mil acessos diários. Os vídeos educacionais, por sua vez, têm seus parâmetros extraídos de arquivos de carga do eTeach, um servidor de conteúdo educacional localizado na Universidade de Wisconsin, EUA. Foram escolhidos um vídeo de entretenimento e outro educacional que possuem durações e padrões de interatividade diferentes. A duração do vídeo de entretenimento escolhido é menor do que a do vídeo educacional. O intervalo entre as chegadas dos participantes no vídeo de entretenimento é menor do que no vídeo educacional e, por isso, o número de participantes simultâneos no sistema é maior para o vídeo de entretenimento. O número de interações do vídeo de entretenimento também é menor do que o do educacional. Quando existe uma interação no vídeo de entretenimento, na maioria das vezes, é uma pausa. Por outro lado, no vídeo educacional a maioria das interações ainda são pausas, mas a frequência dos saltos para trás e para frente e a distância desses saltos é maior do que no vídeo de entretenimento.

Huang *et al.* [2] apresentam os dados referentes a três filmes diferentes. O filme considerado neste trabalho foi escolhido por ter uma duração, aproximadamente, vinte vezes maior do que a do vídeo de entretenimento e cinco vezes maior do que o vídeo educacional. O filme também é mais popular do que o vídeo educacional, uma vez que o intervalo entre as chegadas dos participantes é menor. O número de interações do filme é menor do que o do vídeo educacional, mas vale ressaltar que o sistema PPLive não permite pausas, por isso, são considerados apenas saltos. Huang *et al.* não definem a probabilidade de direção dos saltos e a distância desses saltos. Por isso, considera-se neste trabalho que as probabilidades de um salto para frente ou para trás são as mesmas. Assume-se também que a distância dos saltos é distribuída uniformemente entre 1 segundo e a duração do vídeo. Essa distribuição foi escolhida para avaliar os mecanismos em um cenário com grandes saltos, que são maiores, em média, do que os saltos dos vídeos de entretenimento e educacional.

O primeiro módulo do simulador, responsável pela geração de carga sintética, recebe como entrada as distribuições e parâmetros definidos, respectivamente, nas

Tabela 5.1: Frequência e tipo de interações dos participantes.

| | Entretenimento (TV UOL) | Educacional (eTeach) | Filme |
|---|------------------------------------|---------------------------------|--------------|
| Tamanho do vídeo | 5 minutos | 20 minutos | 110 minutos |
| Número de interações por usuário | 1,29 | 4,74 | 3,60 |
| Pausa | 83% | 57% | - |
| Salto para trás | 13% | 25% | 50% |
| Salto para frente | 4% | 18% | 50% |

Tabelas 5.1 e 5.2 e produz como saída um padrão de comportamento para cada participante do sistema.

O segundo módulo é responsável pela dinâmica do sistema. Anteriormente à seleção de parceiros, o tempo de vida dos participantes e o ponto de reprodução do vídeo são atualizados de acordo com o padrão de comportamento de cada participante em um dado instante de tempo. Em seguida, os parceiros são selecionados ou atualizados, de acordo com o mecanismo de seleção considerado. São avaliados quatro mecanismos de seleção de parceiros diferentes: o mecanismo proposto LIPS, a seleção aleatória, a seleção baseada na capacidade dos participantes e a seleção baseada no tempo de reprodução. O LIPS é implementado de acordo com o Algoritmo 1, apresentado na Seção 5.3. A seleção baseada no tempo de reprodução também é implementada de acordo com mesmo algoritmo, mas, ao invés de se considerar o tempo de vida dos participantes, considera-se o tempo de reprodução no qual se encontra o vídeo no instante da atualização. Dessa forma, os participantes selecionam como parceiros outros participantes em pontos de reprodução próximos ao seu. Na seleção aleatória, um participante sorteia seus parceiros entre os participantes disponíveis no sistema, tanto no instante de entrada quanto no instante de atualização. Os participantes disponíveis são os que estão ativos no sistema e que não atingiram o número máximo de parceiros definido pelo sistema. Na seleção baseada na capacidade, um participante seleciona como parceiros os participantes

Tabela 5.2: Distribuições que caracterizam o comportamento dos participantes.

| | Entretenimento (TV UOL) | Educacional (eTeach) | Filme |
|--|---|---|----------------------------------|
| Tamanho do vídeo | 5 minutos | 20 minutos | 110 minutos |
| Chegada dos participantes | Exponencial (média = 10 s) | Lognormal ($\mu = 3,95, \sigma = 0,95,$ média = 81,55 s) | Exponencial (média = 79,04 s) |
| Duração da pausa | Weibull ($\alpha = 11,11, \beta = 0,57,$ média = 25 s) | Weibull ($\alpha = 13, \beta = 0,42$ média = 55 s) | - |
| Distância de saltos para trás | Exponencial (média = 20 s) | Exponencial (média = 40 s) | Uniforme (1 s, 6600 s) |
| Distância de saltos para frente | Exponencial (média = 7 s) | Exponencial (média = 40 s) | Uniforme (1 s, 6600 s) |

de maior capacidade de *upload* disponíveis. Assume-se, nesse caso, que a capacidade de *download* dos participantes é sempre suficiente para receber os pedaços enviados pelos parceiros na taxa máxima de transmissão. Vale ressaltar que, para todos os mecanismos, calcula-se o número de pedaços de vídeo recebidos de um parceiro durante um período de atualização para determinar a sua utilidade. Dessa forma, os parceiros de menor utilidade podem ser substituídos a cada atualização e, assim, pode-se aumentar a eficiência da troca de pedaços de vídeo. Por fim, os pedaços de vídeo são escalonados de acordo com o algoritmo do mais-raro-primeiro, apresentado na Seção 5.2 e, em seguida, transmitidos. O tempo de transmissão dos pedaços de vídeo é calculado de acordo com a capacidade de *upload* de seus fornecedores. Não são considerados os atrasos de propagação no meio e de espera nas filas dos roteadores da rede física.

5.5 Resultados

O desempenho do mecanismo proposto LIPS é comparado, através de simulações, com o desempenho de outros três mecanismos descritos anteriormente: a seleção

aleatória, a seleção baseada na capacidade de *upload* dos parceiros e a seleção baseada no tempo de reprodução do vídeo. A avaliação está dividida em duas partes. Na primeira, os quatro mecanismos são avaliados para três padrões de comportamento de participantes, apresentados na Seção 5.4, relativos aos vídeos de entretenimento e educacional e ao filme. Na segunda parte, o impacto da variação de parâmetros do sistema nos mecanismos de seleção é avaliado para o vídeo de entretenimento. Analisa-se também o impacto dos parâmetros do anel expansivo no desempenho do LIPS. As métricas usadas nas duas partes da avaliação indicam a eficiência dos parceiros selecionados por cada mecanismo e mostram o impacto de cada mecanismo na continuidade de reprodução do vídeo, um dos parâmetros mais críticos para se determinar a satisfação dos usuários. Mais detalhes sobre a definição de cada métrica são discutidos a seguir.

Os seguintes parâmetros de simulação, resumidos na Tabela 5.3, são considerados nas duas partes da avaliação. Assume-se que a taxa de reprodução dos três vídeos é igual a 350 kb/s e que um pedaço contém 10 s de vídeo. Assim, o tamanho de um pedaço de vídeo é igual a 437,5 kB. Os vídeos de entretenimento e educacional e o filme são compostos, respectivamente, por 30, 120 e 660 pedaços. Durante a simulação, 200 participantes chegam aos sistemas para assistir aos vídeos. A simulação é iniciada no instante zero, no qual somente a fonte participa do sistema, e termina quando não resta mais nenhum participante assistindo ao vídeo. Os mecanismos de seleção também foram avaliados para mais e menos de 200 participantes no sistema e os resultados são similares aos apresentados a seguir. Esse é o número de participantes durante toda a simulação é não o número de participantes simultâneos no sistema. O número de participantes simultâneos indica a popularidade do vídeo que é determinada pelo intervalo entre as chegadas dos participantes. Portanto, o vídeo de entretenimento é o mais popular e o vídeo educacional o menos popular, de acordo os dados da Tabela 5.2 da Seção 5.4. Assume-se que os participantes iniciam a reprodução do ponto inicial do vídeo⁴ e que nenhum falha ou deixa o sistema durante a reprodução do vídeo. São definidas quatro capacidades de *upload* diferentes de acordo com resultados experimentais apresentados por Huang *et al.* [2]. Os participantes podem contribuir com 200 (36%⁵), 360 (28%), 600 (25%)

⁴De acordo com Costa *et al.* [78], mais de 98% iniciam a reprodução do vídeo do seu começo.

⁵O percentual de participantes com essa capacidade no sistema.

e 1000 kb/s (11%). A capacidade da fonte é igual a 1000 kb/s. A cada atualização, os quatro mecanismos substituem o parceiro de menor utilidade de um dado participante. A utilidade de um parceiro é definida pelo número de pedaços de interesse recebidos desse parceiro em um intervalo de atualização. O mesmo padrão de interatividade dos participantes gerado a cada rodada de simulação é aplicado aos quatro mecanismos. Para todos os pontos das curvas apresentadas nessa seção, calcula-se um intervalo de confiança para um nível de confiabilidade de 95%, representado por barras verticais.

Tabela 5.3: Parâmetros considerados em todas as simulações.

| | Entretenimento | Educacional | Filme |
|--|-----------------------|--------------------|--------------|
| Tamanho do vídeo V (pedaços) | 30 | 120 | 660 |
| Taxa de transmissão do vídeo (kb/s) | 350 | 350 | 350 |
| Duração do pedaço de vídeo (s) | 10 | 10 | 10 |
| Tamanho do pedaço de vídeo (kB) | 437,5 | 437,5 | 437,5 |
| Número de participantes | 200 | 200 | 200 |

Assume-se também que o tamanho do *cache* é fixo e igual para todos os participantes. Sendo assim, um *cache* de P pedaços significa que um participante pode armazenar até P pedaços do vídeo em seu disco rígido. Esses P pedaços de vídeo correspondem aos W pedaços que pertencem à atual janela de interesse e aos $P - W$ pedaços anteriores ao início da janela. Os demais pedaços são removidos do *cache* à medida que o vídeo é reproduzido. É importante ressaltar que um *cache* de 100% do tamanho do vídeo significa que todos os pedaços de vídeo recebidos podem ser mantidos no *cache*, independentemente do ponto atual de reprodução, e não que um participante já possui todos os pedaços de vídeo armazenados. Ao entrar no sistema, assume-se que um participante não possui nenhum pedaço de vídeo armazenado no *cache* e que os pedaços são recebidos dos parceiros ao longo da reprodução de vídeo. Assume-se também que um participante anuncia todos os pedaços disponíveis em seu *cache* enviando mapas de *buffer* para seus parceiros.

5.5.1 Avaliação dos Mecanismos para os Diferentes Vídeos

Nessa seção, os quatro mecanismos de seleção de parceiros são avaliados para os diferentes padrões de interatividade, representados pelos vídeos de entretenimento e educacional e o filme. A Tabela 5.4 apresenta os parâmetros de simulação considerados nessa seção. O impacto da variação desses parâmetros é avaliado na Seção 5.5.2. Para todos os mecanismos, o número máximo de parceiros é igual a 4, como sugerido por Zhang *et al.* [14]. O número de participantes que se conectam diretamente à fonte também é limitado a 8. O tamanho da janela de interesse é definido de acordo com a Equação 5.1. Considera-se que o tamanho do *buffer* de reprodução é igual a 60 s [73] para todos os vídeos e, portanto, $W = 6$ pedaços. O período de atualização de parceiros é igual a 10 s. Os parâmetros para o LIPS e a seleção baseada no tempo de reprodução são os seguintes. O limite do anel R é definido de acordo com o intervalo médio entre as chegadas dos participantes considerado para cada vídeo. Caso R seja menor do que esse intervalo, a probabilidade de se escolher parceiros no primeiro passo do anel expansivo é pequena. Sendo assim, $R=10$ s para o vídeo de entretenimento e $R=80$ s para o vídeo educacional e também para o filme. O limiar do fator de expansão é igual a $A = 5$ para todos os vídeos.

