TRANSMISSÃO DE MÍDIA CONTÍNUA AO VIVO EM P2P: MODELAGEM, CARACTERIZAÇÃO E IMPLEMENTAÇÃO DE MECANISMOS DE RESILIÊNCIA A ATAQUES.

ALEX BORGES VIEIRA

TRANSMISSÃO DE MÍDIA CONTÍNUA AO VIVO EM P2P: MODELAGEM, CARACTERIZAÇÃO E IMPLEMENTAÇÃO DE MECANISMOS DE RESILIÊNCIA A ATAQUES.

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

Orientador: Sérgio Vale Aguiar Campos

Belo Horizonte

26 de março de 2010

Vieira, Alex Borges
V658t Transmissão de Mídia Contínua Ao Vivo em P2P: Modelagem, Caracterização e Implementação de Mecanismos de Resiliência a Ataques. / Alex Borges Vieira. — Belo Horizonte, 2010
xxvi, 165 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas Gerais
Orientador: Sérgio Vale Aguiar Campos

1. P2P. 2. IPTV. 3. Segurança. I. Título.

CDU 519.6*22

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

P2P Live streaming: Modelagem, caracterização e implementação de mecanismos de resiliencia a ataques

ALEX BORGES VIEIRA

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. SERGIO VALE AGUIAR CAMPOS - Orientador
Departamento de Ciência da Computação - UFMG

PROF. ELIAS PROCÓPIO DUARTE JÚNIOR
Departamento de Informática - UFPR

PROFA. RAQUEL APARECIDA DE FREITAS MINI
Departamento de Ciência da Computação - PUC-MG

PROF. ANTONIO ALFREDO FERREIRA LOUREIRO
Departamento de Ciência da Computação - UFMG

PROFA. JUSSARA MARQUES DE ALMEIDA
Departamento de Ciência da Computação - UFMG

PROF. RENATO ANTÔNIO CELSO FERREIRA
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 26 de março de 2010.

Agradecimentos

Agradeço a todos que participaram dessa minha conquista.... de forma especial, agradeço à minha família, aos meus pais e meu orientador. Além desses que me apioaram incondicionalmente, nada seria possível se eu não tivesse sido acolhido por uma família de amigos: o Luiz Filipe, o Marcos Augusto e os pais deles.

“e no fim; tudo deu certo.”
()

Resumo

Aplicações de mídia contínua em P2P estão se tornando mais populares a cada dia. Em contraste com o modelo tradicional cliente-servidor, o modelo P2P transpassa problemas como escalabilidade e necessidade de recursos poderosos, concentrados em um único ponto. Em contrapartida, as redes P2P podem ser alvos de ataques e comportamentos oportunistas.

Este trabalho apresenta um modelo descentralizado de reputação para combater ataques em sistemas de mídia contínua P2P. Em particular, são concentrados esforços para combater ataques de poluição. Nesse ataque, os participantes maliciosos alteram ou forjam os dados da mídia contínua na tentativa de disseminar conteúdo indesejado aos demais participantes do sistema.

Os resultados incluem uma modelagem de sistemas P2P e uma verificação formal dos danos causados por ataques de poluição. Nesse sentido, há uma evidencia que a escolha dos protocolos e algoritmos do sistema P2P influenciam no impacto causado por um ataque. Incluem também uma caracterização dos usuários de sistemas de transmissão ao vivo de mídia contínua em P2P. A partir dessa caracterização é possível compreender a dinâmica dos participantes desses sistemas e assim avaliar o impacto de seu comportamento tanto na escalabilidade do sistema, quanto na qualidade da mídia recebida. Finalmente, é desenvolvido um sistema de defesa contra ataques de poluição. Os resultados mostram que o novo mecanismo implementado bloqueia um ataque com uma sobrecarga inferior a 2% da banda de rede necessária para a transmissão, enquanto a solução indicada pela literatura necessita de pelo menos 100% de banda adicional. Além disso, o novo mecanismo é eficaz mesmo com conluio dos participantes e, nesse caso, o pico de retransmissão é inferior a 30%, enquanto as soluções tradicionais necessitam de até 100% de banda adicional. Finalmente, o novo mecanismo permite reabilitação dos participantes que geraram conteúdo indevido e provê medidas de incentivo para o compartilhamento dos recursos na rede P2P.

Palavras-chave: P2P, IPTV, Segurança, Ataques, Reputação, Ataque de Poluição.

Abstract

P2P live streaming applications are becoming more popular each day. In contrast to the client-server model, the P2P model overcomes problems like system scalability and need of powerful resources in a single point. On the other hand, P2P networks may suffer with attacks and opportunistic behavior.

In this work we present a decentralized reputation system to fight attacks in P2P live streaming networks. In particular, we focus on pollution attacks, where a malicious peer alters or forges the streaming data, aiming to disseminate undesirable content.

Our results include a formal model to check pollution attacks damages to P2P live streaming systems. We show that protocol and algorithms influence the impact caused by an attack. We also characterize the client behavior in P2P live streaming systems. Thus, we can understand client dynamics and evaluate the impact that it causes on system scalability and media quality. Finally we implement a reputation mechanism to fight pollution attacks. Our results show that the new mechanism can block a pollution attack with an overhead lower than 2% of network bandwidth, while the solution recommend needs at least 100%. Moreover, the new mechanism can lead with collusion attacks. In this case, it needs a peak less than 30%, while traditional approach needs at least 100% more bandwidth. Finally, allows peers rehabilitation and provides incentive mechanisms.

Keywords: P2P, IPTV, Security, Attacks, Reputation, Data Pollution Attack.

Resumo Estendido

As aplicações de distribuição de mídia contínua (áudio e vídeo) são uma das aplicações mais comuns na Internet. Para ilustrar este cenário promissor e efervescente para aplicações de distribuição de vídeo e áudio, alguns relatórios mostram que o *Youtube* - um dos sistemas mais populares para publicação e distribuição de vídeo armazenado - apresenta importantes marcas de visita e tráfego na rede. Há relatórios que mostram que ele (o Youtube) receba cerca de 20% das visitas globais de usuários da Internet por dia(www.alexa.com - 2009). Outros apontam que o *Youtube* é responsável por 10% de todo o tráfego de Internet da América do Norte (www.ellacoya.com - 2008).

Entretanto, o envio de mídia contínua ao vivo na Internet necessita de uma quantidade significativamente grande de recursos para seu pleno funcionamento. Da mesma forma que ocorre na distribuição do conteúdo armazenado, há a necessidade de uma grande quantidade de banda de rede para realizar a transmissão da mídia. Isso ocorre principalmente porque a maioria das aplicações são baseadas no modelo tradicional *cliente-servidor*. Além disso, estas transmissões originam-se em um único servidor, o que torna o sistema suscetível a ataques e falhas nesse ponto.

Atualmente, tecnologias baseadas na arquitetura P2P (entre pares ou Peer-to-Peer) estão substituindo a arquitetura tradicional cliente-servidor. Arquiteturas baseadas em P2P podem aliviar a carga imposta aos servidores e às redes de computadores. Nesta arquitetura, cada participante do sistema é capaz de obter o serviço de visualização da mídia, e também capaz de contribuir com outros participantes, fornecendo parte do conteúdo da mídia.

Existem vários estudos que tratam de protocolos e da organização do sistema de transmissão ao vivo utilizando arquiteturas P2P [35, 37, 69, 89, 103, 105]. Atualmente, grande parte dos sistemas mais populares para transmissão ao vivo em arquiteturas P2P, como PPLive [66], SopCast [81], PPStream [67] e GridMedia [33, 103], usam métodos de organização dos seus participantes baseados em malha e com pedido explícito por dados, também conhecido como *"data-driven mesh-pull overlay"* [35, 56]. O funcionamento dessas aplicações é similar ao funcionamento do Bittorrent [5].

Durante o planejamento e o desenvolvimento de sistemas de transmissão de mídia contínua ao vivo em P2P, é assumido um comportamento altruísta e não malicioso de seus participantes. Porém, esses sistemas podem ser alvos de ataques e comportamentos indesejados de seus participantes. Há facilitadores para ataques em aplicações de transmissão ao vivo em P2P, tais como o uso de mensagens de controle e de dados sem nenhum tipo de codificação ou criptografia por várias aplicações populares [20].

Um dos ataques que podem impactar diretamente os sistemas de transmissão ao vivo em P2P são os ataques de poluição. Neste ataque, os participantes maliciosos alteram ou forjam os dados da mídia transmitida. Em complemento, eles podem realizar um conluio, que é um acordo entre eles para trapacear o sistema, alcançando um ataque com maior impacto [7].

Existem alguns trabalhos que abordam a transmissão de conteúdo poluído em transmissão ao vivo em sistema P2P [20, 30, 31]. Em [20], é realizado um experimento onde os autores comprovam o efeito de conteúdo poluído em um sistema real de transmissão ao vivo em P2P. Em [30, 31], os autores sugerem técnicas para verificar a integridade dos dados que são transmitidos pelo sistema P2P e verificam a sobrecarga imposta por essas técnicas no volume de dados transmitidos (isto é, no consumo de largura de banda da rede).

Neste trabalho é apresentado um modelo descentralizado de reputação para combater ataques em sistemas de mídia contínua P2P. Em particular, são concentrados esforços para combater ataques de poluição. Nesse ataque, os participantes maliciosos alteram ou forjam os dados da mídia contínua na tentativa de disseminar conteúdo indesejado aos demais participantes da rede.

Os resultados desse trabalho incluem uma modelagem de sistemas P2P e uma verificação formal dos danos causados por ataques de poluição. Nesse sentido, há uma evidencia que a escolha dos protocolos e algoritmos do sistema P2P influencia no impacto causado por um ataque. Incluem também uma caracterização dos usuários de sistemas de transmissão ao vivo de mídia contínua em P2P. A partir dessa caracterização é possível compreender a dinâmica dos participantes desses sistemas e assim avaliar o impacto causado por seu comportamento tanto na escalabilidade do sistema, quanto na qualidade da mídia recebida. Finalmente, é desenvolvido um sistema de defesa contra ataques de poluição.

Os resultados encontrados mostram que o novo mecanismo implementado bloqueia um ataque com uma sobrecarga inferior a 2% da banda de rede necessária para a transmissão, enquanto a solução indicada pela literatura necessita de pelo menos 100% de banda adicional. Além disso, o novo mecanismo é eficaz mesmo com conluio dos participantes e, nesse caso, o pico de retransmissão é inferior a 30%, enquanto as

soluções tradicionais necessitam de até 200% de banda adicional. Finalmente, o novo mecanismo permite reabilitação dos participantes que geraram conteúdo indevido e provê medidas de incentivo para o compartilhamento dos recursos na rede P2P.

A análise do comportamento dos participantes e o modelo criado podem ser utilizados tanto para simulações de sistemas P2P quanto como ponto de partida para criação de novos protocolos para transmissão ao vivo.

Os participantes foram caracterizados com foco nos tempos de ON, tempos de OFF e nas características das parcerias como duração e tamanho. Nas transmissões do Sopcast analisadas, os participantes se encaixaram em um dos dois perfis mostrados na tabela 1. Esse enquadramento dos participantes depende principalmente do tipo da transmissão do canal, se é um evento de grande interesse para um público específico ou se é uma transmissão ordinária.

Tabela 1: Resumo dos perfis de participantes do SopCast.

Canal	Tempo entre chegadas	Número de sessões	ON	OFF	Tempo de Parceria
Sem Evento	Lognormal	≤ 2 em	Gamma|Lognormal	Exp.	Gamma
Com Evento	Lognormal	90% dos casos	Weibull	Exp.	Gamma

A principal diferença entre as classes de participantes identificadas é o tempo de ON dos participantes. Porém, mesmo distribuições iguais apresentadas para certas características apresentam valores de parâmetros diferentes (e.g Tempo de Parceria).

Os tempos de ON são menores em canais típicos do SopCast, como a CCTV e o canal especializado em programas esportivos, do que os tempos encontrados em canais transmitindo eventos de grande interesse, como o canal que transmitia o jogo de futebol. No caso específico do canal em dia de jogo de futebol, há um número considerável de participantes que permanecem no sistema por praticamente toda a transmissão.

Finalmente, o modelo formal para verificação dos danos causados por um ataque de poluição em sistemas de transmissão ao vivo em P2P pode ser utilizado para comparar o impacto de ataques de poluição em diferentes estratégias de busca de dados em redes P2P de mídia ao vivo. Esta comparação é baseada principalmente em uma métrica: sobrecarga de rede (o quanto a mais de dados é necessário para a visualização da mídia devido aos dados poluídos).

Por esse modelo, foram analisadas as dua estratégias mais comuns de seleção de dados para busca na rede P2P. Uma das estratégias tenta distribuir mais rapidamente os chunks recém criados e a outra tenta evitar perdas na exibição do conteúdo. Essas estratégias influenciam no resultado de um ataque de poluição e, dependendo da estratégia escolhida, a sobrecarga imposta à rede pode ser maior ou menor.

Os resultados desse modelo mostram que a utilização do algoritmo de seleção de dados “EDF” (estratégia gulosa) leva a um cenário otimista. Neste cenário, todos os parceiros podem atender aos pedidos realizados por um determinado participante. Porém, essa estratégia de seleção só é viável em situações que a latência da rede não é importante. Para os cenários mais comuns, onde a relação entre poluidores e parceiros é baixa, a sobrecarga para esse cenário ficou abaixo de 5%. Caso o algoritmo de seleção de dados seja “RF” (mais raro primeiro), os resultados mostram um cenário pessimista, onde a sobrecarga imposta por ataques é alta. Na maior parte dos casos, os participantes têm que ter pelo menos 3 vezes a quantidade de banda de rede necessária para uma transmissão ao vivo.

Lista de Figuras

1.1 Estrutura para transmissão ao vivo utilizando redes P2P. 2
1.2 Exemplo do mecanismo de checar e pedir retransmissão. 5

2.1 Árvore de multicast em nível da camada de aplicação. 20
2.2 Manutenção da árvore de multicast em nível da camada de aplicação. . . . 22
2.3 Sistema baseado em múltiplas árvores com dois subfluxos. 23
2.4 Atividade inicial de um novato - rede P2P baseada em malha. 24
2.5 Troca de dados na aplicação baseada em malha. 27

3.1 Modelo de sistema utilizado. 49
3.2 Mecanismo de consumo da mídia ao vivo. 56

4.1 Modelo do participante. 65
4.2 Modelo hierárquico das sessões dos participantes. 66
4.3 Utilização do canal durante um dia. 69
4.4 Utilização do canal durante os dias da semana. 70
4.5 Proporção dos tempos entre mensagens encontrados. 71
4.6 Distribuição do tempo entre requisições. 72
4.7 Tempo entre as chegadas de sessões. 73
4.8 Número de sessões de participante. 75
4.9 Tempo ON do participante. 76
4.10 Tempo ON do participante - caracterização em dia sem eventos. 76
4.11 Tempo ON do participante - caracterização em dia com eventos 77
4.12 Tempo OFF dos participantes. 78
4.13 Número de parceiros. 80
4.14 Duração das parcerias 81
4.15 Duração das parcerias - caracterização em dia com eventos 82
4.16 Duração das parcerias - caracterização em dia sem eventos 83
4.17 Participantes são impacientes. 84

4.18 Tamanho da rede Sopcast para os canais analisados. 85

5.1 Mecanismo de janela deslizante dos participantes da rede P2P. 88
5.2 Descarga de um *chunk* da rede - abordagem "marcar e checar". 91
5.3 Encadeamento de requisições até um sucesso. 91
5.4 Exemplo de difusão de um *chunk* usando "mais raro primeiro". 96

6.1 Cenário do sistema simulado. 108
6.2 Visão de um participante sob ataque de poluição. 109
6.3 Visão geral do sistema sob ataque de poluição - Dados. 110
6.4 Visão geral do sistema sob ataque de poluição - Parcerias. 111
6.5 Taxa de transmissão de dados na rede
abordagem "marcar e checar". 113

7.1 Taxa de retransmissão dos participantes do SopCast. 117
7.2 Número de retransmissões de um dado poluído. 118
7.3 Utilização da banda da rede no SopCast. 119
7.4 Número de parceiros do poluidor. 120
7.5 Número de participantes da rede SopCast. 120
7.6 Visão geral do SopCast sob ataque de poluição. 122

8.1 Mecanismo de alteração do limite de reputação. 131
8.2 Cenário do sistema simulado. 135
8.3 Lista negra centralizada. 137
8.4 Sistema de reputação distribuído convencional. 139
8.5 Sistema de reputação simplificado. 140
8.6 Comparação entre os métodos sem conluio dos poluidores. 143
8.7 Comparação entre os métodos com conluio dos poluidores. 144
8.8 Variação do parâmetro y_i do modelo simplificado. 145
8.9 Variação do parâmetro P. 146
8.10 Variação do parâmetro G. 147
8.11 Ataques dissimulados com o sistema simplificado. 148

Lista de Tabelas

1 Resumo dos perfis de participantes do SopCast. xvii

2.1 Resumo dos elementos do modelo de lista negra centralizada. 45

3.1 Resumo dos parâmetros de um sistema P2P de transmissão ao vivo. 52
3.2 Resumo dos parâmetros utilizados na geração de conteúdo. 53
3.3 Resumo dos parâmetros utilizados na realização de parcerias. 54

4.1 Distribuição dos tempos de chegada de sessões. 74
4.2 Distribuição dos tempos ON. 77
4.3 Distribuição dos tempos OFF. 79
4.4 Distribuição do número de parceiros: Resumo. 80
4.5 Duração das parcerias. 83
4.6 Resumo dos perfis de participantes 83

5.1 Resumo dos elementos do modelo. 90
5.2 Parâmetros para avaliação dos modelos propostos. 97
5.3 Resultado da avaliação dos mdelos - 100 parceiros em média. 98
5.4 Avaliação do modelo - 50 parceiros em média. 99

6.1 Waxman - parâmetros de AT&T e DFN G-Win. 105
6.2 Waxman - parâmetros para topologia Internet. 105
6.3 Topologias de rede geradas por Waxman. 106
6.4 Parâmetros da simulação. 108

7.1 Parceiros contaminados ou que contaminaram. 123

8.1 Resumo dos elementos do modelo descentralizado. 128
8.2 Resumo dos elementos do modelo simplificado. 133
8.3 Topologias de rede geradas por Waxman. 133

8.4 Parâmetros da simulação. 135
8.5 Sobrecarga causada pelo ataque em um sistema com a lista negra centralizada (% da banda necessária). 138
8.6 Sobrecarga causada pelo ataque em um sistema com a abordagem convencional (% da banda necessária). 140
8.7 Sobrecarga causada pelo ataque em um sistema com a abordagem simplificada (% da banda necessária). 141

Sumário

Agradecimentos **vii**

Resumo **xi**

Abstract **xiii**

Resumo Estendido **xv**

Lista de Figuras **xix**

Lista de Tabelas **xxi**

1 Introdução **1**

2 Arquiteturas de Redes Par-a-Par **9**

2.1 Redes Par-a-Par (P2P) 9

2.1.1 Arquiteturas P2P 10

2.1.2 Mecanismos para Descoberta de Recursos 13

2.1.3 Exemplos de Sistemas P2P 15

2.2 Distribuição de Mídia Contínua ao Vivo em Arquiteturas P2P 18

2.2.1 Estrutura Baseada em Árvore 19

2.2.2 Estrutura Baseada em Malha 22

2.2.3 Estrutura Híbrida 28

2.3 Ataques aos Sistemas P2P 30

2.3.1 Ataques e Comportamentos Indesejados em Transmissão ao Vivo 30

2.3.2 Ataques de Poluição 34

2.3.3 Comportamentos Oportunistas 36

2.3.4 Ataques e Poluição de Dados aos Sistemas de Transmissão ao Vivo em P2P 38

2.3.5 Verificação do Conteúdo Distribuído 40

2.3.6 Lista Negra Centralizada 42
2.4 Resumo do Capítulo 46

3 Sistema de Transmissão ao Vivo em P2P 49
3.1 Sistema P2P para Transmissão ao Vivo 50
3.2 Geração do Conteúdo da Transmissão 52
3.3 Realização de Parcerias 53
3.4 Armazenamento e Consumo de Dados 54
3.5 Estratégias de Seleção de *Chunks* 56
3.5.1 Estratégia "Mais Raro Primeiro" 56
3.5.2 Estratégia "Gulosa" 57
3.6 Seleção de Parceiros para Troca de Dados 57
3.7 Resumo do Capítulo 58

4 Caracterização do Comportamento dos Participantes de um Sistema de Transmissão ao Vivo em P2P 61
4.1 Modelo de P2P ao Vivo Adotado 64
4.2 Metodologia para Coleta dos Dados 67
4.2.1 Canais Analisados 67
4.2.2 Coleta dos Dados do Sopcast 68
4.3 Características de Acesso dos Participantes 69
4.4 Características dos Participantes 70
4.4.1 Delimitação das Sessões dos Participantes 71
4.4.2 Processo de Chegada de Sessões 72
4.4.3 Características das Sessões 74
4.4.4 Tempos de ON 74
4.4.5 Tempos de OFF 77
4.4.6 Características das Parcerias 79
4.4.7 Resumo das Características Encontradas 83
4.5 Resumo do Capítulo 86

5 Modelo Formal de um Ataque de Poluição 87
5.1 Modelo Básico 88
5.2 Estratégias de Seleção de *Chunks* 89
5.3 Impacto Gerado pela Disseminação de Conteúdo Poluído 89
5.3.1 Estratégia "Gulosa" - Cenário Otimista 93
5.3.2 Estratégia *"Mais Raro Primeiro"* - Cenário Pessimista 93
5.4 Análise dos Resultados 97

5.5 Resumo do Capítulo 99

6 Sistema de Transmissão ao Vivo em P2P Sob Ataque de Poluição 101
6.1 Modelo de Simulação 103
6.1.1 Topologia da Rede Simulada 103
6.1.2 Topologia Lógica 106
6.1.3 Modelo de Transmissão ao Vivo em P2P 106
6.1.4 Modelo do participante: 107
6.1.5 Cenário Adotado 107
6.2 Resultados de um Ataque 108
6.2.1 Sistema Sem Verificação da Integridade dos Dados 108
6.2.2 Sistema Com Verificação da Integridade dos Dados 111
6.3 Resumo do Capítulo 114

7 Ataques de Poluição no SopCast 115
7.1 Poluição de Dados no SopCast 116
7.2 Experimentos Realizados 116
7.3 Análise dos Resultados Experimentais 117
7.4 Limite Inferior para Simulação 121
7.5 Resumo do Capítulo 123

8 Combate à Poluição em Transmissões ao Vivo em P2P 125
8.1 Abordagem Distribuída 126
8.2 Abordagem Distribuída Simplificada 128
8.3 Análise do Combate à Poluição 132
8.3.1 Modelo do Participante 134
8.3.2 Cenário Adotado 134
8.3.3 Comparação das Abordagens. 135
8.4 Análise dos Resultados Utilizando o Sistema Simplificado 142
8.4.1 Variação dos Parâmetros Exponencial y_i 143
8.4.2 Variação dos Parâmetros α_p 145
8.4.3 Variação dos Parâmetros α_g 145
8.4.4 Ataques Dissimulados 146
8.5 Resumo do Capítulo 148

9 Conclusões 151
9.1 Resumo 151
9.2 Limitações 152

9.3 Problemas em Aberto 153

Referências Bibliográficas **155**

Capítulo 1

Introdução

As aplicações de distribuição de mídia contínua são uma das aplicações mais comuns na Internet. De fato, a facilidade para publicação de conteúdo e os diversos sistemas existentes para armazenar e distribuir vídeo e áudio geram um efeito bola de neve. Um ciclo virtuoso onde mais conteúdo desperta o interesse de um maior público que, em contrapartida, gera mais publicações de conteúdo [63].