Tabela 5.4: Parâmetros para a avaliação dos mecanismos para os diferentes vídeos.

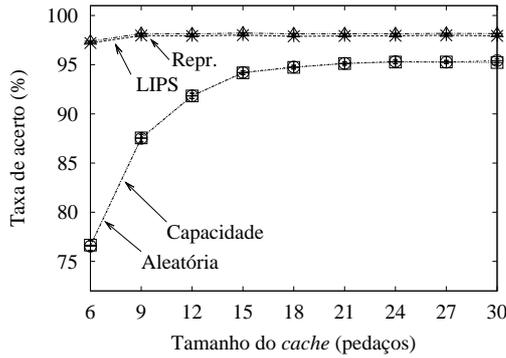
| | Entretenimento | Educacional | Filme |
|---|-----------------------|--------------------|--------------|
| Número máximo de parceiros | 4 | 4 | 4 |
| Número máximo de parceiros da fonte | 8 | 8 | 8 |
| Janela de interesse W (pedaços) | 6 | 6 | 6 |
| <i>Buffer</i> de reprodução (s) | 60 | 60 | 60 |
| Intervalo de atualização (s) | 10 | 10 | 10 |
| Limite do anel R (s) | 10 | 80 | 80 |
| Limiar do fator de expansão A | 10 | 10 | 10 |

Primeiramente, analisa-se a eficiência dos parceiros selecionados por cada um dos mecanismos. As métricas são avaliadas em função do tamanho do *cache*. Na avaliação, o tamanho do *cache* varia de 20% a 100% do tamanho de vídeo de entretenimento em termos de pedaços de vídeo. Para os outros dois vídeos, o tamanho do

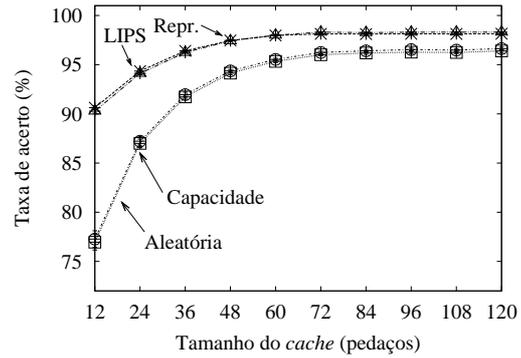
cache varia de 10% a 100%. Quanto maior o número de pedaços armazenados no *cache*, maior a probabilidade dos participantes terem pedaços de interesse para trocar. Esse comportamento esperado é evidenciado nas Figuras 5.1 e 5.2. Essas figuras apresentam a taxa de acertos e o número médio de pedaços de interesse por parceiro em função do tamanho do *cache*. Cada ponto das curvas representa o valor médio da métrica em questão para todos os participantes do sistema. Em algumas figuras, as curvas do LIPS e da seleção baseada no tempo de reprodução estão sobrepostas, assim como, as curvas da seleção aleatória e da seleção baseada na capacidade dos participantes. Isso se deve ao fato de que tais mecanismos apresentam desempenhos similares em alguns dos cenários analisados.

A taxa de acerto (*hit ratio*) é definida como o percentual de pedaços de interesse que um participante encontra armazenados no *cache* de seus parceiros, excluindo-se a fonte. Os pedaços de interesse de um participante são os pedaços ainda não recebidos que estão no intervalo de sua janela de interesse. O número médio de pedaços de interesse por parceiro indica quantos fornecedores um pedaço possui. Essas duas métricas mostram as interseções de interesse entre um participante e seus parceiros. Quanto maior a taxa de acertos e quanto mais pedaços de interesse por parceiro, menor a probabilidade de se esperar por pedaços para continuar a reprodução do vídeo sem interrupções.

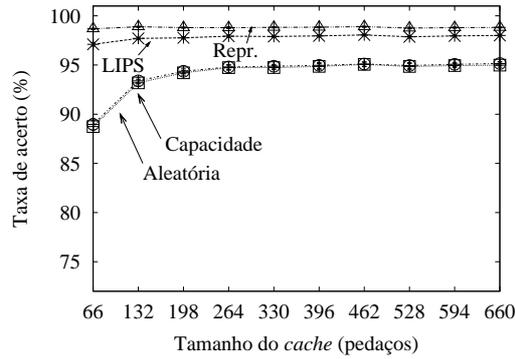
A Figura 5.1 mostra que o LIPS é o mais eficiente dos três mecanismos. O mecanismo proposto proporciona taxas de acerto mais altas do que a seleção aleatória e a seleção baseada na capacidade de *upload* para os três vídeos, independentemente do tamanho do *cache*. O LIPS, com poucos pedaços de vídeo armazenados em disco, provê taxas de acerto mais altas do que as máximas obtidas com esses dois mecanismos. Além disso, o LIPS proporciona taxas de acerto similares à seleção baseada no tempo de reprodução com a vantagem de não exigir mensagens de controle adicionais, como é visto adiante. Para o vídeo de entretenimento, de acordo com a Figura 5.1(a), o LIPS atinge uma taxa de acerto superior a 97% considerando um *cache* de apenas 6 pedaços de vídeo, o que corresponde a apenas 20% do tamanho do vídeo. Por outro lado, essa taxa de acerto não é atingida pela seleção aleatória e pela seleção com base na capacidade de *upload*, cujas taxas máximas são, respectivamente, 95,2% e 95,4% quando 100% do vídeo pode ser armazenado no *cache*.



(a) Entretenimento ($V = 30$ pedaços).



(b) Educacional ($V = 120$ pedaços).



(c) Filme ($V = 660$ pedaços).

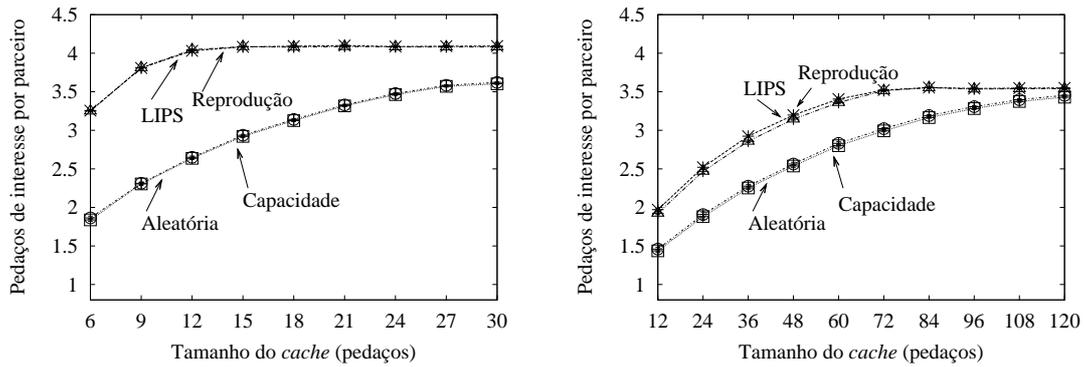
Figura 5.1: A taxa de acerto em função do tamanho do *cache*.

Portanto, para o vídeo de entretenimento, o LIPS proporciona uma taxa de acerto maior do que a máxima proporcionada pelos mecanismos de seleção aleatória e baseada na capacidade de *upload* e, para tanto, requer apenas que 20% dos pedaços de vídeo sejam armazenados em disco. Observa-se também que a taxa máxima de acerto da seleção baseada no tempo de reprodução é apenas 0,3 pontos percentuais superior à taxa máxima do LIPS. Para o vídeo educacional, ilustrado pela Figura 5.1(b), a taxa de acerto proporcionada pelo LIPS, considerando um *cache* que pode armazenar até 40% dos pedaços do vídeo, já é superior à taxa máxima de acerto da seleção aleatória e da seleção baseada na capacidade dos participantes. A taxa de acerto com o LIPS é de aproximadamente 97,5%. Assim como para o vídeo de entretenimento, a taxa máxima obtida com o LIPS é igual a 98,2% que é apenas 0,1 pontos percentuais inferior a da seleção com base no tempo de reprodução. Para o filme, ilustrado pela Figura 5.1(c), o LIPS também é o mecanismo mais eficiente. Para um *cache* de 66 pedaços, ou seja, que armazena até 10% do vídeo, a diferença

entre as taxas de acerto do LIPS e da seleção aleatória é de 11 pontos percentuais. Para esse mesmo tamanho de *cache*, que corresponde a 28 MB de espaço em disco, a taxa de acerto proporcionada pelo LIPS é de 97% que é superior à taxa máxima da seleção aleatória e da baseada na capacidade, que é próxima a 95% e obtida para um *cache* de 288 MB, que corresponde a 100% do vídeo. Ainda para um *cache* que pode armazenar 10% do vídeo, a diferença entre as taxas de acerto do LIPS e da seleção baseada na reprodução é de apenas um ponto percentual.

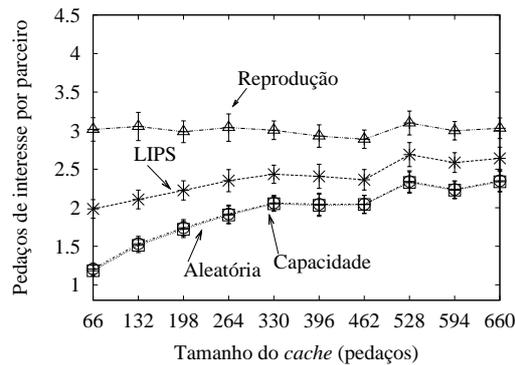
É possível concluir também que o desempenho do mecanismo proposto é influenciado pela popularidade dos vídeos. Para vídeos mais populares, taxas próximas à máxima são alcançadas para tamanhos de *cache* pequenos se comparados ao tamanho do vídeo. Com o LIPS, para um *cache* de 20% do tamanho do vídeo, a taxa de acerto do vídeo de entretenimento, o mais popular, é apenas 0,7 pontos percentuais inferior à taxa máxima. Para o vídeo educacional, o menos popular, essa diferença é de 7,5 pontos percentuais. A popularidade do vídeo é dada pelo número de participantes simultâneos no sistema. Quanto mais popular é o vídeo, maior é o número de participantes simultâneos no sistema e, conseqüentemente, mais candidatos a parceiros um participante possui. Assim, torna-se mais fácil selecionar pares com pedaços de interesse para trocar. Durante as simulações, o número de participantes simultâneos é de aproximadamente 30, 16 e 20, respectivamente para os vídeos de entretenimento, educacional e o filme. Além disso, para vídeos populares, os intervalos entre a chegada dos participantes são menores. Com isso, menos pedaços de vídeo precisam ser armazenados porque a diferença entre os pontos de reprodução dos participantes tende a ser menor. Tais fatores explicam a diferença de desempenho do LIPS para os três vídeos.

As Figuras 5.2(a), 5.2(b) e 5.2(c) mostram o número médio de pedaços de interesse por parceiro, respectivamente, para os vídeos de entretenimento, educacional e o filme. Com o LIPS, os parceiros selecionados têm mais pedaços de interesse disponíveis para os três tipos de vídeo, quando comparado à seleção aleatória e à seleção com base na capacidade dos parceiros. O desempenho do LIPS e da seleção com base no tempo de reprodução são similares, lembrando que o mecanismo que usa tempo de reprodução requer muito mais mensagens de controle. Para o vídeo de entretenimento, os parceiros selecionados pelo LIPS possuem mais de 3,8 pedaços



(a) Entretenimento ($V = 30$ pedaços).