Para ilustrar este cenário promissor e efervescente para aplicações de distribuição de vídeo e áudio, alguns relatórios mostram que o *Youtube* - um dos sistemas mais populares para publicação e distribuição de vídeo armazenado - apresenta importantes marcas de visita e tráfego na rede. Há relatórios que mostram que ele (o Youtube) receba cerca de 20% das visitas globais de usuários da Internet por dia (www.alexa.com 2009). Outros apontam que o *Youtube* é responsável por 10% de todo o tráfego de Internet da América do Norte (www.ellacoya.com 2008).

As tecnologias utilizadas para a transmissão de mídia contínua (*streaming*) também trazem aplicações mais interessantes que a distribuição do conteúdo armazenado. Elas podem ser utilizadas para transmitir conteúdo ao vivo, tal qual o sistema tradicional de uma TV, mas de maneira muito mais flexível [98]. Devido ao custo operacional, as redes tradicionais de TV oferecem um canal ou uma programação somente se existe um número suficiente de usuários próximo à sua base de operações. Por exemplo, uma TV americana poderia ofertar conteúdo em chinês para a cidade de Nova York, ou em português para Miami. Nessas duas cidades há um grande número de imigrantes chineses e brasileiros respectivamente, mas em outras partes do país, estes públicos provavelmente não são tão representativos. Porém, a utilização da Internet para transmissão de conteúdo ao vivo flexibiliza este contexto, possibilitando a oferta de um canal ou programação para qualquer local que tenha acesso à Internet.

O envio de mídia contínua ao vivo na Internet necessita de uma quantidade

significativamente grande de recursos para seu pleno funcionamento. Da mesma forma que ocorre na distribuição do conteúdo armazenado, há a necessidade de uma grande quantidade de banda de rede para realizar a transmissão da mídia. Isso ocorre principalmente porque a maioria das aplicações são baseadas no modelo tradicional *cliente-servidor*. Além disso, geralmente essas transmissões originam-se em um único servidor, o que torna o sistema suscetível a ataques e falhas nesse ponto.

Atualmente, tecnologias baseadas na arquitetura P2P (entre pares ou Peer-to-Peer) estão substituindo a arquitetura tradicional cliente-servidor. Arquiteturas baseadas em P2P podem aliviar a carga imposta aos servidores e às redes de computadores. Nesta arquitetura, cada participante do sistema é capaz de obter o serviço de visualização da mídia, e também capaz de contribuir com outros participantes, fornecendo parte do conteúdo da mídia. Assim, a banda de rede necessária em um único ponto no modelo cliente-servidor é compartilhada entre os diversos participantes do sistema de transmissão de mídia contínua ao vivo.

A figura 1.1 exemplifica este sistema onde, o servidor de mídia ao vivo repassa o conteúdo a poucos participantes do sistema e, os demais participantes realizam parcerias entre si para trocar informações e dados já recebidos. Desta maneira, o conteúdo ao vivo é visualizado sem a necessidade de se obter toda a mídia de um servidor centralizado.

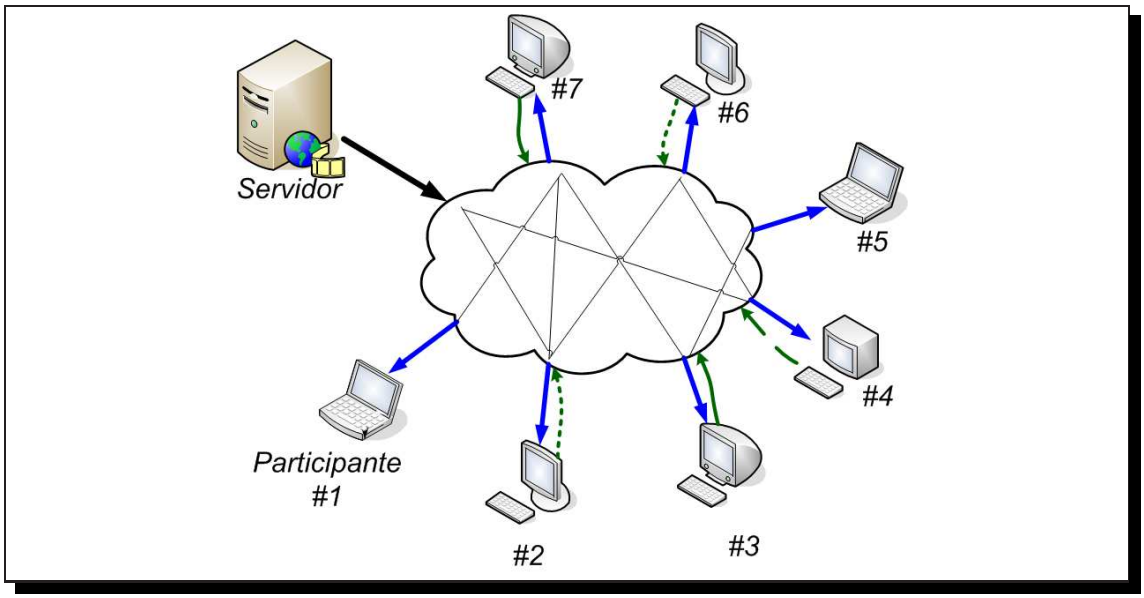


Figura 1.1: Estrutura para transmissão ao vivo utilizando redes P2P.

Existem vários estudos que tratam de protocolos e da organização do sistema de transmissão ao vivo utilizando arquiteturas P2P [35, 37, 69, 89, 103, 103, 105]. Estes protocolos e técnicas de organização têm como objetivo a escalabilidade do sistema

quanto ao número de usuários e à qualidade do vídeo transmitido. Assim, espera-se poder suportar um grande número, assim como índice de qualidade e um baixo atraso percebido na mídia transmitida ao vivo.

As técnicas de organização e disseminação da transmissão ao vivo se dividem em dois grandes grupos. Parte dos trabalhos propõe uma organização dos participantes do sistema sob uma estrutura de árvore, e outra parte propõe a utilização de uma estrutura baseada em malha. Na estrutura de árvores, os usuários nos níveis mais altos da hierarquia se encontram mais próximos à origem da mídia ao vivo. Nesse caso, o dado é repassado de *pai* para *filho*, sem a necessidade de requisição explícita pelo dado da mídia. No segundo caso, onde a organização é baseada em uma malha, os usuários fazem parcerias aleatórias e pedem explicitamente pelos trechos da mídia que eles precisam para visualização desta. Nesta abordagem, os diversos participantes do sistema trocam com seus parceiros informações que possibilitam o mapeamento dos dados disponíveis e a reconstrução do fluxo da mídia transmitida. Além dessas duas frentes, há propostas híbridas que combinam várias maneiras de busca por dados e organização dos participantes do sistema.

Atualmente, grande parte dos sistemas mais populares para transmissão ao vivo em arquiteturas P2P, como PPLive [66], SopCast [81], PPStream [67] e GridMedia [33, 103], usam métodos de organização dos seus participantes baseados em malha e com pedido explícito por dados, também conhecido como *"data-driven mesh-pull overlay"* [35, 56]. O funcionamento dessas aplicações é similar ao funcionamento do Bittorrent [5], um popular sistema de compartilhamento de arquivos. Em tais sistemas, o vídeo é codificado e dividido em pequenas partes por uma fonte sorvedora. Essas pequenas unidades de mídia são denominadas fatias ou *chunks*. Ao se juntar ao sistema, um novo participante estabelece conexões com um subconjunto aleatório de participantes que já se encontram assistindo o conteúdo ao vivo. Eles trocam informações entre si, anunciando as fatias de dados disponíveis e desejadas. A partir daí, eles podem realizar ou receber pedidos por dados, e, dessa forma, podem receber e enviar dados de/para seus parceiros.

Durante o planejamento e o desenvolvimento de sistemas de transmissão de mídia contínua ao vivo, baseados em arquiteturas P2P, é assumido um comportamento altruísta e não malicioso de seus participantes. Assim, as aplicações são projetadas com a expectativa de que os usuários compartilhem dados de maneira proporcional aos dados recebidos. Mais ainda, não se esperam ataques contra a estrutura e o serviço de transmissão ao vivo. Entretanto, sistemas reais podem possuir usuários egoístas e maliciosos, o que pode levar a um comportamento inesperado de todo o sistema de distribuição de conteúdo ao vivo em P2P.

Apesar da atenção recebida e da grande quantidade de trabalhos relacionados a ataques e comportamentos oportunistas em aplicações de compartilhamento de arquivos em P2P [14, 15, 19, 91], ainda são poucos os esforços neste sentido em transmissão ao vivo P2P [20,30]. Mesmo adotando estruturas semelhantes, as premissas para ambos os sistemas são diferentes. Enquanto um se preocupa com a distribuição correta de um arquivo completo, o outro tem que se preocupar com a distribuição de trechos em um curto intervalo de tempo. Assim, não há garantias de sucesso ao adotar as soluções de um sistema diretamente no outro, uma vez que conclusões válidas para uma aplicação baseada em P2P podem não ser válidas para outras aplicações P2P [90].

Um dos ataques observados em redes de compartilhamento de arquivos em P2P é a poluição de conteúdo [48]. Nesse ataque, participantes maliciosos alteram ou forjam o conteúdo compartilhando, tornando-o inútil para os demais. Os participantes do sistema podem obter e repassar esses dados poluídos, consumindo recursos com dados indesejados e aumentando o problema de poluição no sistema. Assim, o conteúdo legítimo fica menos disponível por se confundir com o conteúdo poluído e, por consequência, o sistema de compartilhamento fica comprometido.

Da mesma forma que ocorre poluição de conteúdo em compartilhamento de arquivos P2P, os sistemas de distribuição ao vivo em P2P também podem ser alvo desse ataque. Há facilitadores para ataques nessas aplicações, tais como o uso de mensagens de controle e de dados sem nenhum tipo de codificação ou criptografia por várias aplicações populares [20]. Assim, um participante que deseja atacar o sistema pode alterar ou injetar mensagens falsas na mídia transmitida. Ele pode também se articular e realizar um ataque combinado, juntamente com outros participantes para causar um maior dano ao sistema de transmissão ao vivo P2P. Assim, eles realizam um conluio, que é um acordo entre os participantes maliciosos para trapacear o sistema, alcançando um ataque com maior impacto [7].

Os estudos mais recentes para conter comportamentos indesejados em aplicações de transmissão ao vivo em P2P enfatizam principalmente o comportamento egoísta dos participantes da rede e tentativas de gerar ataques de negação de serviço. Nessa linha, há propostas de sistemas de reputação e mecanismos de incentivo em [13, 40, 85]. Tais propostas tentam incentivar a colaboração dos participantes do sistema e fornecem maneiras para reputar o comportamento destes. Nesses trabalhos, a avaliação da reputação de um integrante é realizada a partir das observações dos demais participantes, sendo também considerada a interação direta entre o participante que está sendo avaliado e o que deseja conhecer sua reputação. A partir dessa avaliação de reputação, um participante do sistema pode decidir se é compensador estabelecer uma nova parceria, trocar dados e informações ou encerrar a parceria existente.

Existem alguns trabalhos que abordam a transmissão de conteúdo poluído em transmissão ao vivo em sistema P2P [20, 30, 31]. Em [20], é realizado um experimento onde os autores comprovam o efeito de conteúdo poluído em um sistema real de transmissão ao vivo em P2P. Em [30, 31], os autores sugerem técnicas para verificar a integridade dos dados que são transmitidos pelo sistema P2P e verificam a sobrecarga imposta por essas técnicas no volume de dados transmitidos.

Estes três trabalhos evidenciam que é possível marcar o dado na fonte geradora e realizar a checagem dos participantes do sistema, com um baixo custo adicional. Utilizando os esquemas propostos, o custo de processamento para marcar o dado é $O(n)$, onde n é a quantidade de fatias (*chunks*) que se está marcando. Para realizar a checagem, o custo de processamento é $O(1)$ por conjunto de *chunks*. Para a transmissão da mídia com seus dados marcados, é necessário um aumento de 5% na banda de rede.

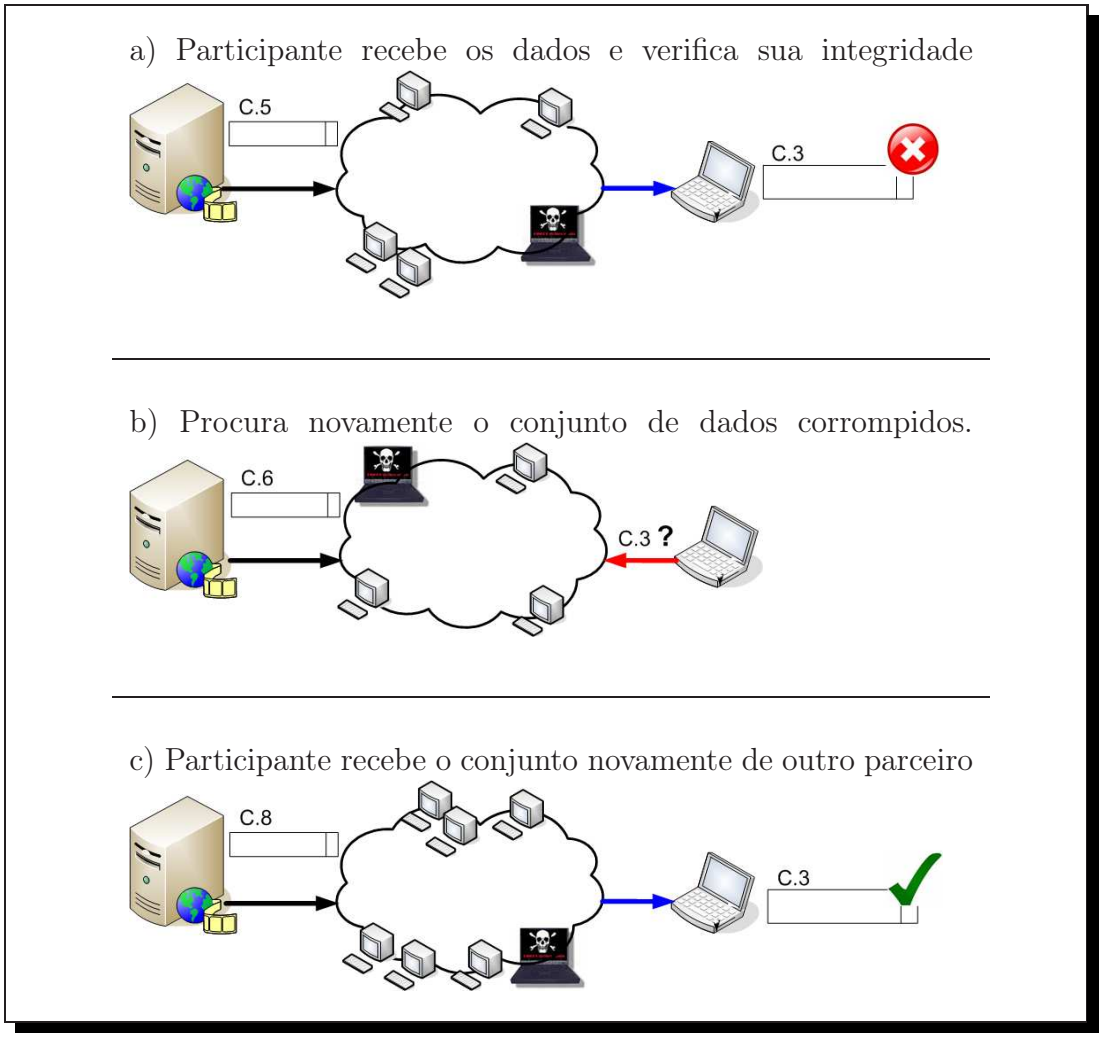


Figura 1.2: Exemplo do mecanismo de checar e pedir retransmissão.

Esse processo de verificação dos dados transmitidos ao vivo é mostrado na figura 1.2. Um participante recebe de algum de seus parceiros um conjunto de dados e faz a verificação (figura 1.2-a). Quando este dado está correto, ele armazena em sua área de armazenamento temporário (*buffer*) e, posteriormente, executa este trecho da mídia. Quando o dado está inválido, o usuário o descarta e refaz o pedido por este dado a algum outro parceiro (figura 1.2-b). Esse processo é repetido até que o dado recebido esteja correto (figura 1.2-c).

Apesar dessa estratégia ser intuitiva, algumas perguntas chave ainda permanecem em aberto:

- Qual o real impacto de um ataque de poluição e conluio em um sistema de transmissão ao vivo em redes P2P?
- Como comportamentos oportunistas dos usuários (e.g. egoísmo) e comportamentos indesejados (e.g. entradas e saídas constantes dos participantes) podem influenciar na transmissão ao vivo?
- Qual o comportamento do sistema, ao adotar a verificação da mídia para tentar conter um possível ataque de poluição?
- Há a possibilidade de se adotar as medidas existentes em compartilhamento de arquivos em P2P para o contexto de transmissão ao vivo? Elas serão eficientes no novo contexto?
- Há métodos que consigam coibir de maneira eficiente os ataques e comportamentos indesejados supracitados? Quais são as métricas que expressam a eficiência destes métodos? Quais são os limites destes métodos?

Atualmente, o tratamento de poluição e comportamento oportunista em sistemas de transmissão de vídeo ao vivo em P2P, nos poucos trabalhos atuais, são tratados como um problema de checagem de dados transmitidos pelo sistema. O principal trabalho desta tese será refutar que a checagem dos dados é eficiente ao tratar os ataques de poluição. Assim, há o objetivo de mostrar que a adoção de um sistema auxiliar à aplicação de transmissão ao vivo em P2P, como um sistema de reputação, leva a resultados mais rápidos, eficientes e com menor sobrecarga que as maneiras já propostas. Mais ainda, nesse contexto, um sistema baseado somente em informações locais dos participantes leva a resultados melhores que a maneira intuitiva de agregar as informações dos diversos participantes do sistema de transmissão ao vivo.

Esta tese tem como objetivo estudar estas questões.

Contribuições

As principais contribuições desta tese são:

- Um modelo de funcionamento das aplicações de distribuição de mídia contínua ao vivo em P2P, mostrado no capítulo 3. Este modelo é utilizado como base para a verificação dos sistemas propostos.
- Uma caracterização dos participantes de sistema de transmissão ao vivo em P2P, apresentada no capítulo 4. A partir desse modelo, pode-se recriar o comportamento dos participantes em diferentes tipos de transmissões ao vivo, como por exemplo, eventos esportivos ou transmissões de noticiários.
- Uma avaliação formal dos impactos de ataque de poluição em sistemas de transmissão ao vivo P2P, apresentada no capítulo 5.
- Finalmente, a proposta de um sistema de reputação com um algoritmo que utiliza somente informação local, apresentado no capítulo 8. Esse sistema é capaz de tratar os problemas causados por um ataque de poluição, isolar os atacantes e, apesar de mais simples que abordagens tradicionais, tem um desempenho melhor, como menor tempo de resposta ao ataque e menor sobrecarga na banda de rede dos participantes da aplicação.

Organização da Tese

O restante da tese está estruturado em 7 capítulos como segue:

Capítulo 2 [Arquiteturas de Redes Par-a-Par]: São apresentadas os conceitos de sistemas P2P; seu funcionamento, organização e controle. São apresentados também os conceitos de distribuição de mídia contínua ao vivo em P2P. Além disso, são discutidos os problemas de ataques em P2P, assim como as medidas atuais para o combate.

Capítulo 3 [Sistemas de Transmissão ao Vivo em P2P]: São apresentadas definições básicas, notações, desafios e técnicas a respeito de transmissão de fluxo contínuo ao vivo, estruturas de redes P2P e abordagens utilizadas para a transmissão ao vivo em redes P2P.

Capítulo 4 [Caracterização do Comportamento dos Participantes do SopCast]: É apresentada uma caracterização do comportamento dos participantes de uma das aplicações mais populares para transmissão ao vivo em P2P, o SopCast. Essa caracterização trás todas as informações necessárias para reconstrução do comportamento dos usuários do sistema de transmissão ao vivo em P2P.

Capítulo 5 [Modelo Formal de um Ataque de Poluição]: Apresenta um modelo formal para verificar o dano causado por um ataque de poluição aos sistemas de transmissão ao vivo em P2P. Os resultados encontrados nesse capítulo servem de base para verificar os resultados das simulações realizadas nessa tese. Além disso, são analisados os resultados do impacto de poluição com dependência dos algoritmos de escolha de parceria e dados utilizados pelo sistema P2P.

Capítulo 6 [Sistemas de Transmissão ao Vivo em P2P Sob Ataque de Poluição]: São apresentadas simulações que motivam o estudo mais aprofundado de técnicas para tratamento e controle de ataques em sistemas P2P de transmissão ao vivo, sobretudo poluição. Nas simulações realizadas, verifica-se que a simples verificação do conteúdo distribuído pela rede P2P não é suficiente para evitar os danos ao sistema.

Capítulo 7 [Ataques de Poluição no Sopcast]: Nesse capítulo é apresentado um experimento que também serve como base de verificação das simulações realizadas na tese. É apresentado os resultados de um ataque de poluição direcionado ao SopCast.

Capítulo 8 [Combate à Poluição em Transmissão ao Vivo em P2P]: Neste capítulo é apresentada a principal contribuição da tese. Nele é discutido uma nova abordagem para combater ataques aos sistemas de transmissão ao vivo em P2P. O foco continua sendo ataque de poluição e os resultados encontrados indicam que a nova abordagem apresenta um melhor desempenho que as sugestões encontradas na literatura. Deve-se ressaltar que a nova abordagem é mais simples que as demais.

Capítulo 9 [Conclusões]: Finalmente, as contribuições, as limitações e possíveis trabalhos futuros são apresentados, e assim, a tese é concluída.

Capítulo 2

Arquiteturas de Redes Par-a-Par

Este capítulo apresenta uma revisão dos esforços prévios no sentido de estruturar e organizar os sistemas de transmissão ao vivo em arquiteturas P2P. Assim, a seção 2.1 faz uma breve introdução aos sistemas baseados no modelo P2P. Nessa seção são explorados sistemas para compartilhamento de recursos, por exemplo, arquivos. A seção 2.2 apresenta as principais abordagens utilizadas para criar e gerenciar os serviços de transmissão ao vivo em redes P2P. Nesta seção, são discutidas a organização da rede P2P e a transmissão dos dados entre os participantes do sistema. A seção 2.3 faz uma revisão dos principais problemas de segurança e comportamentos indesejados dos usuários de sistemas P2P. Finalmente, a seção 2.4 resume os principais pontos discutidos neste capítulo.