(b) Educacional ($V = 120$ pedaços).



(c) Filme ($V = 660$ pedaços).

Figura 5.2: O número de pedaços de interesse por parceiro em função do tamanho do *cache*.

de interesse para tamanhos de *cache* que podem armazenar 9 ou mais pedaços de vídeo. Como a janela de interesse possui 6 pedaços, isso significa que, em média, um parceiro selecionado pelo LIPS possui mais de 60% dos pedaços de interesse de um participante. Por exemplo, esse valor é igual a 38%, para a seleção aleatória e o mesmo tamanho de *cache*. Para o vídeo educacional e o filme, o número médio de pedaços de interesse por parceiro varia, respectivamente, entre 33 e 58% e 35 e 42% da janela de interesse. Esses valores são superiores ou similares aos proporcionados pelos demais mecanismos. Portanto, os parceiros selecionados pelo LIPS são mais eficientes do que os selecionados pela seleção aleatória e a seleção de acordo com a capacidade dos participantes, porque, como é visto a seguir, quanto maior o número de pedaços de interesse por parceiros, mais fácil é para se escalonar e receber os pedaços de vídeo antes do seu tempo de reprodução.

A explicação para o melhor desempenho do LIPS é a seguinte. A probabilidade

de um participante recém-chegado ao sistema selecionar parceiros com pedaços de interesse depende de dois fatores: o tempo em que os demais participantes chegaram ao sistema e o número de interações já feitas pelos participantes. A seleção aleatória e a baseada na capacidade de *upload* dos participantes não considera nenhum desses parâmetros. Por outro lado, o LIPS seleciona os parceiros de acordo com o tempo em que o participante está no sistema. Isso também explica o bom desempenho da seleção de parceiros baseada no tempo de reprodução. Dessa forma, a probabilidade do estabelecimento de parcerias entre participantes com pedaços de interesse para trocar aumenta, pois com o LIPS essa probabilidade depende apenas da interatividade dos participantes. Como mostram os resultados, a escolha dos parceiros baseada simplesmente no tempo de vida é suficiente para aumentar o desempenho do sistema em relação aos demais mecanismos analisados para diferentes padrões de interatividade. Isso mostra que a escolha de parceiros com tempos de entrada muito diferentes afeta mais a disponibilidade de pedaços de interesse entre os parceiros do que a interatividade dos participantes.

A principal diferença entre o LIPS e a seleção baseada no tempo de reprodução está na quantidade de mensagens de controle requerida por cada um dos mecanismos. Enquanto no LIPS não são necessárias mensagens de controle adicionais para cada interação dos participantes, o mecanismo baseado no tempo de reprodução precisa difundir mensagens de atualização a cada interação. O número de mensagens de controle não é analisado em função do tamanho do *cache*, pois só depende do número de interações realizadas pelos participantes. A Tabela 5.5 mostra os valores para o número médio de mensagens de controle considerando todas as rodadas de simulação para todos os valores de tamanho de *cache* apresentados anteriormente. O mecanismo proposto LIPS não exige nenhuma mensagem de controle adicional, em virtude das interações dos participantes. Por outro lado, como mostra a Tabela 5.5, a seleção de parceiros baseada no tempo de reprodução introduz, aproximadamente, de 6700 a 53000 mensagens de controle, dependendo do padrão de interatividade dos participantes. Considera-se que a cada interação, um participante envia uma mensagem de controle para a fonte para que ela seja atualizada sobre o seu ponto atual de reprodução. Além disso, ao se escolher novos parceiros, um participante deve enviar uma nova mensagem de controle à fonte requisitando os participantes

ativos com tempos de reprodução próximos ao seu. A fonte, por sua vez, responde ao participante com outra mensagem de controle contendo a lista de candidatos. Portanto, cada interação gera uma mensagem de controle adicional e a cada seleção de um novo parceiro são necessárias mais duas mensagens de controle. Conclui-se, então, que com LIPS é possível se obter uma eficiência similar à da seleção baseada no tempo de reprodução, porém, sem nenhuma mensagem de controle adicional.

Tabela 5.5: Número de mensagens de controle adicionais para o LIPS e para a seleção baseada no tempo de reprodução.

| | Entretenimento | Educacional | Filme |
|----------------------------|------------------------|--------------------------|---------------------------|
| LIPS | 0 | 0 | 0 |
| Tempo de reprodução | 6733 ($\sigma = 54$) | 26941 ($\sigma = 196$) | 52960 ($\sigma = 2571$) |

A Figura 5.3 ilustra a sobrecarga da fonte para cada um dos mecanismos de seleção. A sobrecarga da fonte é definida pela razão entre os pedaços de vídeo recebidos pelos participantes diretamente da fonte e o número total de pedaços recebidos pelos participantes. Para os vídeos mais populares, o preço pago pelo melhor desempenho do LIPS é um pequeno aumento da sobrecarga da fonte em relação à seleção aleatória e a baseada na capacidade dos participantes. Para o vídeo de entretenimento e o filme, respectivamente, a sobrecarga proporcionada pelo LIPS é no máximo 8 pontos percentuais maior do que a sobrecarga mais baixa proporcionada por um dos mecanismos. Para o vídeo educacional, o menos popular, entretanto, o LIPS, juntamente com a seleção baseada no tempo de reprodução, proporciona a sobrecarga de controle mais baixa. Esse sobrecarga é até 2,5 pontos percentuais menor do que a sobrecarga mais alta proporcionada por um dos mecanismos. Essa inversão de comportamento é explicada pela popularidade de cada vídeo. Para o LIPS, independentemente da popularidade, sempre que o parâmetro α do anel expansivo atinge o limiar A a fonte é adicionada ao conjunto de parceiros. Com a seleção aleatória, todos os participantes têm a mesma probabilidade de serem escolhidos como parceiros. Por isso, quanto mais participantes simultâneos no sistema, menor é a probabilidade da fonte ser selecionada como um dos parceiros. A sobrecarga da fonte para a seleção baseada na capacidade dos participantes, por sua vez,

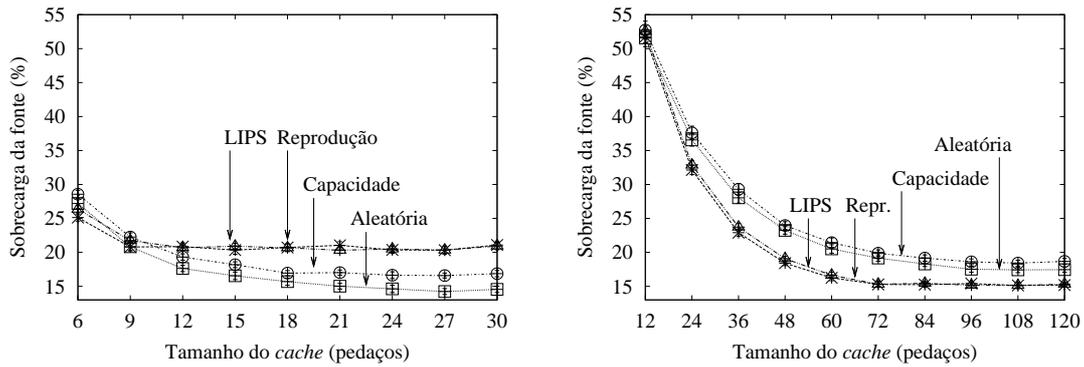
é maior do que na seleção aleatória. Com esse mecanismo, a probabilidade da fonte ser selecionada com um dos parceiros é maior do que na seleção aleatória, uma vez que a fonte é um dos participantes de maior capacidade de *upload* do sistema. A menor sobrecarga proporcionada pela seleção baseada no tempo de reprodução para o filme, Figura 5.3, é explicada pelo padrão de interatividade desse vídeo, composto por saltos, em média, de grande distância. Dependendo da distância do salto dado por um participante, seus atuais parceiros podem não ter pedaços de interesse relativos ao novo ponto de reprodução. Na seleção baseada no tempo de reprodução, a cada período de atualização após realizar um salto, um participante poderá selecionar um novo parceiro que esteja em um ponto de reprodução próximo. Dessa forma, esse mecanismo reage às interações sem ser necessário solicitar pedaços de vídeo diretamente à fonte. Por fim, observa-se também que quanto maior o tamanho do *cache* dos participantes, menor a sobrecarga da fonte. Com mais pedaços de vídeo disponíveis nos participantes, torna-se menor a necessidade de solicitar pedaços diretamente à fonte.

Após avaliar a eficiência dos mecanismos de seleção, analisa-se o impacto de cada mecanismo na qualidade do vídeo recebido. As métricas consideradas são o índice de continuidade, o número médio de pedaços de vídeo recebidos com atraso e o tempo de espera para receber esses pedaços atrasados. Dependendo dos parceiros selecionados, é possível que um participante não possua armazenado em seu *cache* ou em seu *buffer* de reprodução um pedaço de vídeo relativo ao ponto atual de reprodução. Conseqüentemente, a reprodução do vídeo nesses participantes é interrompida até que ele receba o pedaço atrasado ou um temporizador estoure. O tempo total em que a reprodução do vídeo fica interrompida por causa da falta de pedaços de vídeo necessários para o ponto atual de reprodução é chamado de tempo de espera. De acordo com o tempo de espera t_e , é possível calcular o índice de continuidade c , relativo à reprodução do vídeo em cada participante, com base na seguinte equação

$$c = \frac{t_r - t_e}{t_r}, \quad (5.3)$$

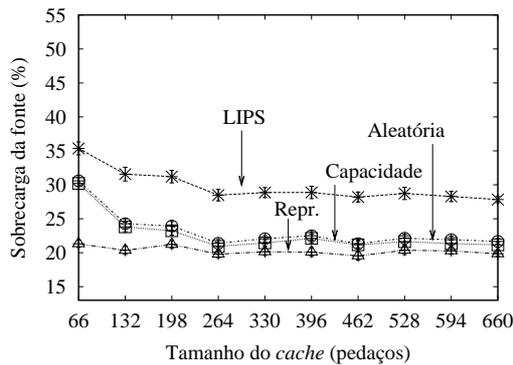
onde t_r é o tempo total de reprodução do vídeo, ou seja, o tempo em que o usuário assiste ao vídeo normalmente. Um índice de continuidade igual 100% significa que um usuário recebeu o vídeo sem interrupções.

Quanto mais pedaços de interesse disponíveis nos parceiros selecionados, mais



(a) Entretenimento ($V = 30$ pedaços).

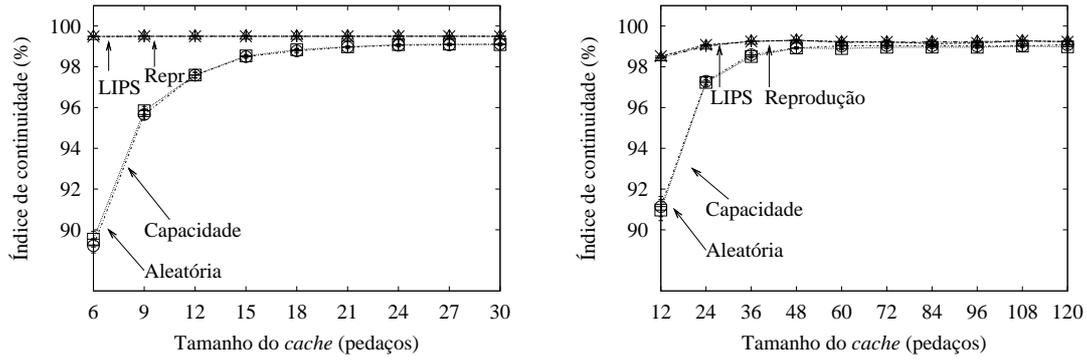
(b) Educacional ($V = 120$ pedaços).