2.1 Redes Par-a-Par (P2P)

Os serviços disponíveis em redes de computadores são tradicionalmente baseados no modelo cliente-servidor. Nesse modelo, um ou mais servidores centralizados provêem todo o serviço desejado, e realizam todo o processamento da aplicação localmente.

Apesar da grande quantidade de aplicações baseadas no modelo cliente-servidor, as limitações tornam-se evidentes em sistemas distribuídos de larga escala. Nesses sistemas, é necessário o uso de sofisticados mecanismos de balanceamento de carga e algoritmos de tolerância a falhas, para prover o serviço com tempos de respostas em níveis aceitáveis [65]. Mais ainda, essas aplicações estão sujeitas aos limites da banda de rede e, caso um número grande de usuários tenham interesse em obter serviço simultaneamente, a rede pode torna-se um ponto de contenção.

Tais problemas motivaram o desenvolvimento de abordagens que distribuem o processamento e a carga, além de dividir a banda de rede por todos os participantes.

Essas abordagens apresentam arquiteturas de computadores distribuídas, chamadas de Par-a-Par ou Peer-to-Peer (P2P). Esses sistemas são projetados para compartilhamento de recursos computacionais por uma troca direta entre os participantes, sem a necessidade de requisição do intermédio ou suporte de um servidor centralizado [1].

Mais precisamente, uma rede P2P é uma rede de computadores que exibe auto-organização, comunicação simétrica e controle distribuído [72]. Os conceitos da arquitetura P2P foram inicialmente descritos no primeiro "Request for Comments" (RFC 1) "Host Software"de 7 de Abril de 1969 [17]. Entretanto, a primeira rede P2P de grande uso foi o sistema *"Usenet news server"* de 1979. Na Usenet, enquanto os usuários finais acessavam as notícias através de servidores, os servidores comunicavam entre si para propagar os artigos de notícias. A comunicação entre os servidores era feita de forma similar às aplicações P2P atuais, sem a necessidade de centralização.

Atualmente, existe uma grande quantidade de aplicações baseadas no modelo P2P. Grande parte dessas aplicações ganhou notoriedade por compartilhar arquivos, geralmente músicas e vídeos. Porém, há outros importantes sistemas P2P que, além de compartilhamento de arquivos, se preocupam também com outros recursos, como processamento, banda de rede e armazenamento de informações.

2.1.1 Arquiteturas P2P

Uma rede P2P é uma rede composta por computadores participantes (peers ou nós), e as conexões existentes entre eles. Essa rede é formada sobre e independentemente da rede física de computadores existentes (tipicamente uma rede IP). Há várias maneiras de se classificar as redes P2P, como descrito a seguir.

2.1.1.1 Definição Quanto à Centralização

Apesar da suposição que as redes P2P são totalmente descentralizadas, na prática, encontram-se variações desse modelo e os sistemas apresentam graus de centralização. Em relação a um maior ou menor grau de centralização, as arquiteturas P2P apresentam três categorias bem identificadas:

(A) Arquitetura Puramente Descentralizada:
Nesta arquitetura, todos os nodos apresentam exatamente o mesmo comportamento e realizam as mesmas operações. Eles não se distinguem entre servidores ou clientes, e têm capacidade de agir como ambos. Além disso, não há uma unidade centralizadora para coordenar as atividades da rede.

Os sistemas puramente descentralizados são inerentemente escaláveis e tolerantes a falhas. A escalabilidade é alcançada porque tais arquiteturas não apresentam elementos centralizadores ou servidores. Esses elementos centralizadores podem ser pontos de contenção e geralmente restringem a escalabilidade de um sistema qualquer. Como não há um ponto de falha localizado e a perda de participantes do sistema pode ser facilmente compensada, essas arquiteturas tornam-se tolerantes a falhas. Há também uma grande autonomia local em relação ao controle dos recursos existentes no sistema, o que faz que os participantes sejam pouco afetados por ausência de outros.

Apesar dessas vantagens, os sistemas totalmente distribuídos apresentam uma descoberta de informações lenta, e sem garantia quanto à qualidade do serviço. Principalmente pela impossibilidade de se ter uma visão global do sistema, o que dificulta a previsão do comportamento do mesmo [65].

Sistemas como Gnutella [25], Freenet [12], Chord [82] e CAN [70] são exemplos desse tipo de sistema. Entre esses, o Gnutella se destaca como uma importante aplicação para compartilhamento de arquivos, sobretudo arquivos de música.

(B) Arquitetura Híbrida Descentralizada:
Em um sistema P2P híbrido, há um servidor centralizado que facilita as interações entre os participantes. Esse servidor mantém metadados que descrevem os recursos compartilhados e os participantes do sistema. Apesar da centralização para obter informações da rede, as interações são realizadas diretamente entre os participantes. Por exemplo, uma troca de dados ou de arquivos é realizada entre os dois participantes do sistema que estabeleceram uma parceria para realizar tal tarefa. Apesar disso, servidor centralizado tipicamente acarreta baixa escalabilidade, vulnerabilidade a ataques, e atos de censura [1].

Um dos exemplos mais importantes dessa arquitetura é o sistema Napster [62]. O Napster foi um dos primeiros sistemas populares para compartilhamento de arquivos na Internet, sobretudo arquivos de música. O Napster teve seu serviço interrompido devido a restrições judiciais, principalmente por causa de conteúdo pirateado compartilhado por meio desse sistema.

(C) Arquitetura Parcialmente Centralizada:
Este tipo de arquitetura é semelhante à arquitetura puramente descentralizada, porém, alguns de seus participantes apresentam papéis mais importantes no sistema. Eles atuam como elementos centralizadores, facilitando a localização e a troca de recursos entre os demais participantes. Os nodos mais importantes são conhecidos como

super-nodos [4] e, em geral, são dinamicamente escolhidos entre os participantes da rede para desempenhar sua função.

A maneira como esses super-nodos desempenham suas funções varia de sistema para sistema, porém, deve-se ressaltar que eles não são pontos de falha para a rede P2P [1, 4]. Como são dinamicamente escolhidos, caso algum deles falhe, a rede irá automaticamente substituí-lo por outro participante.

Exemplos desse tipo de arquitetura são o Kazaa [43] e o Morpheus [61]. Ambos são utilizados para compartilhamento de arquivos, sobretudo de áudio e vídeo.

2.1.1.2 Definição Quanto à Estrutura

A definição de uma arquitetura quanto à estrutura refere-se à maneira como o sistema é criado e mantido. Há três maneiras de se criar a estrutura. A primeira, não é determinista na adição de novos participantes e/ou conteúdo. A outra segue regras específicas para a criação da estrutura P2P. A terceira situa-se entre as duas inicialmente citadas.

(A) Redes Sobrepostas P2P Não Estruturadas:
No primeiro caso, as redes não são estruturadas. A localização de um recurso não tem relacionamento com a topologia da rede sobreposta e os participantes são conectados diretamente uns aos outros. Nesses sistemas, as buscas por recursos ocorrem por uma série de pesquisas aleatórias, onde vários participantes são questionados se possuem o recurso em questão.

Nessa arquitetura, a localização do conteúdo precisa de mecanismos de busca que vão desde métodos baseados na força bruta (e.g. inundação de mensagens) e maneiras mais sofisticadas, que incluem roteamento aleatório e índices dos elementos da rede. Os mecanismos de busca empregados em uma rede não estruturada têm uma implicação forte no desempenho do sistema, principalmente, na escalabilidade, disponibilidade e persistência [1].

Os sistemas não estruturados são, geralmente, apropriados para acomodar população de nodos altamente transientes. Ou seja, populações em que os participantes entram e abandonam o sistema constantemente. Alguns exemplos de sistemas P2P não estruturados são: Napster [62], Gnutella [25] e Kazaa [43].

(B) Redes Sobrepostas P2P Estruturadas:
Rede sobreposta P2P estruturada é uma rede onde a sua formação e manutenção seguem regras bem definidas. A topologia é controlada e os recursos são

disponibilizados em locais específicos do sistema. Esses sistemas provêem um mapeamento entre o identificador do recurso e a localização, na forma de uma tabela de roteamento. Assim, as pesquisas podem ser eficientemente roteadas para o nodo que contém o recurso desejado [54].

Essas arquiteturas estruturadas surgiram principalmente para tratar questões de escalabilidade em sistemas, que originalmente tinham esse problema. Elas oferecem uma solução para casamento exato das consultas, ou seja, as consultas são direcionadas diretamente para o recurso procurado. Uma desvantagem de sistemas estruturados é o alto custo para manter a estrutura necessária para um roteamento eficiente das mensagens [55]. Os participantes do sistema podem ter um comportamento dinâmico e transiente, entrando e abandonando o sistema com uma alta taxa.

Exemplos de redes P2P estruturados são: Chord [82], CAN [70], e Tapestry [107].

(C) Redes Sobrepostas P2P Fracamente Estruturadas:
Uma rede fracamente estruturada é uma categoria de rede que se situa entre a condição de uma rede estruturada e uma não estruturada. Embora a localização de um recurso não seja completamente especificada, ela é afetada por indicações de uma provável rota. Um exemplo típico para esse tipo de sistema é Freenet [12].

2.1.2 Mecanismos para Descoberta de Recursos

Os sistemas distribuídos P2P, geralmente, necessitam de um mecanismo de descoberta de recursos. Por exemplo, em um sistema distribuído P2P para compartilhamento de arquivos é necessário um mecanismo para a busca dos arquivos desejados.

Historicamente, há três grandes vertentes em mecanismos de descoberta de recursos em P2P [2, 44]. A primeira delas é baseada em estruturas centralizadas, como o esquema utilizado no Napster [27]. Essa pode ser pouco escalável e suscetível a ataques [2], uma vez que existe um único ponto de falhas. A segunda vertente adota soluções distribuídas, como o esquema utilizado no Gnutella [71]. Essas aplicações distribuem consultas para todos os participantes conectados à aplicação. A terceira trata as pesquisas baseando-se no modelo de roteamento da rede P2P. Assim, as pesquisas são encaminhadas de maneira serializada pela rede, baseada na similaridade das chaves de pesquisa armazenadas nas tabelas de roteamento dos participantes. Essas três grandes vertentes são detalhadas a seguir.

(A) Índices Centralizados e Repositórios:
Esse mecanismo de descobrimento de recurso é utilizado em arquiteturas híbridas, onde parte do sistema é centralizado e a troca de recursos é realizada de forma distribuída. Nesse modelo, os participantes se conectam a um serviço centralizado que armazena todas as informações a respeito da localização e do uso dos recursos do sistema. Quando um participante realiza uma busca no servidor centralizado, um casamento pelo padrão é realizado e o participante escolhe algum (ou alguns) outro(s) participante(s) para realizar a parceria. A escolha pelo parceiro pode se dar por qualquer forma de distinção entre os parceiros candidatos, como por exemplo, o mais barato, o mais rápido, o mais perto, o mais disponível, e assim por diante. A troca de recursos é realizada diretamente entre os participantes, sem a necessidade de intermédio de algum outro nodo.

O Napster [27, 62], uma das aplicações P2P pioneiras em compartilhamento de arquivos, utiliza esse método. Os seus servidores armazenam um índice com metadados de todos os arquivos compartilhados na rede, além dos dados dos usuários participantes. Quando um novo usuário torna-se um participante do sistema, ele contata o servidor central e se registra, informando seu endereço e os metadados de seus arquivos. Em uma busca, o participante envia sua requisição ao servidor. Os servidores do Napster devolvem uma lista de usuários que podem atender à requisição. Assim, o participante abre uma conexão direta com o parceiro escolhido.

(B) Inundação de Consultas:
Esse é um modelo P2P puro, onde os participantes não mantêm contato com nenhum servidor centralizado para obter informações para buscas. Cada participante publica informações sobre o seu conteúdo compartilhado na rede. Assim, como nenhum participante conhece todos os recursos da rede, eles necessitam enviar consultas aos demais participantes para descobrir a localização do recurso desejado. Cada consulta é enviada aos parceiros do participante; esses parceiros reenviam aos seus parceiros, e assim por diante; até que alguma condição interrompa o processo de inundação de pesquisas. Essa condição de parada pode ser a localização do recurso ou um número máximo de reenvio de mensagens.

A inundação de mensagens é utilizada pela arquitetura original do Gnutella [25], para realizar as pesquisas por arquivos em sua rede. No Gnutella, quando um novo participante se junta à rede, este envia mensagens para quaisquer participantes que ele consiga se conectar. Esses participantes, candidatos à parceria inicial, são descobertos através de um mecanismo central. Os participantes que recebem o pedido de parceria devolvem uma mensagem com sua apresentação e também propagam a apresentação do novo participante. Para evitar sobrecarga da rede, o Gnutella emprega um mecanismo

de máximo número de saltos que uma mensagem pode dar. Assim, as mensagens apresentam um raio de circulação, a partir do participante que iniciou a inundação.

O mecanismo de inundação pode gerar resultados eficientes em uma rede com um número pequeno ou médio de usuários [53]. Em redes maiores, esse mecanismo não escala bem e, não há garantias de descoberta acurada dos recursos. Além disso, o mecanismo de número máximo de saltos pode criar partições na rede, o que faz com que um participante tenha acesso a um horizonte delimitado de parceiros ou recursos.

(C) Modelos Baseados em Roteamento:
O modelo baseado em roteamento adiciona uma estrutura para organizar a maneira com que os recursos são armazenados. Geralmente, utilizam tabelas "hash" para identificar os recursos e criar o mecanismo de rota. Assim, esse modelo cria um mapeamento entre o recurso armazenado na rede e a sua localização, sob a forma de uma tabela de roteamento. Dessa forma, as consultas por recursos podem ser roteadas de maneira eficiente ao participante que dispõe do recurso desejado. Essa abordagem pode reduzir o número de saltos que uma mensagem precisa realizar na rede, até atingir o seu destino (a localização do recurso).

O mecanismo de busca é implementado através da organização dos participantes em uma rede sobreposta estruturada, e pelo roteamento da mensagem na rede sobreposta até o participante responsável pelo conteúdo [24, 65]. Várias propostas adotam a implementação da busca baseada no roteamento da rede, por exemplo, Freenet [12], Chord [82], CAN [70] e Pastry [74].

2.1.3 Exemplos de Sistemas P2P

Freenet:
O Freenet [12] é uma aplicação que provê um serviço de armazenamento de arquivos, ao invés de compartilhamento puramente descentralizada. Sua arquitetura é fracamente estruturada e opera como uma rede P2P auto-organizável.

Nesse sistema, cada participante tem um identificador e conhece um subconjunto de outros participantes da rede. Quando um documento é adicionado ao Freenet, um identificador é atribuído a esse documento, baseado no seu nome e em um valor "hash". Cada participante irá rotear o documento para o participante com identificador mais similar ao identificador do documento. Esse processo é repetido até que o participante com o identificador mais semelhante seja o participante atual na operação de roteamento. Assim, cada operação de roteamento assegura a manutenção de uma cópia local do documento. O processo de busca é semelhante ao descrito acima, e

quando um participante necessita de um documento, suas pesquisas são roteadas até o participante com identificador mais semelhante ao conteúdo desejado.

Gnutella:
O Gnutella [25] é um sistema para compartilhamento de arquivos que, originalmente, utiliza inundação de mensagens na busca por conteúdo. Sua arquitetura também utiliza o conceito de *"supernodos"* [65], ou seja, participantes com um maior poder computacional que fazem parte da rede, formando uma estrutura de suporte aos demais participantes. Dessa maneira, quando um participante realiza uma busca, esses supernodos podem auxiliar e retornar os resultados da pesquisa que estão em seu armazenamento temporário (cache).

O processo de busca por inundação do Gnutella é tolerante à dinâmica dos participantes (entrada e saída da rede), porém, esse mesmo mecanismo não se mostra escalável, e gera sobrecargas inesperadas para a rede [53].

Para se juntar ao sistema Gnutella, um novo participante se conecta inicialmente a um dos diversos participantes disponíveis em um servidor de endereços centralizado. Uma vez conectado a um dos outros participantes da rede, o novato envia mensagens para interagir e descobrir mais parceiros.

FastTrack:
A rede P2P FastTrack é um sistema descentralizado de compartilhamento de arquivos. Essa rede utiliza uma estrutura organizada de supernodos para realizar buscas mais rápidas e eficientes. Os demais participantes do sistema (participantes que não são supernodos) transmitem os metadados de seus arquivos compartilhados aos supernodos. A busca neste sistema é semelhante ao sistema de busca utilizado pelo Gnutella, onde uma inundação de mensagens é enviada pela rede e os resultados armazenados no supernodos podem reduzir o tempo de resposta. A rede P2P do FastTrack pode existir sem a existência dos supernodos, mas como esperado, os tempos de respostas para consultas podem se elevar consideravelmente. Uma das aplicações mais importantes que utiliza a rede FastTrack é o KaZaA [43].

BitTorrent:
Atualmente, o BitTorrent [5] é uma das aplicações mais populares para compartilhamento de arquivos. Ele utiliza um mecanismo centralizado para gerenciar a descarga dos arquivos realizadas pelos participantes do sistema. O BitTorrent usa uma abordagem de incentivo baseada em trocas, conhecido como *"tit-for-tat"*. Nesse mecanismo de incentivo, os participantes do sistema compartilham dados somente com

os parceiros que também tenham interesse em compartilhar. Assim, um participante conseguirá boas parcerias e receberá os dados desejados, somente se atuar de forma altruísta e como um bom parceiro.

Desta forma, o BitTorrent desencoraja usuários pouco altruístas, conhecidos como *"free-riders"*. Os participantes do sistema com altas taxas de envio de dados, provavelmente, conseguem mais recursos no sistema, realizando descargas de arquivos mais rápidas. A velocidade de descarga poderá ser reduzida caso o participante limite sua capacidade de compartilhamento. Esse mecanismo também assegura que um arquivo se espalhe mais rapidamente entre os participantes da rede do BitTorrent.

A arquitetura do BitTorrent apresenta um participante centralizado denominado *"tracker"*. Cada usuário que deseja obter o arquivo da rede P2P se conecta a esse participante para obter informações a respeito da rede. O endereço do *"tracker"*, assim como os metadados do arquivo desejado, são obtidos a partir de um arquivo proprietário da aplicação BitTorrent, geralmente, com a extensão *torrent*.

O *"tracker"* mantém informações sobre todos os envolvidos no compartilhamento dos arquivos. Ele gerencia os participantes capturando informações, como seus endereços de rede, as partes do arquivo presentes e desejadas por cada um. Assim, quando um novo participante se junta à rede, este obtém junto ao *"tracker"* uma lista de endereços dos seus parceiros candidatos. Assim, os participantes se conectam, estabelecem parcerias, e trocam dados, enquanto participam da rede BitTorrent.

CAN - Rede de Conteúdo Endereçável:
O CAN [70] (Rede de conteúdo endereçável) apresenta uma tabela do tipo *"hash"* que mapeia os nomes de arquivos a sua localização na rede. Cada participante da rede CAN armazena partes da tabela *"hash"*, e armazena também parte das informações das tabelas de alguns participantes adjacentes. As requisições para inserção, remoção e busca de uma chave (um arquivo) são roteadas através dos participantes adjacentes; até chegar ao participante responsável pela tabela dessa chave.

Quando um novo participante se junta ao CAN, é alocado a ele uma porção de chaves. Isso é feito pela divisão de um conjunto de chaves alocadas a um participante já existente no sistema. Quando um participante deixa o CAN, a porção de chaves que era de sua responsabilidade é atribuída a algum de seus parceiros.

Chord:
O Chord [82] é um sistema onde os participantes mantêm uma tabela de roteamento distribuída sob a forma de um anel lógico. Nesse anel, os participantes são responsáveis por partes específicas da tabela de roteamento e do conteúdo da rede P2P.

Os participantes também são identificados por chaves. As chaves atribuídas aos arquivos e aos participantes são geradas a partir de uma variante de *"hash"* [1, 42]. A única informação de roteamento necessária para cada participante é a localização de seu sucessor na cadeia de participantes. As pesquisas são encaminhadas por esse anel, até que o participante que contenha a chave pesquisada seja identificado.

Quando um novo participante se junta à rede, a ele é atribuída uma chave e parte das chaves de arquivos que seu sucessor possui serão atribuídas ao novato. Quando algum participante deixa a rede, o seu sucessor no anel lógico irá se responsabilizar pelas suas chaves.

Tapestry:
O Tapestry é baseado na localização dos participantes e em mecanismos de roteamento de mensagens [107]. A sua estrutura distribuída de dados permite que os participantes localizem os recursos (arquivos) e encaminhem as mensagens através da rede, para um número arbitrário de participantes na rede sobreposta.

A topologia da rede Tapestry é auto-organizada, à medida que os participantes se juntam e abandonam o sistema. As informações de localização e roteamento são distribuídas entre os participantes da rede e a consistência da topologia da rede é verificada durante sua existência. As reconstruções, caso sejam necessárias (e.g. falhas), são facilmente realizadas [1].

Cada participante mantém um mapa de vizinhos e esse mapa contém vários níveis. Por exemplo, o nível 1 contém os endereços dos participantes que o identificador combine com 1 dígito. Cada entrada na tabela de vizinhos é um apontador para o participante mais próximo da rede que o identificador combina com o mapa de vizinhos.

2.2 Distribuição de Mídia Contínua ao Vivo em Arquiteturas P2P

Esta seção apresenta uma revisão dos esforços prévios no sentido de estruturar e organizar os sistemas de transmissão ao vivo em arquiteturas P2P. São discutidas as duas principais vertentes utilizadas para enviar um fluxo de mídia contínua ao vivo em P2P. Uma abordagem é baseada em estruturas de árvores e com uma forte estruturação.

A outra abordagem é baseada em malha e não apresenta uma estrutura rígida entre seus participantes.

As aplicações de vídeo na Internet têm atraído um grande número de usuários na Internet recentemente. Somente o Youtube, um dos hospedeiros mais populares de conteúdo de vídeo, hospedava em agosto de 2006, mais de 45 terabytes de vídeos. Além disso, o Youtube atraiu mais de 1.73 bilhões de visualizações nesta época [50, 102]. Alguns relatórios recentes apontam que esse hospedeiro de vídeos é responsável por mais de 10% de todo o tráfego de Internet da América do Norte [22].

Os vídeos na Internet pode se classificar em duas grandes categorias: vídeos ao vivo (*live*) e vídeos pré-armazenados, enviados sob demanda (*on-demand*). Os usuários de vídeos assistidos sob demanda têm a flexibilidade de assistir um conteúdo previamente armazenado, da maneira que eles querem e no momento desejado. De forma contrária, um conteúdo ao vivo é transmitido no mesmo momento em que o fluxo é gerado. Logo, todos os usuários devem estar sincronizados e devem assistir o fluxo de vídeo ao mesmo tempo.

A solução básica para o envio do fluxo de vídeo na Internet é a utilização do modelo cliente-servidor. Nesse modelo, um cliente cria uma conexão com um servidor de vídeo e o conteúdo é enviado para o cliente diretamente do servidor. Existem algumas variantes deste modelo, mas as soluções baseadas em cliente-servidor demandam uma larga banda no servidor, o que gera um alto custo operacional [50].