(c) Filme ($V = 660$ pedaços).

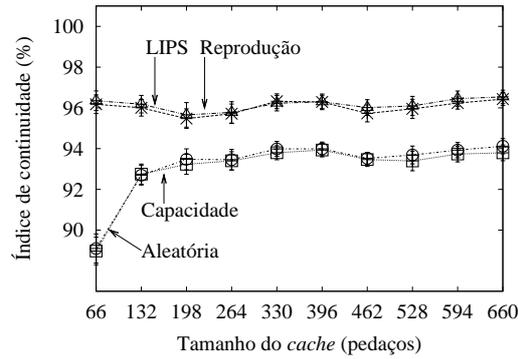
Figura 5.3: A sobrecarga da fonte em função do tamanho do *cache*.

fácil se torna manter a continuidade do vídeo. Essa afirmativa é confirmada pela Figura 5.4 e explica o melhor desempenho do LIPS. Essa figura mostra o índice de continuidade médio de todos os participantes em função do tamanho do *cache* para os quatro mecanismos de seleção. Nota-se que o mecanismo proposto proporciona um índice de continuidade superior a 99% para os vídeos de entretenimento e educacional, Figuras 5.4(a) e 5.4(b), independentemente do tamanho do *cache*. Para o filme, Figura 5.4(c), esse índice é sempre superior a 96%. Essa redução do índice de continuidade se deve ao padrão de interatividade do filme que, como visto anteriormente, é composto por saltos de grande distância. Apesar dessa redução, é para o filme que o LIPS proporciona os maiores ganhos em relação à seleção aleatória e à seleção baseada na capacidade dos participantes. Os ganhos são da ordem de 3 pontos percentuais, o que significa que os participantes com o LIPS esperam, em média, menos 200 s por pedaços de vídeo atrasados, considerando que o tempo total de reprodução é aproximadamente a duração do filme.



(a) Entretenimento ($V = 30$ pedaços).

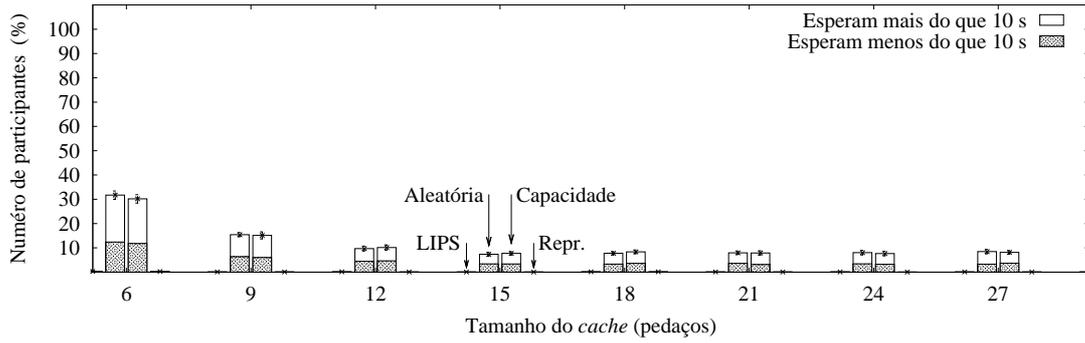
(b) Educacional ($V = 120$ pedaços).



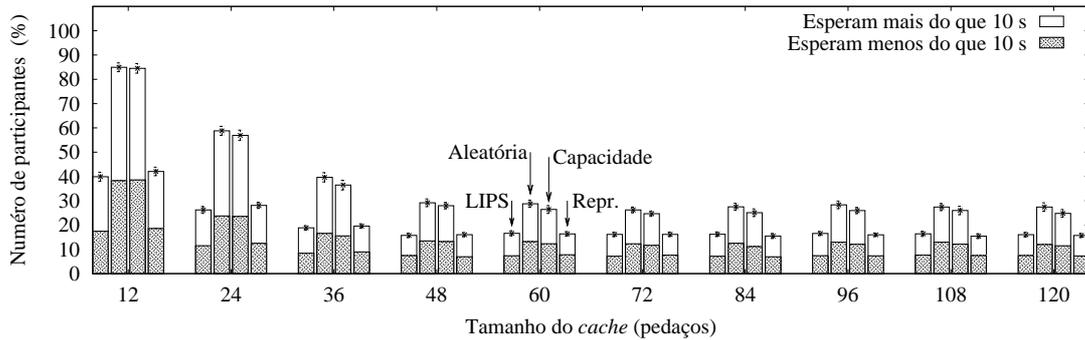
(c) Filme ($V = 660$ pedaços).

Figura 5.4: O índice de continuidade em função do tamanho do *cache*.

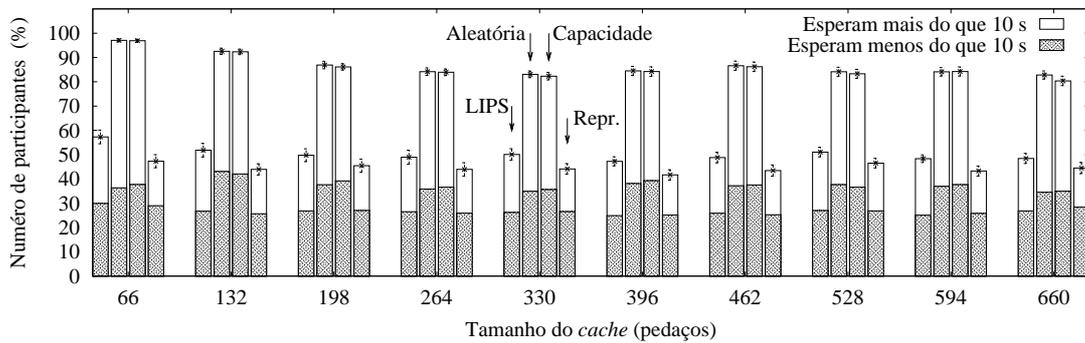
A Figura 5.5 mostra o número de participantes que esperam por pedaços atrasados e o tempo de espera em função do tamanho do *cache* para os quatro mecanismos. As partes listradas e brancas de cada barra vertical representam, respectivamente, o número de participantes que esperam menos e mais de 10 s para receber um pedaço de vídeo atrasado. Essa é a duração de um pedaço. A soma das duas partes representa o número total de participantes que esperam por pedaços atrasados. Para cada tamanho de *cache*, tem-se um grupo de quatro barras, cada uma representando um mecanismo. Da esquerda para direita, as barras representam, nessa ordem, o LIPS, a seleção aleatória, a baseada na capacidade e a baseada no tempo de reprodução. Para todos os vídeos e mecanismos, quanto maior o tamanho do *cache*, menor o tempo de espera experimentado pelos participantes. Para o vídeo de entretenimento, Figura 5.5(a), com o LIPS, menos de 1% dos participantes esperam por pedaços atrasados, mesmo que apenas 6 pedaços de vídeo possam ser armazenados no *cache*. Esse desempenho é similar ao da seleção baseada no tempo de



(a) Entretenimento ($V = 30$ pedaços).



(b) Educacional ($V = 120$ pedaços).



(c) Filme ($V = 660$ pedaços).

Figura 5.5: O número de participantes que esperam por um pedaço de vídeo em função do tamanho do *cache*.

reprodução. Para os demais mecanismos, o tempo de espera varia entre 30% e 8%, respectivamente, para o menor e o maior tamanho de *cache* considerado e a maioria dos participantes esperam mais do que 10 s para receber um pedaço atrasado. Para o vídeo educacional, um *cache* de 36 pedaços (30% do tamanho de vídeo) é suficiente para garantir o melhor desempenho do LIPS quando se compara o meca-

nismo proposto com a seleção aleatória e seleção baseada na capacidade. Para esse tamanho de *cache* com o LIPS ou com a seleção baseada no tempo de reprodução, aproximadamente 18% dos participantes podem experimentar uma interrupção na reprodução do vídeo. Esse número cresce para aproximadamente 25% quando os outros dois mecanismos podem armazenar todos os pedaços de vídeo no *cache*. Para o filme, Figura 5.5(c), todos os mecanismos apresentam um aumento no número de participantes que esperam por pedaços atrasados. Isso se deve aos saltos de grande distância, que caracterizam o comportamento interativo dos participantes desse vídeo. Mesmo nesse cenário de grande interatividade, o LIPS se mostra mais eficiente. Portanto, pode-se concluir que com o LIPS, os participantes esperam por menos tempo os pedaços de vídeo atrasados.

5.5.2 Avaliação dos Mecanismos para os Diferentes Parâmetros do Sistema

O impacto de diferentes parâmetros dos sistemas par-a-par de vídeo sob demanda é avaliado nessa seção. Primeiramente, são considerados os parâmetros comuns a todos os mecanismos: o intervalo de atualização do conjunto de parceiros, o número máximo de parceiros de um participante e da fonte e o tamanho da janela de interesse. Em seguida analisa-se o impacto dos parâmetros específicos do mecanismo proposto. Nas duas análises, considera-se o vídeo de entretenimento, por esse ser o mais popular. Todos os parâmetros, exceto o que está sendo analisado, têm seus valores apresentados nas Tabelas 5.3 e 5.4. Assume-se também que o tamanho do *cache* é igual a 15 pedaços, o que significa que até 50% dos pedaços do vídeo de entretenimento podem ser armazenados.

Parâmetros do Sistema Par-a-Par de Vídeo

O primeiro parâmetro analisado é o intervalo de atualização do conjunto de parceiros. As Figuras 5.6(a) e 5.6(b) mostram, respectivamente, o índice de continuidade e a taxa de acerto para os quatro mecanismos em função desse parâmetro. Pode-se notar que a variação do intervalo de atualização reduz o desempenho da seleção aleatória e da seleção baseada na capacidade. Tanto o índice de continuidade quanto a taxa de acerto decrescem mais rapidamente para esses dois mecanismos

quando comparados ao LIPS e ao mecanismo de seleção de acordo com o tempo de reprodução. Isso ocorre porque somente no instante de atualização os participantes podem substituir os parceiros selecionados. Como os parceiros escolhidos pela seleção aleatória e pela seleção com base na capacidade são menos eficientes do que os selecionados pelo LIPS, quanto maior o intervalo de atualização, maior o tempo para se substituir um parceiro pouco eficiente. Conseqüentemente, maior o tempo que esse parceiro não contribui para o bom funcionamento do sistema.

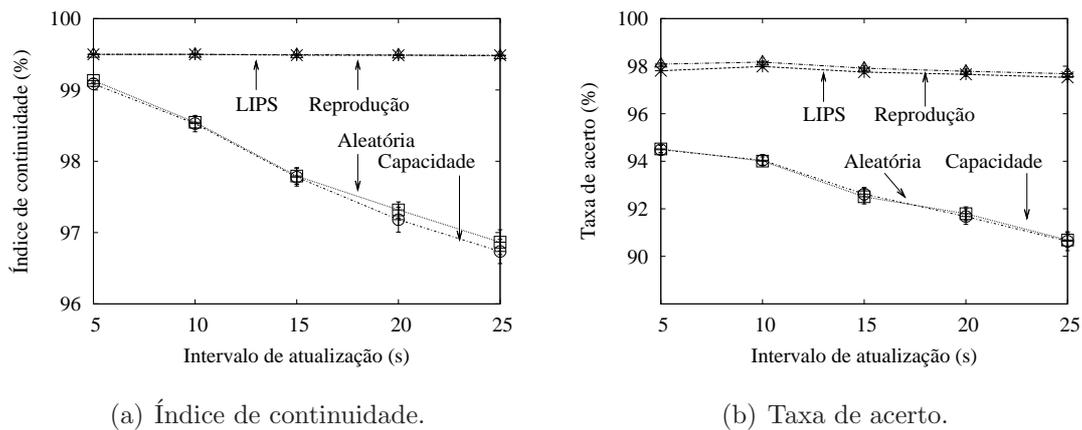


Figura 5.6: O impacto do intervalo de atualização nos mecanismos de seleção de parceiros.