Recentemente, vários sistemas P2P foram desenvolvidos para prover conteúdo de vídeo ao vivo e sob-demanda na Internet, com baixo custo operacional [8, 39, 66, 69, 81, 103, 103, 105]. As redes entre pares (P2P) emergiram como um novo paradigma para construir aplicações distribuídas [50]. Neste tipo de aplicação, os usuários são encorajados para atuarem como clientes e servidores. Em uma rede P2P, os participantes, além de obterem serviços da rede, também os provêm. Assim, a banda de rede dos usuários finais é utilizada para reduzir a grande demanda por banda de rede, outrora necessária aos servidores.

Os sistemas de envio de vídeo que utilizam arquitetura P2P podem ser classificados em duas categorias quanto sua estrutura: podem ser baseados em uma estrutura de árvore ou em malha. As seções a seguir descrevem e discutem o funcionamento de cada uma destas estruturas.

2.2.1 Estrutura Baseada em Árvore

Sistemas baseados em árvore têm uma estrutura sobreposta bem organizada e, tipicamente, distribuem o fluxo de vídeo enviando dos nodos para seus filhos. Um

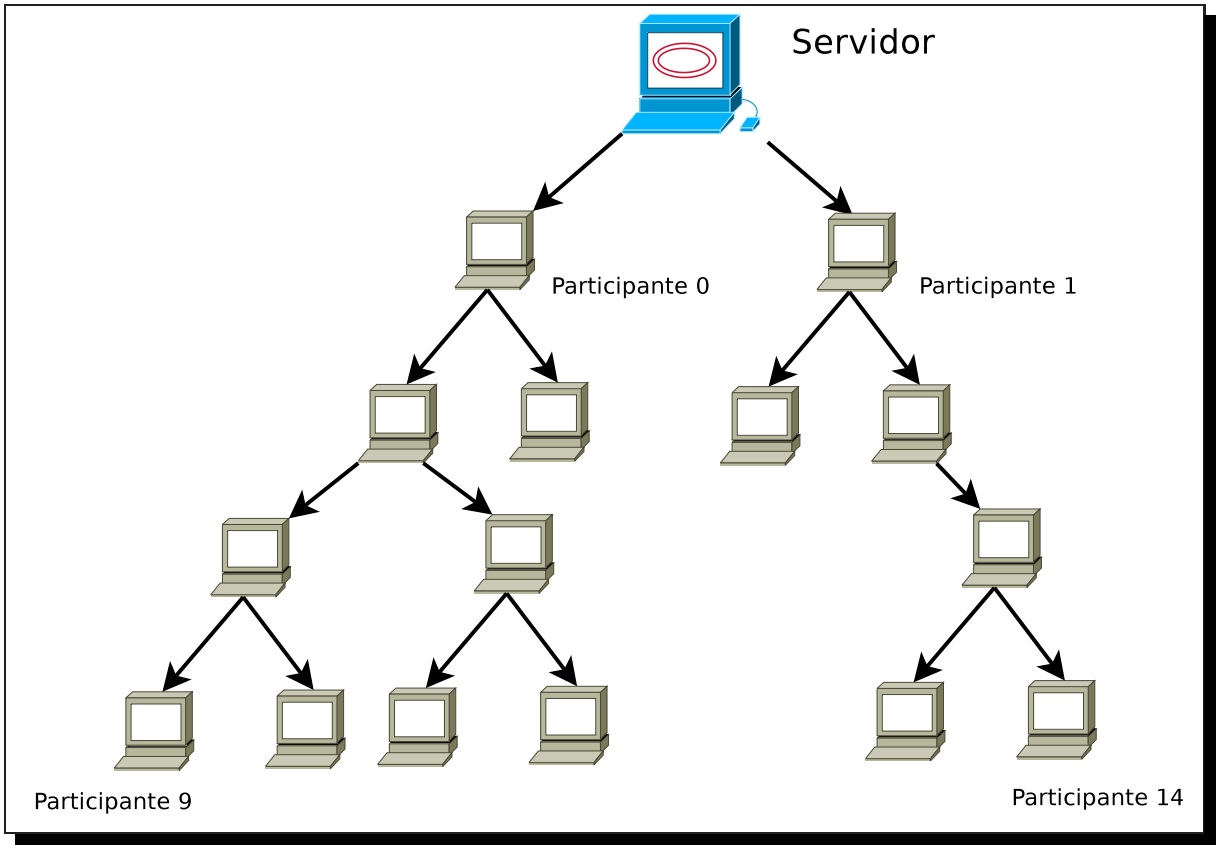


Figura 2.1: Árvore de multicast em nível da camada de aplicação.

dos maiores problemas desta abordagem é que são vulneráveis à entrada e abandono dos participantes da rede (*churns*) [18, 57]. Assim, quando um participante deixa a rede, a estrutura de árvore se rompe, e parte do sistema sofre, temporariamente, uma ruptura no fluxo do vídeo.

Uma maneira eficiente de se estruturar e enviar um fluxo de vídeo a um grupo de usuários na Internet seria a utilização de *multicast* no nível de IP [50]. Em uma sessão de *multicast* IP uma estrutura de árvore é formada. A fonte de vídeo se torna a raiz desta árvore *multicast*, e os clientes recebem o fluxo de vídeo através dos vários nodos desta árvore, formado pelos roteadores que suportam o *multicast* em nível de IP.

Para contornar a falta de suporte de *multicast* em nível de IP, a função equivalente tem sido implementada no nível da camada de aplicação. Os servidores de vídeo e os usuários formam uma rede sobreposta à rede real, e assim se, organizam para distribuir o fluxo de vídeo. De maneira similar ao *multicast* IP, formado por uma árvore de roteadores no nível de rede, os participantes da sessão de vídeo formam uma árvore na camada de aplicação, cuja origem é o servidor de vídeo.

Cada usuário do sistema se conecta à árvore em um certo nível. Ele recebe o

vídeo de seus pais, no nível superior, e reenvia o conteúdo aos seus filhos, no nível mais baixo. Algumas aplicações, como Overcast [39], utilizam esta abordagem. A figura 2.1 ilustra um sistema com quinze nodos participantes.

Existem várias maneiras possíveis de se construir a árvore para o envio de fluxo de vídeo. Deve-se considerar a altura da árvore e a quantidade de filhos de cada nodo da árvore. Nodos em níveis inferiores da árvore recebem o fluxo de vídeo após ele percorrer vários outros nodos, e isto pode induzir a grandes latências. Para reduzir esse problema, deve-se preferir uma árvore com o mínimo de níveis possível, o que pode requerer usuários com grande largura de banda, retransmitindo para vários filhos.

Tão importante quanto a construção da árvore é manutenção da sua estrutura. Os usuários de uma aplicação de vídeo em sistemas P2P podem ser muito dinâmicos, entrando e deixando a rede de forma muito imprevisível. Quando um nodo abandona a aplicação de transmissão de fluxo contínuo em P2P, ele interrompe a transmissão, e todos os seus descendentes ficam sem uma fonte do fluxo de vídeo. Para reduzir essas interrupções, a árvore de envio de fluxo de vídeo deve ser reconstruída o mais rapidamente possível. A Figura 2.2 ilustra um cenário em que um nodo deixa o sistema de vídeo e a árvore de *multicast* ao nível de aplicação que deve ser reconstruída.

A construção e manutenção da árvore de envio de fluxo P2P pode ser realizada de maneira centralizada ou descentralizada. Em uma abordagem centralizada, um servidor controla a construção da árvore e sua recuperação. Para grandes sistemas de envio de vídeo, uma abordagem centralizada pode se tornar um gargalo e um ponto de falha [50]. Vários algoritmos distribuídos abordam e tratam o problema de manutenção e construção da árvore de maneira distribuída [89]. Mesmo assim, uma abordagem baseada em árvore não consegue se recuperar de maneira rápida o suficiente para lidar com a dinâmica dos participantes, pois a constante interrupção do fluxo e a reconstrução da árvore de envio de fluxo contínuo podem causar uma sensação de baixa qualidade no serviço oferecido [18, 50, 57].

Outro problema encontrado ao se usar uma árvore simples é que os nodos, que estão na folha da árvore, acabam por não contribuir com o sistema. Assim a utilização de banda não é totalmente aproveitada. Uma vez que existe um grande número de nodos folhas, a capacidade da árvore se torna subestimada. Para lidar com esse problema, foram propostas abordagens baseadas em múltiplas árvores como em [8]. Nesta abordagem, um servidor divide o fluxo de vídeo em vários subfluxos e para cada um destes, uma árvore *multicast* ao nível de aplicação é construída. Cada participante deve se conectar a todas as árvores criadas, para obter um fluxo de vídeo completo. Preferivelmente, os participantes se conectam em lugares diferentes nos vários níveis existentes. Assim, os nodos folhas de uma árvore, podem se tornar nodos internos em

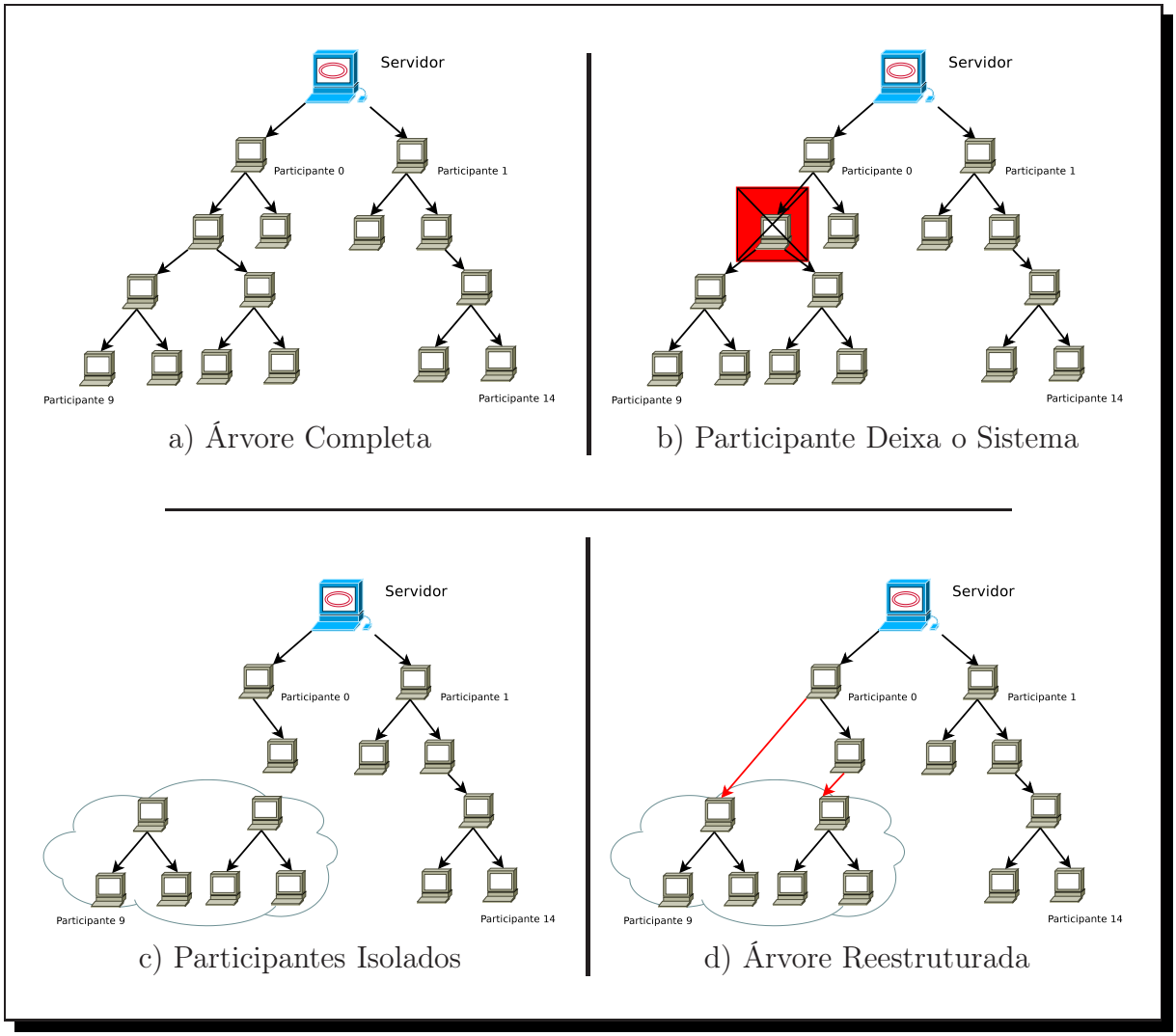


Figura 2.2: Manutenção da árvore de multicast em nível da camada de aplicação.

outra, fazendo melhor uso da capacidade disponível. A Figura 2.3 ilustra uma aplicação de envio de fluxo de vídeo com duas árvores.