O número máximo de parceiros por participante também influencia os mecanismos de seleção, como mostram as Figuras 5.7(a) e 5.7(b). O efeito esperado é que com mais parceiros por participante, menor a sobrecarga da fonte. Esse efeito é observado para todos os mecanismos. Entretanto, a partir de um determinado valor, que varia de acordo com o mecanismo de seleção, o índice de continuidade é pouco influenciado pelo número de parceiros. Para o LIPS, por exemplo, a continuidade se mantém constante com a variação do número de parceiros. Nesse caso, com o LIPS, quando se tem menos parceiros, adiciona-se a fonte ao conjunto de parceiros para aumentar a eficiência do sistema. Tal fato pode ser observado na Figura 5.7(b). Por outro lado, para a seleção aleatória, a partir de 12 parceiros a continuidade se mantém constante. Isso reforça a afirmação de Zhang *et al.* [14] de que existe um limiar para o número máximo de parceiros.

Outro parâmetro avaliado é o número máximo de parceiros da fonte. Esses participantes são os que recebem pedaços de vídeo diretamente da fonte. Com esse

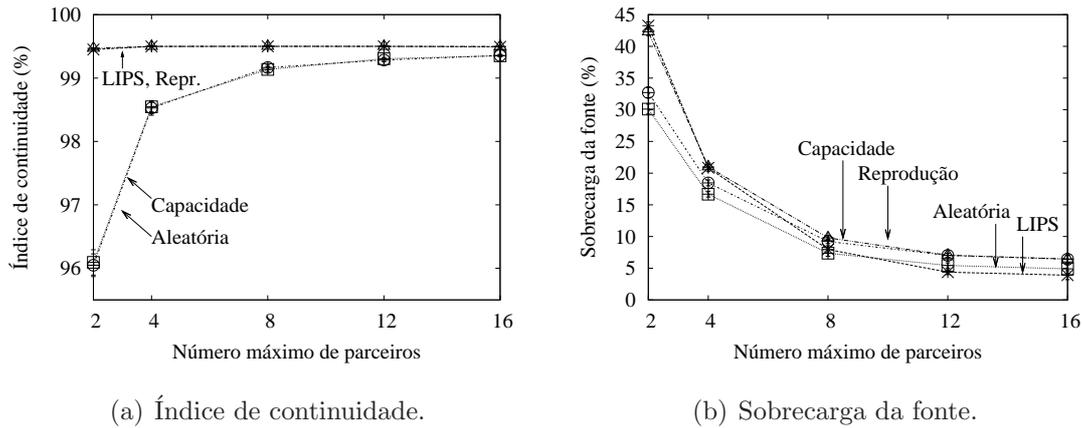


Figura 5.7: O impacto do número máximo de parceiros nos mecanismos de seleção de parceiros.

parâmetro, é possível controlar a sobrecarga da fonte e verificar o seu impacto na continuidade do vídeo, como ilustram as Figuras 5.8(a) e 5.8(b). Observa-se que para o LIPS o índice de continuidade e a sobrecarga se mantêm constantes à medida que o número máximo de parceiros da fonte aumenta. Isso ocorre porque sempre que o parâmetro α do anel expansivo atinge o limiar A a fonte é adicionada ao conjunto de parceiros, mesmo que a fonte já tenha o número máximo de parceiros permitido. Dessa forma, aumenta-se a eficiência do LIPS em troca do aumento da sobrecarga da fonte. Para a seleção aleatória e para a seleção baseada na capacidade dos participantes, quanto maior o número de parceiros da fonte, maior a sobrecarga e maior o índice de continuidade. Entretanto, para um número máximo de parceiros da fonte maior ou igual a 12, a sobrecarga da fonte imposta por esses dois mecanismos é maior do que a do LIPS e, mesmo assim, o índice de continuidade do mecanismo proposto se mantêm superior. Isso demonstra que a seleção de parceiros com mais pedaços de interesse disponíveis é o fator que mais contribui para o melhor desempenho do LIPS e não o aumento da sobrecarga da fonte.

A variação do tamanho da janela de interesse também é avaliada. Como visto, o tamanho janela de interesse é calculado em função do tamanho do *buffer* de reprodução, de acordo com a Equação 5.1. Portanto, assumindo que a taxa de transmissão do vídeo e o tamanho dos pedaços são constantes, aumentar o tamanho da janela significa aumentar o tamanho do *buffer* de reprodução. Dessa forma, tem-se um intervalo maior de tempo entre a recepção de um pedaço de vídeo e a sua re-

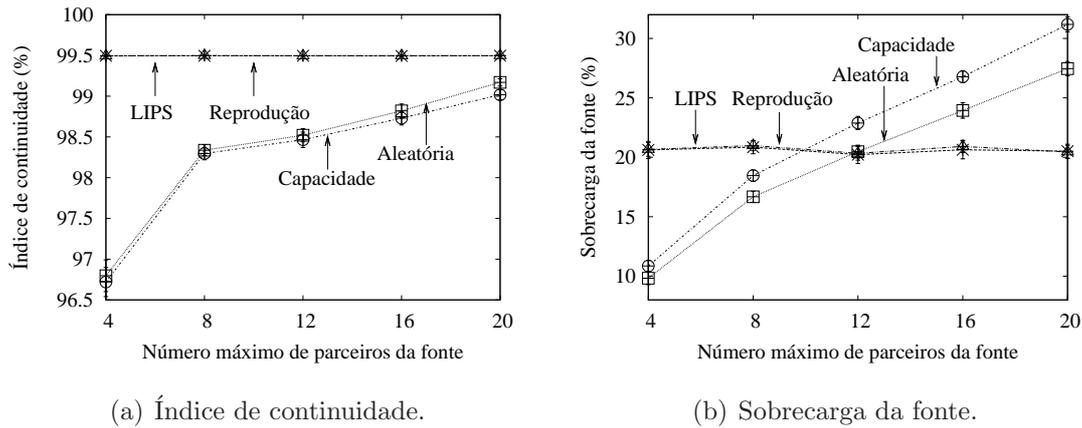


Figura 5.8: O impacto do número máximo de parceiros da fonte nos mecanismos de seleção de parceiros.

produção e, assim, reduz-se a probabilidade de um pedaço chegar atrasado. Esse efeito pode ser observado para todos os mecanismos na Figura 5.9(a). Pode-se observar também que, para todos os mecanismos, o número médio de pedaços de interesse por parceiro aumenta com o aumento do tamanho da janela. Porém, a razão entre o número médio de pedaços de interesse por parceiro e o tamanho da janela se mantém constante. Para o LIPS, esse valor é de aproximadamente 65%. Para a seleção aleatória, entretanto, é de 50%. Isso mostra que o LIPS é mais eficiente na seleção de parceiros, uma vez que com mais pedaços de interesse por parceiro, maior o número de fornecedores potenciais para um pedaço de vídeo.

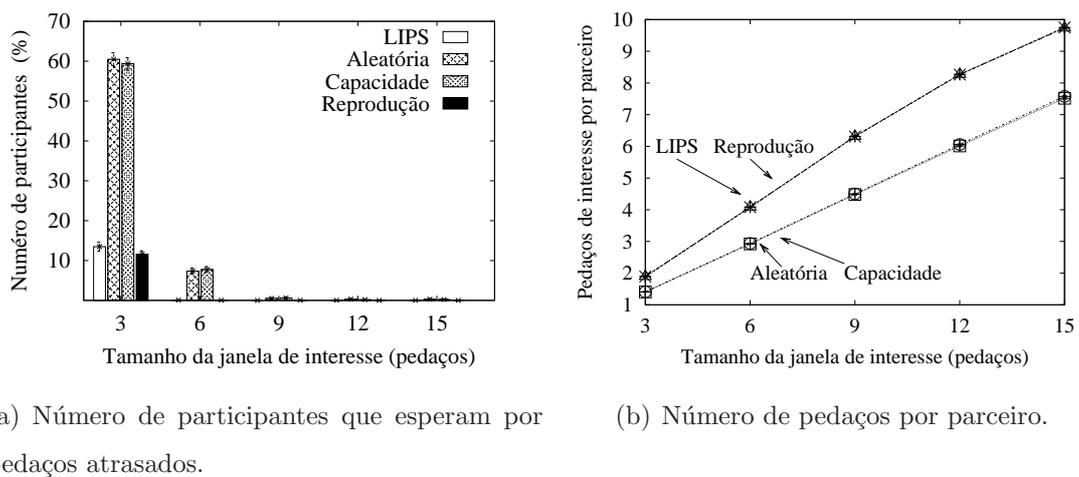


Figura 5.9: O impacto do tamanho da janela de interesse nos mecanismos de seleção de parceiros.

Parâmetros do LIPS

Dois parâmetros específicos influenciam o desempenho do LIPS: o fator de expansão R e o limiar de expansão A . Como visto na Seção 5.3, o anel implementado pelo LIPS é expandido de acordo com o produto αR , onde $\alpha = 1, 2, \dots, A$. Esse produto determina a diferença tolerável entre os tempos de vida de um participante e de seus candidatos a parceiros. Quando $\alpha = A$, tem-se a máxima diferença tolerável. Para avaliar o impacto do produto αR no desempenho do LIPS, assume-se que o limiar A é fixo e igual a 10 e que o parâmetro R pode assumir os seguintes valores: 5, 10 e 20 s. Esses valores são definidos de acordo com o tempo médio entre as chegadas dos participantes. Esse tempo é igual a 10 s para o vídeo de entretenimento e, assim, avalia-se o LIPS para um valor inferior e outro superior a esse tempo médio. O desempenho para o LIPS com diferentes valores de R é apresentado na Figura 5.10.

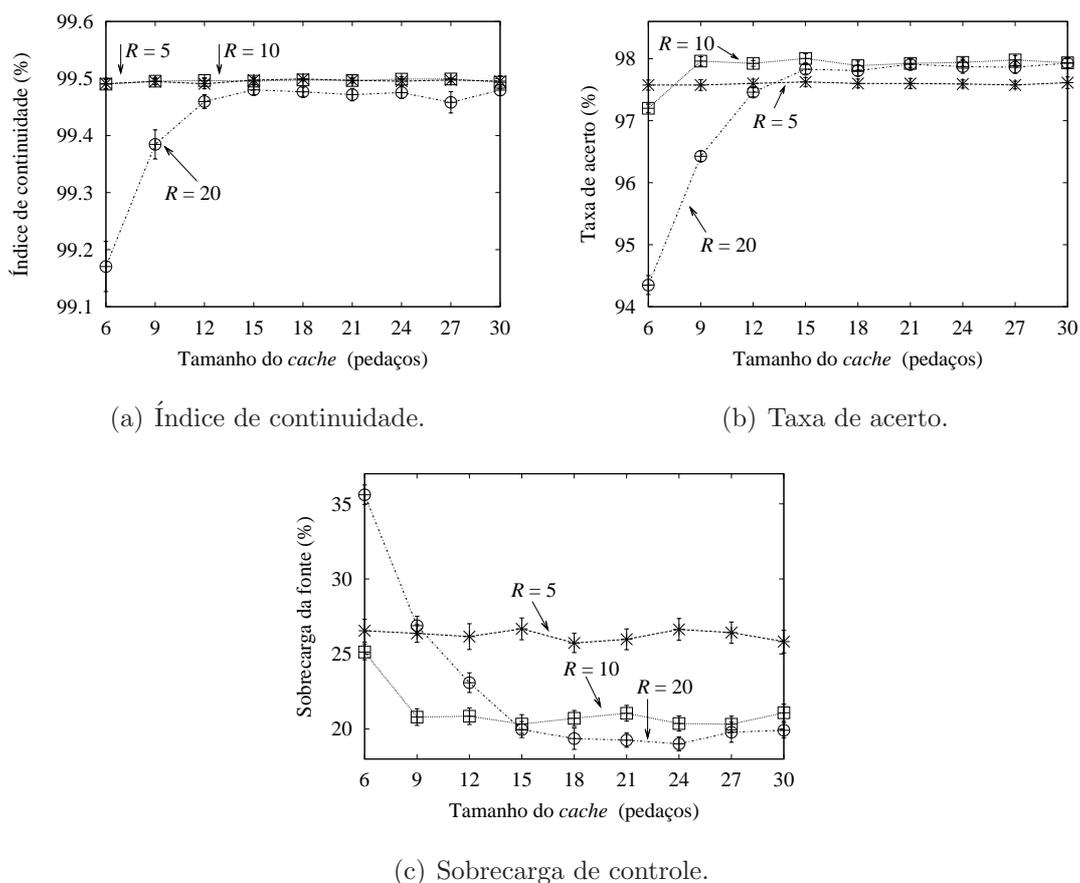


Figura 5.10: O impacto dos parâmetros do anel expansivo no LIPS.

Sabe-se que quanto maior o produto αR , maior a diferença tolerável entre os

tempos de vida de um participante e de seus parceiros. Dessa forma, a probabilidade de se escolher parceiros que não estejam assistindo ao mesmo trecho de vídeo aumenta. Conseqüentemente, o sistema se torna mais dependente do tamanho do *cache*, como mostram as Figura 5.10(a) e 5.10(b). O índice de continuidade para o LIPS com $R = 5$ e $R = 10$ se mantém constante, independentemente do tamanho do *cache*. Entretanto, quando $R = 20$ é necessário um *cache* capaz de armazenar pelo menos metade dos pedaços do vídeo, para que o índice de continuidade alcance o valor proporcionado pelo LIPS com limites de expansão menores.

A sobrecarga da fonte também é influenciada pela variação do produto αR , como ilustra a Figura 5.10(c). Quanto maior o produto αR , menor a probabilidade de se atingir o limiar A e, conseqüentemente, de se adicionar a fonte ao conjunto de parceiros. Sendo assim, quanto maior o valor de αR , menor a sobrecarga da fonte. Porém, nota-se que essa relação só é válida a partir de um determinado valor para o tamanho do *cache*. No cenário analisado, esse valor é igual a 50% dos pedaços do vídeo. Para tamanhos de *cache* menores do que esse, a relação se inverte. Para um *cache* de 6 pedaços, por exemplo, o LIPS com $R = 20$ apresenta a maior sobrecarga de controle. Como, nesse caso, o produto αR é maior, aumenta-se a probabilidade da diferença entre os tempos de vida da fonte e do participante estarem dentro do limite do anel.

Existe, portanto, um compromisso entre o valor do produto αR , o tamanho do *cache* e a sobrecarga da fonte para garantir a eficiência da seleção de parceiros e a continuidade de recepção do vídeo. Se o objetivo do sistema é economizar o espaço em disco exigido pelo *cache* de pedaços nos participantes, devem-se utilizar valores de R menores. O preço pago é o aumento da sobrecarga da fonte. Por outro lado, se o objetivo é reduzir a sobrecarga da fonte, devem-se usar valores de R maiores. Em contrapartida, exige-se mais espaço em disco nos participantes.

Capítulo 6

Conclusões

O uso de sistemas par-a-par é uma das soluções mais promissoras para aumentar a eficiência da distribuição de vídeo na Internet, principalmente, devido à escalabilidade desses sistemas. Hoje, o tráfego gerado por aplicações de distribuição de vídeo já corresponde a um grande percentual do tráfego total da Internet. Embora alguns sistemas par-a-par de vídeo já suportem milhares de usuários simultâneos, tais sistemas ainda apresentam uma série de desafios a serem suplantados para sua utilização na distribuição de vídeo. Dois desses desafios são abordados neste trabalho: a avaliação experimental de sistemas par-a-par de difusão de vídeo em redes sem fio e a seleção de parceiros em sistemas par-a-par de vídeo sob demanda.

Primeiramente, avaliou-se o impacto da mobilidade dos usuários no desempenho de alguns dos sistemas par-a-par de difusão de vídeo mais populares em uma rede sem fio experimental. Os experimentos reproduziram um cenário de mobilidade no qual um usuário caminha através de diferentes *gateways*. O usuário possui um computador portátil e está usando um dos aplicativos de distribuição de vídeo par-a-par para assistir a um dado canal à medida que anda pela área de cobertura da rede de testes. O desempenho dos sistemas par-a-par foi comparado também com um aplicativo cliente-servidor. Os resultados mostram que os sistemas par-a-par de vídeo se adaptam melhor ao cenário de mobilidade do que o aplicativo cliente-servidor. Uma única mudança de *gateway* é suficiente para degradar o desempenho do cliente servidor. Além disso, mostra-se que a adaptação dos sistemas par-a-par é diferente dependendo do tipo de protocolo de transporte usado. Os sistemas que usam o UDP se adaptam melhor à mobilidade dos usuários na rede sem fio. No

entanto, apesar de serem menos eficientes do que os sistemas que usam o UDP, os sistemas que usam o TCP também conseguem se adaptar ao cenário de mobilidade, pois utilizam conexões de curta duração, que é característica comum dos sistemas par-a-par, pois um nó recebe o vídeo de múltiplos pares. Conclui-se então que a arquitetura distribuída dos sistemas par-a-par é o fator que mais contribui para adaptação de tais sistemas em redes sem fio com usuários móveis.

Neste trabalho, foi proposto também um mecanismo de seleção de parceiros específico para sistemas par-a-par de vídeo sob demanda. A ideia básica do mecanismo LIPS é formar parcerias entre participantes com tempos de vida próximos e, conseqüentemente, aumentar a disponibilidade de pedaços de interesse entre os parceiros. O tempo de vida de um participante é definido como o intervalo de tempo entre a chegada desse participante ao sistema e o tempo atual, independentemente das interações que ele tenha feito. O desempenho do LIPS foi avaliado e comparado com outros três mecanismos: a seleção aleatória, a seleção baseada na capacidade dos participantes e a seleção baseada no tempo de reprodução. Através de simulação, os quatro mecanismos de seleção de parceiros foram avaliados para diferentes cenários, considerando diferentes padrões de comportamento dos participantes. Foram analisados três tipos de vídeo: um vídeo de entretenimento, um vídeo educacional e um filme. Os parâmetros para caracterizá-los foram obtidos através de cargas de trabalho reais [78, 2]. Os vídeos, além do padrão de interatividade, possuem durações e popularidades diferentes. As métricas usadas na avaliação indicam a eficiência dos parceiros selecionados por cada mecanismo e mostram o impacto de cada mecanismo na continuidade de reprodução do vídeo, um dos parâmetros mais críticos para se determinar a satisfação dos usuários.

Os resultados mostram que o mecanismo proposto LIPS é o que seleciona os parceiros mais eficientes e, por isso, garante uma alta continuidade de reprodução do vídeo nos participantes para todos os cenários analisados. O LIPS proporciona uma taxa de acerto mais alta e um número médio de pedaços de interesse por parceiro maior se comparado à seleção aleatória e à seleção baseada na capacidade dos participantes, independentemente do tamanho do *cache*. Para o vídeo de entretenimento, o mais popular e de menor duração, a taxa de acerto máxima e o número máximo de pedaços de interesse por parceiro com o LIPS são, respectivamente, 97,9% e 4,09.

Com a seleção aleatória e a seleção baseada na capacidade, esses valores são iguais a 95,4% e 3,6. Além disso, o LIPS, com poucos pedaços de vídeo armazenados em disco, provê, por exemplo, taxas de acerto mais altas do que as máximas obtidas com esses dois mecanismos. Por exemplo, para o filme, o vídeo de maior duração e com saltos longos, o LIPS supera a taxa máxima de acerto da seleção aleatória exigindo um tamanho de *cache* cerca de 250 MB menor. Os resultados mostram também que o LIPS é tão eficiente quanto a seleção baseada no tempo de reprodução nos cenários analisados, com a vantagem de não introduzir nenhuma mensagem de controle adicional, em virtude das interações dos participantes. Apesar do tempo de reprodução ser uma métrica mais fiel para se determinar o atual intervalo de pedaços de interesse do vídeo, mostrou-se que a sobrecarga de controle pode ser muito alta, podendo ser da ordem de dezenas de milhares de mensagens dependendo do número de participantes e de suas interações com a reprodução do vídeo. Por isso, a escolha do tempo de vida como métrica de seleção se justifica. Quanto maior a eficiência dos parceiros selecionados, menor a probabilidade de se esperar por pedaços para continuar a reprodução do vídeo sem interrupções. O LIPS é o mecanismo que proporciona os mais altos índices de continuidade. Com o LIPS, o índice de continuidade da reprodução para os vídeos de entretenimento e educacional é superior a 99%, independentemente da quantidade de pedaços de vídeo que se pode armazenar no *cache*. Para o filme, esse índice é superior a 96%. Além disso, o mecanismo proposto reduz o número de participantes que esperam por pedaços de vídeo atrasados. Para o filme, o vídeo com o padrão de interatividade mais agressivo, essa redução é de aproximadamente 35% em comparação com a seleção aleatória, dependendo do tamanho do *cache*.

A explicação para o melhor desempenho do LIPS vem do fato de que o mecanismo proposto seleciona os parceiros de acordo com o tempo em que o participante está no sistema. Esse é um dos principais fatores que afetam a probabilidade de um participante recém-chegado ao sistema selecionar parceiros com pedaços de interesse. Com o LIPS, essa probabilidade passa a depender apenas da interatividade dos participantes e, como demonstram os resultados, a escolha de parceiros com tempos de entrada muito diferentes afeta mais a disponibilidade de pedaços de interesse entre os parceiros do que a interatividade dos participantes para os diferentes padrões

analisados.

O impacto de alguns parâmetros dos sistemas par-a-par de vídeo sob demanda também foi analisado, bem como os parâmetros do mecanismo proposto que definem o anel expansivo. Foram analisados o intervalo de atualização dos parceiros, o número máximo de parceiros por participante, o número máximo de parceiros da fonte e o tamanho da janela de interesse. O vídeo de entretenimento foi considerado nessa análise. Mostrou-se que os mecanismos de seleção aleatória e de seleção baseada na capacidade de *upload* são mais influenciados com o aumento do intervalo de atualização, pois quanto maior o intervalo de atualização, maior o tempo para se substituir um parceiro pouco eficiente. Verificou-se também que, a partir de um determinado valor, que varia de acordo com o mecanismo de seleção, a continuidade de reprodução do vídeo é pouco influenciada pelo número máximo de parceiros por participante. Outro parâmetro avaliado foi o número máximo de parceiros da fonte, que determina o número de participantes que podem receber pedaços de vídeo diretamente da fonte. Observa-se que, em alguns cenários, garante-se a eficiência do LIPS em troca do aumento da sobrecarga da fonte. Entretanto, para um número máximo de parceiros da fonte maior ou igual a 12, a sobrecarga da fonte imposta pelos mecanismos de seleção aleatória e de seleção baseada na capacidade dos participantes é maior do que a do LIPS e, mesmo assim, o índice de continuidade do mecanismo proposto se mantém superior. Isso demonstra que a seleção de parceiros com mais pedaços de interesse disponíveis é o fator que mais contribui para o melhor desempenho do LIPS e não o aumento da sobrecarga da fonte. Verificou-se também que, para todos os mecanismos, à medida que o tamanho da janela de interesse aumenta, a probabilidade de um pedaço de vídeo ser recebido após o seu tempo de reprodução diminui. Isso ocorre porque se aumenta o intervalo de tempo entre a recepção de um pedaço de vídeo e a sua reprodução, uma vez que o tamanho da janela de interesse é calculado em função do tamanho do *buffer* de reprodução. Por fim, os parâmetros do mecanismo proposto foram analisados. Mostrou-se que existe um compromisso entre o limite do anel expansivo definido pelo LIPS, o tamanho do *cache* e a sobrecarga da fonte para garantir a eficiência da seleção de parceiros e a continuidade de reprodução do vídeo.

Um dos possíveis trabalhos futuros é analisar o desempenho dos mecanismos de

seleção de parceiros considerando outros mecanismos de seleção de pedaços diferentes da janela deslizante. A proposta de Hoffman Filho *et al.* [81] é usar duas janelas diferentes: uma janela de reprodução e uma janela de previsão. A janela de reprodução é definida com base no ponto atual de reprodução do vídeo e a janela de previsão é estimada de acordo com o comportamento interativo apresentado pelos participantes. Existem também mecanismos que não definem janelas de pedaços de interesse [82, 83]. Tais mecanismos classificam todos os pedaços de vídeo de acordo com o tempo que falta até a sua reprodução. Formam-se, assim, dois conjuntos de pedaços: um conjunto de maior prioridade e outro de menor prioridade. O conjunto de maior prioridade é formado pelos n pedaços mais próximos de sua reprodução. Os demais pedaços formam o conjunto de menor prioridade. Para cada conjunto é atribuída uma probabilidade e , assim, a cada nova requisição, o conjunto do qual dever ser solicitado o pedaço de vídeo é determinado. Os pedaços de vídeo pertencentes a cada conjunto podem ser requisitados sequencialmente, de acordo com o tempo restante para a sua reprodução, ou de acordo com a sua popularidade. Essa é a diferença entre as propostas de Vlavianos *et al.* [82] e Zhou *et al.* [83]. Outro trabalho que poderá ser realizado é a avaliação do impacto de algoritmos de substituição de *cache* [66] na seleção de parceiros em sistemas de vídeo sob demanda. Nesse trabalho, considerando que L e W são, respectivamente, o tamanho do *cache* e de janela de interesse, define-se que os pedaços de vídeo anteriores aos pedaços no intervalo $L - W$, iniciado no primeiro pedaço da janela de interesse, podem ser removidos. Uma das possíveis modificações é estimar o comportamento interativo de um participante e com base nessa informação atribuir uma probabilidade a cada pedaço de vídeo armazenado no *cache*, como feito na seleção de pedaços. De acordo com essa probabilidade, determinam-se quais pedaços devem ser removidos do *cache*. Pode-se analisar também o impacto do armazenamento de pedaços de múltiplos vídeos no *cache* [52]. Por fim, existem áreas de interesse relacionadas aos sistemas par-a-par que não foram abordadas neste trabalho. Os sistemas par-a-par terão um papel fundamental, por exemplo, na definição da arquitetura da Internet do Futuro. Esse é um tema desafiador e de grande interesse atualmente por parte da comunidade acadêmica.

Referências Bibliográficas

- [1] HUANG, C., LI, J., ROSS, K. W., “Can Internet Video-on-Demand be Profitable?” Em: *ACM SIGCOMM*, pp. 133–144, Ago. 2007.
- [2] HUANG, Y., FU, T. Z., CHIU, D.-M., et al., “Challenges, design and analysis of a large-scale P2P-VoD system”. Em: *ACM SIGCOMM*, pp. 375–388, Ago. 2008.
- [3] LIU, J., RAO, S. G., LI, B., et al., “Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast”, *Proceedings of the IEEE*, v. 96, n. 1, pp. 11–24, Jan. 2008.
- [4] IDG NOW!, “Partida entre Portugal e França bate novo recorde de transmissão online”, <http://idgnow.uol.com.br/internet/2006/07/12/idgnoticia.2006-07-12.6926355736/>, Julho 2006, Acessado em novembro de 2009.
- [5] SENTINELLI, A., MARFIA, G., GERLA, M., et al., “Will IPTV ride the peer-to-peer stream?” *IEEE Communications Magazine*, v. 45, n. 6, pp. 86–92, 2007.
- [6] REJAIE, R., “Anyone can Broadcast Video over the Internet”, *Communications of the ACM*, v. 49, n. 11, pp. 55–58, Nov. 2006.
- [7] MORAES, I. M., CAMPISTA, M. E. M., DUARTE, J. L., et al., “On the Impact of User Mobility on Peer-to-Peer Video Streaming”, *IEEE Wireless Communications Magazine*, v. 15, n. 6, pp. 54–62, 2008.
- [8] MORAES, I. M., CAMPISTA, M. E. M., MOREIRA, M. D. D., et al., “Distribuição de Vídeo sobre Redes Par-a-Par: Arquiteturas, Mecanismos e

- Desafios”. Em: *Minicursos do Simpósio Brasileiro de Redes de Computadores - SBRC*, pp. 115–171, Maio 2008.
- [9] TANG, Y., LUO, J.-G., ZHANG, Q., et al., “Deploying P2P networks for large-scale live video-streaming service”, *IEEE Communications Magazine*, v. 45, n. 6, pp. 100–106, 2007.
- [10] SILVERSTON, T., FOURMAUX, O., “Measuring P2P IPTV Systems”. Em: *International Workshop on Network and Operating System Support for Digital Audio and Video - NOSSDAV*, Junho 2007.
- [11] RODRIGUEZ, P., TAN, S.-M., GKANTSIDIS, C., “On the feasibility of commercial, legal P2P content distribution”, *ACM SIGCOMM Computer Communication Review*, v. 36, n. 1, pp. 75–78, Jan. 2006.
- [12] VRATONJIC, N., GUPTA, P., KNEZEVIC, N., et al., “Enabling DVD-like Features in P2P Video-on-demand Systems”. Em: *Peer-to-Peer Streaming and IP-TV Workshop - P2P-TV*, pp. 329–334, Ago. 2007.
- [13] ANNAPUREDDY, S., GUHA, S., GKANTSIDIS, C., et al., “Exploring VoD in P2P Swarming Systems”. Em: *IEEE INFOCOM*, pp. 2571–2575, Maio 2007.
- [14] ZHANG, X., LIU, J., LI, B., et al., “CoolStreaming/DONet: A Data-driven Overlay Network for Peer-to-Peer Live Media Streaming”. Em: *IEEE INFOCOM*, pp. 2102–2111, Março 2005.
- [15] MORAES, I. M., BICUDO, M. D. D., CARDOSO, K. V., et al., “Desenvolvimento de um Ambiente de Testes com suporte à Qualidade de Serviço para a Transmissão de Vídeo Digital”. Em: *IV Workshop da RNP2*, Maio 2003.
- [16] DEERING, S., “Host Extensions for IP Multicasting”, IETF Network Working Group RFC 1112, Ago. 1989.
- [17] COSTA, L. H. M. K., DUARTE, O. C. M. B., “Roteamento Multicast na Internet”. Em: *Minicursos do Simpósio Brasileiro de Redes de Computadores - SBRC*, Maio 2003.

- [18] HOLBROOK, H., “Source-Specific Multicast for IP”, IETF Network Working Group RFC 4607, Ago. 2006.
- [19] BHATTACHARYYA, S., “An Overview of Source-Specific Multicast (SSM)”, IETF Network Working Group RFC 3569, Julho 2003.
- [20] HOLBROOK, H. W., CHERITON, D. R., “IP Multicast Channels: EXPRESS Support for Large-scale Single-source Applications”. Em: *ACM SIGCOMM*, pp. 65–78, Set. 1999.
- [21] COSTA, L. H. M. K., FDIDA, S., DUARTE, O. C. M. B., “Incremental Service Deployment Using the Hop By Hop Multicast Routing Protocol”, *IEEE/ACM Transactions on Networking*, v. 14, n. 3, pp. 543–556, Junho 2006.
- [22] EL-SAYED, A., ROCA, V., MATHY, L., “A Survey of Proposals for an Alternative Group Communication Service”, *IEEE Network*, v. 17, n. 1, pp. 46–51, Jan. 2003.
- [23] LAO, L., CUI, J.-H., GERLA, M., et al., “A Comparative Study of Multicast Protocols: Top, Bottom, or In the Middle?” Em: *IEEE INFOCOM*, pp. 13–17, Março 2005.
- [24] BANERJEE, S., BHATTACHARJEE, B., *A Comparative Study of Application Layer Multicast Protocols*, Rel. técnico, University of Wisconsin-Madison, 2005.
- [25] ZHUANG, S., ZHAO, B., JOSEPH, A., et al., “Bayeux: An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination”. Em: *International Workshop on Network and Operating System Support for Digital Audio and Video - NOSSDAV*, pp. 11–20, Junho 2001.
- [26] GUO, L., CHEN, S., REN, S., et al., “PROP: a scalable and reliable P2P assisted proxy streaming system”. Em: *International Conference on Distributed Computing Systems - ICDCS*, pp. 778–786, 2004.

- [27] HEFEEDA, M. M., BHARGAVA, B. K., YAU, D. K. Y., “A hybrid architecture for cost-effective on-demand media streaming”, *Computer Networks*, v. 44, n. 3, pp. 353–382, Fev. 2004.
- [28] CHU, Y.-H., RAO, S. G., SESHAN, S., et al., “A Case for End System Multicast”, *IEEE Journal on Selected Areas in Communications*, v. 20, n. 8, pp. 1456–1471, Out. 2002.
- [29] TRAN, D. A., HUA, K. A., DO, T. T., “A Peer-to-Peer Architecture for Media Streaming”, *IEEE Journal on Selected Areas in Communications*, v. 22, n. 1, pp. 121–133, Jan. 2004.
- [30] BANERJEE, S., BHATTACHARJEE, B., KOMMAREDDY, C., “Scalable Application level multicast”. Em: *ACM SIGCOMM*, pp. 205–217, Ago. 2002.
- [31] LIU, J., LI, B., ZHANG, Y.-Q., “Adaptive Video Multicast over the Internet”, *IEEE Multimedia*, v. 10, n. 1, pp. 22–33, Jan. 2003.
- [32] MCCANNE, S., JACOBSON, V., VETTERLI, M., “Receiver-driven layered multicast”. Em: *ACM SIGCOMM*, pp. 117–130, Out. 1996.
- [33] VICKERS, B., DE ALBUQUERQUE, C. V. N., SUDA, T., “Source-Adaptive Multi-Layered Multicast Algorithms for Real-Time Video Distribution”, *IEEE/ACM Transactions on Networking*, v. 8, n. 6, pp. 720–733, Fev. 2000.
- [34] WANG, Y., REIBMAN, A. R., LIN, S., “Multiple Description Coding for Video Delivery”, *Proceedings of the IEEE*, v. 93, n. 1, pp. 57–70, Jan. 2005.
- [35] DE ALBUQUERQUE, C. V. N., PROENÇA, T., OLIVEIRA, E., “TVoIP: TV sobre IP Arquiteturas para Transmissão em Larga Escala”. Em: *Minicursos do Simpósio Brasileiro de Redes de Computadores - SBRC*, Maio 2006.
- [36] FREEDMAN, M. J., FREUDENTHAL, E., MAZIÈRES, D., et al., “The Coral Content Distribution Network”, <http://www.coralcdn.org>, 2008, Acessado em novembro de 2009.

- [37] PAI, V., PETERSON, L., PARK, K., “CoDeeN - A Content Distribution Network for PlanetLab”, <http://codeen.cs.princeton.edu>, 2008, Acessado em novembro 2009.
- [38] LIMELIGHT NETWORKS, “High Performance Content Delivery Network for Digital Media: Limelight Networks”, <http://www.limelightnetworks.com>, 2008, Acessado em novembro de 2009.
- [39] SEVERN STREAM MEDIA NETWORKS, “Severn Stream Media Networks - The Intelligent Content Delivery Network For Digital Multimedia”, <http://www.severnstream.com>, 2007, Acessado em novembro de 2009.
- [40] CDNETWORKS CO., LTD., “CDNetworks :: Global CDN Service Leader”, <http://www.us.cdnetworks.com>, 2004, Acessado em novembro de 2009.
- [41] AKAMAI TECHNOLOGIES, “Akamai: The Leader in Web Application Acceleration and Performance Management, Streaming Media Services and Content Delivery”, <http://www.akamai.com>, 2008, Acessado em novembro de 2009.
- [42] AKAMAI TECHNOLOGIES, “Akamai helps MSN deliver more than 65 million video streams around the world for record-breaking Live Earth event”, http://www.akamai.com/html/customers/case_study_live-earth.html, 2008, Acessado em novembro de 2009.
- [43] CASTRO, M., DRUSCHEL, P., KERMARREC, A.-M., et al., “SplitStream: High-Bandwidth Multicast in Cooperative Environments”. Em: *ACM Symposium on Operating Systems Principle - SOSOP*, pp. 298–313, 2003.
- [44] DAN, G., FODOR, V., CHATZIDROSSOS, I., “On the Performance of Multiple-Tree-Based Peer-to-Peer Live Streaming”. Em: *IEEE INFOCOM*, pp. 2556–2560, Maio 2007.
- [45] MAGHAREI, N., REJAIE, R., GUO, Y., “Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches”. Em: *IEEE INFOCOM*, pp. 1424–1432, Maio 2007.

- [46] HALES, D., PATARIN, S., “Computational sociology for systems “in the wild” : the case of BitTorrent”, *IEEE Distributed Systems Online*, v. 6, n. 7, Julho 2005.
- [47] ZHANG, M., TANG, Y., ZHAO, L., et al., “Gridmedia: A Multi-Sender Based Peer-to-Peer Multicast System for Video Streaming”. Em: *IEEE International Conference on Multimedia and Expo - ICME*, pp. 614–617, Julho 2005.
- [48] LI, B., XIE, S., KEUNG, G. Y., et al., “An Empirical Study of the Coolstreaming+ System”, *IEEE Journal on Selected Areas in Communications*, v. 25, n. 9, pp. 1627–1639, Nov. 2007.
- [49] COOLSTREAMING, “Coolstreaming - Broadcast your TV”, <http://www.coolstreaming.us>, 2008, Acessado em novembro de 2008.
- [50] GANESH, A. J., KERMARREC, A.-M., MASSOULI, L., “Peer-to-peer membership management for gossip-based protocols”, *IEEE Transactions on Computers*, v. 52, n. 2, pp. 139–149, Fev. 2003.
- [51] LENSTRA, J. K., SHMOYS, D. B., TARDOS, É., “Approximation algorithms for scheduling unrelated parallel machines”, *Mathematical Programming*, v. 46, n. 3, pp. 259–271, Jan. 1990.
- [52] CHENG, B., STEIN, L., JIN, H., et al., “GridCast: Improving peer sharing for P2P VoD”, *ACM Transactions on Multimedia Computing, Communications, and Applications*, v. 4, n. 4, pp. 26–31, Out. 2008.
- [53] CAMPISTA, M. E. M., PASSOS, D. G., ESPOSITO, P. M., et al., “Routing Metrics and Protocols for Wireless Mesh Networks”, *IEEE Network*, v. 22, n. 1, pp. 6–12, Jan. 2008.
- [54] WU, C., LI, B., ZHAO, S., “Characterizing Peer-to-Peer Streaming Flows”, *IEEE Journal on Selected Areas in Communications*, v. 25, n. 9, pp. 1612–1625, Nov. 2007.

- [55] XIE, S., KEUNG, G. Y., LI, B., “A Measurement of a large-scale Peer-to-Peer Live Video Streaming System”. Em: *International Conference on Parallel Processing Workshops - ICPPW*, pp. 57–57, Set. 2007.
- [56] HE, Y., LEE, I., GU1, X., et al., “Centralized Peer-to-Peer Video Streaming Over Hybrid Wireless Network”. Em: *IEEE International Conference on Multimedia and Expo - ICME*, pp. 550–553, Julho 2005.
- [57] SHIHAB, E., LIN, C., FENGDAN, W., et al., “Wireless Mesh Networks for In-Home IPTV Distribution”, *IEEE Network*, v. 22, n. 1, pp. 52–57, Jan. 2008.
- [58] SHIWEN, M., HOU, Y., SHERALI, H. D., et al., “Multimedia-Centric Routing for Multiple Description Video in Wireless Mesh Networks”, *IEEE Network*, v. 22, n. 1, pp. 19–24, Jan. 2008.
- [59] VELLOSO, P. B., CUNHA, D. O., AMODEI JR, A., et al., “Redes Domílicares: Princípios e Desafios das Tecnologias sem Novos Fios”. Em: *Minicursos do Simpósio Brasileiro de Redes de Computadores - SBRC*, pp. 221–269, Maio 2004.
- [60] CAMPISTA, M. E. M., MORAES, I. M., ESPOSITO, P. M., et al., “The Ad Hoc Return Channel: a Low-Cost Solution for Brazilian Interactive Digital TV”, *IEEE Communications Magazine*, v. 45, n. 1, pp. 136–143, Jan. 2007.
- [61] AMIR, Y., DANILOV, C., HILSDALE, M., et al., “Fast Handoff for Seamless Wireless Mesh Networks”. Em: *ACM Mobisys*, pp. 83–95, Junho 2006.
- [62] ROSENBERG, J., WEINBERGER, J., HUITEMA, C., et al., “STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)”, IETF Network Working Group RFC 3489, Março 2003.
- [63] GUHA, S., TAKEDA, Y., FRANCIS, P., “NUTSS: A SIP-based Approach to UDP and TCP Network Connectivity”. Em: *ACM SIGCOMM*, Ago. 2004.

- [64] UPNP FORUM, “UPnP Forum [online]”, <http://www.upnp.org>, 2008, Acessado em novembro de 2009.
- [65] HANBALL, A. A., ALTMAN, E., NAIN, P., “A Survey of TCP over Ad Hoc Networks”, *IEEE Communications Surveys & Tutorials*, v. 7, n. 3, pp. 22–36, 2005.
- [66] WU, J., LI, B., “Keep Cache Replacement Simple in Peer-Assisted VoD Systems”. Em: *IEEE INFOCOM Mini-Conference*, Abril 2009.
- [67] LIU, Z., SHEN, Y., PANWAR, S., et al., “On Scalability of Proximity-Aware Peer-to-Peer Streaming”. Em: *IEEE INFOCOM*, pp. 2561–2565, Maio 2007.
- [68] LIANG, J., NAHRSTEDT, K., “RandPeer: Membership Management for QoS Sensitive Peer-to-Peer Applications”. Em: *IEEE INFOCOM*, pp. 1–10, Abril 2006.
- [69] CUI, Y., DAI, L., XUE, Y., “Optimizing P2P Streaming Throughput under Peer Churning”. Em: *IEEE GLOBECOM*, pp. 231–235, Nov. 2007.
- [70] REJAIE, R., STAFFORD, S., “A framework for architecting peer-to-peer receiver-driven overlays”. Em: *International Workshop on Network and Operating System Support for Digital Audio and Video - NOSSDAV*, pp. 42–47, Junho 2004.
- [71] SHARMA, A., BESTAVAROS, A., MATTA, I., “dPAM: a distributed prefetching protocol for scalable asynchronous multicast in P2P systems”. Em: *IEEE INFOCOM*, pp. 1139–1150, Março 2005.
- [72] GUO, Y., SUH, K., KUROSE, J., et al., “P2Cast: peer-to-peer patching scheme for VoD service”. Em: *International World Wide Web Conference - WWW*, pp. 301–309, 2003.
- [73] SHAH, P., PÂRIS, J. F., “Peer-to-Peer Multimedia Streaming Using BitTorrent”. Em: *IEEE International Performance, Computing, and Communications Conference - IPCCC*, pp. 340–347, 2007.

- [74] MORAES, I. M., DUARTE, O. C. M. B., “Um Mecanismo de Seleção de Pares Baseado no Tempo de Vida dos Participantes em Sistemas de Vídeo Par-a-Par”. Em: *Simpósio Brasileiro de Redes de Computadores - SBRC*, pp. 715–728, Maio 2009.
- [75] MORAES, I. M., DUARTE, O. C. M. B., “A Lifetime-Based Peer Selection Mechanism for Peer-to-Peer Video-on-Demand Systems”. Em: *IEEE International Communications Conference - ICC*, Maio 2010.
- [76] SILVA, T., MOTA, V., VALADAO, E., et al., “Caracterização do Comportamento dos Espectadores em Transmissões de Vídeo ao Vivo Geradas por Usuários”. Em: *Simpósio Brasileiro de Redes de Computadores - SBRC*, pp. 1–14, Maio 2009.
- [77] VELOSO, E., ALMEIDA, V., MEIRA JR., W., et al., “A hierarchical characterization of a live streaming media workload”, *IEEE/ACM Transactions on Networking*, v. 14, n. 1, pp. 133–146, 2006.
- [78] COSTA, C., CUNHA, I., BORGES, A., et al., “Analyzing Client Interactivity in Streaming Media”. Em: *International World Wide Web Conference - WWW*, pp. 534–543, Ago. 2004.
- [79] VIELMOND, C. C. L. B., LEÃO, R. M. M., DE SOUZA E SILVA, E., “Um modelo HMM hierárquico para usuários interativos acessando um servidor multimídia”. Em: *Simpósio Brasileiro de Redes de Computadores - SBRC*, pp. 469–482, Maio 2007.
- [80] CHESIRE, M., WOLMAN, A., VOELKER, G. M., et al., “Measurement and Analysis of a Streaming Media Workload”. Em: *USENIX Symposium on Internet Technology and Systems - USITS*, pp. 1–12, Março 2001.
- [81] HOFFMANN FILHO, L. J., DA SILVA RODRIGUES, C. K., LEÃO, R. M. M., “Acesso interativo para aplicações P2P de streaming de vídeo”. Em: *Simpósio Brasileiro de Redes de Computadores - SBRC*, pp. 599–612, Maio 2009.

- [82] VLAVIANOS, A., ILIOFOTOU, M., FALOUTSOS, M., “BiToS: Enhancing BitTorrent for Supporting Streaming Applications”. Em: *IEEE Global Internet Symposium - GIS*, pp. 1–6, Abril 2006.
- [83] ZHOU, Y., CHIU, D. M., LUIS, J., “A Simple Model for Analyzing P2P Streaming Protocols”. Em: *IEEE International Conference on Network Protocols - ICNP*, pp. 226–235, Out. 2007.