2.2.2 Estrutura Baseada em Malha

Em uma estrutura baseada em malha (*mesh-based*), os participantes não se organizam em uma topologia estática. As relações são estabelecidas baseando-se nos recursos disponíveis momentaneamente. Um participante se conecta a um subconjunto de outros participantes do sistema e, periodicamente, eles trocam informações. Os dados são buscados nos participantes que já os têm. Como um participante tem múltiplos vizinhos ao mesmo tempo, a organização em malha é robusta à dinâmica dos nodos. Entretanto,

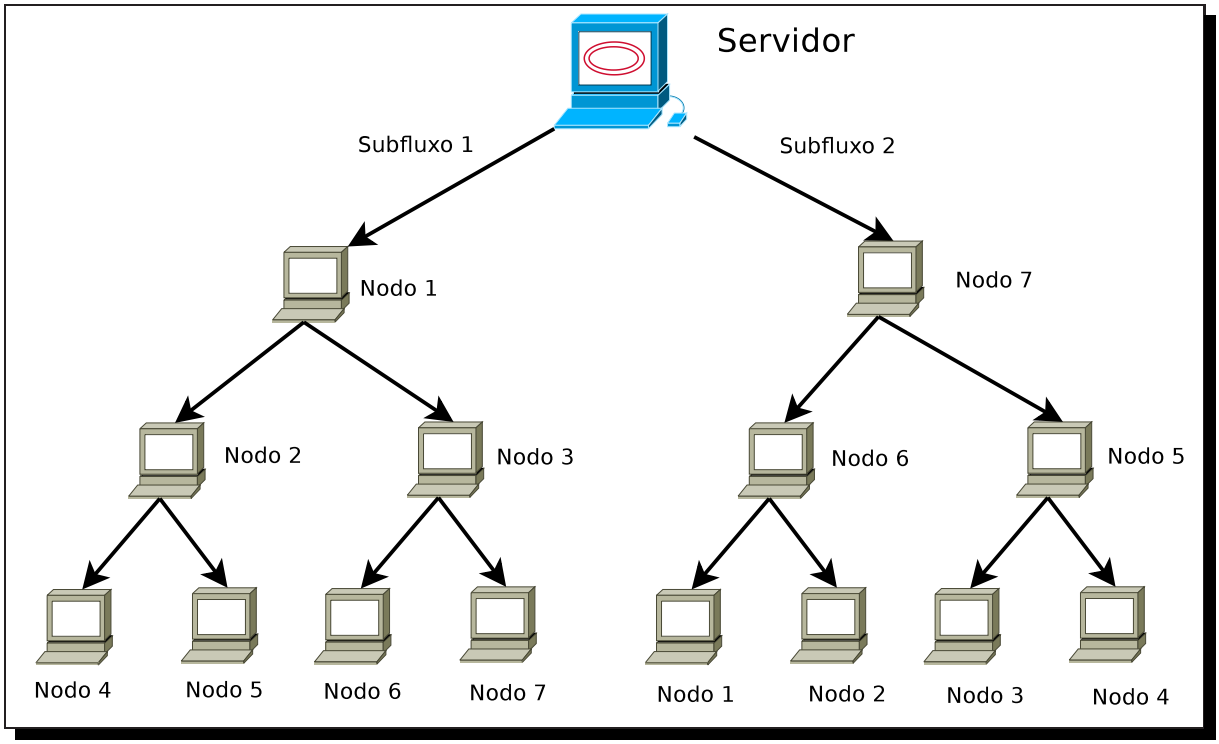


Figura 2.3: Sistema baseado em múltiplas árvores com dois subfluxos.

essa relação dinâmica faz com que a distribuição de vídeo se torne imprevisível.

Diversos trabalhos recentes na área de fluxo contínuo P2P adotam uma estrutura baseada em malha [35, 99, 103, 106]. Em um sistema desse tipo, não existe uma topologia fixa da rede P2P. Os nodos estabelecem suas conexões dinamicamente, de acordo com seus interesses. Os participantes sempre mantêm parcerias com vários outros vizinhos. Eles podem fazer envio ou recepção de dados de múltiplos parceiros e, se um participante deixa o sistema, seus vizinhos continuam recebendo o conteúdo desejado dos demais nodos, com os quais eles mantêm contato. Caso seja do interesse de um participante, ele poderá encontrar novos parceiros para manter um nível de conectividade alto. Um alto grau de conectividade faz com que a estrutura em malha torne-se robusta à dinâmica dos participantes do sistema. Trabalhos recentes, como [57], mostram que uma estrutura baseada em malha tem um desempenho superior que uma estrutura baseada em árvores.

De maneira similar ao que acontece a um dos sistemas de compartilhamento de arquivos mais populares, o Bittorrent [5], uma estrutura em malha, tem um servidor centralizado. Esse servidor mantém uma lista dos participantes ativos na sessão de vídeo. Quando um usuário junta-se à aplicação de distribuição de mídia contínua ao vivo, ele contata este servidor e se cadastra. O servidor de *bootstrap*, *rendevouz* ou

tracker, como costuma ser chamado [35, 50, 103], retorna ao novo participante uma lista com informação de um subconjunto aleatório de participantes da sessão de vídeo.

Após receber a lista com os possíveis parceiros, o novo participante tenta realizar as parcerias. Se a parceria é aceita pelo nodo contatado, o novo participante irá adicioná-lo a sua lista de vizinhos. Depois de obter alguns vizinhos, o novo participante começa a trocar pedaços de vídeo com seus parceiros. A Figura 2.4 mostra o processo inicial de cadastro no sistema e realização das parcerias iniciais.

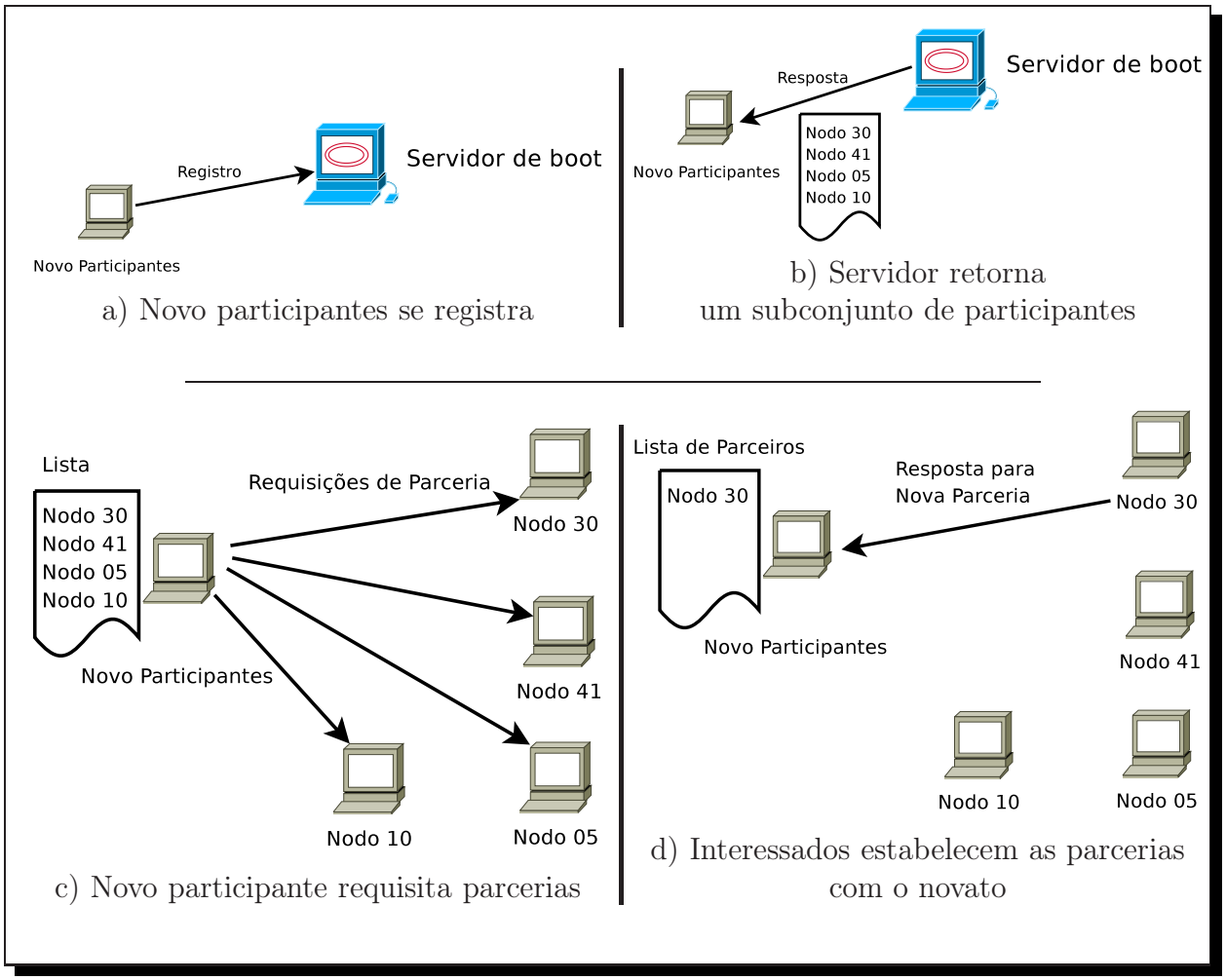


Figura 2.4: Atividade inicial de um novato - rede P2P baseada em malha.

Para lidar com a dinâmica na rede P2P, ou seja, as frequentes chegadas e partidas dos usuários, um participante sempre se mantém atualizado sobre o estado de seus parceiros. Caso o número de conexões ou o nível de serviço caia, ele pode recorrer ao servidor centralizado para obter novas listas de possíveis parceiros. Também pode tentar achar novos parceiros enquanto troca dados e mensagens de controle.

Os participantes do sistema trocam regularmente mensagens de informação de

vida (*keep-live messages* ou *ping*). Caso um vizinho não responda às mensagens de vida, um participante o remove da lista e, possivelmente, tenta obter novos parceiros para manter sua conectividade [103]. Uma parceria é estabelecida por um acordo mútuo entre os participantes. Os diferentes sistemas existentes possuem estratégias variadas para estabelecimento destes acordos. Por exemplo, o número de vizinhos que os participantes possuem, a banda de rede disponível, a dinâmica dos seus vizinhos e a qualidade percebida do fluxo de vídeo [50]. Com base nesses critérios, um participante se conecta a um novo vizinho e também procura por novas parcerias.

Em uma estrutura baseada em árvore, o fluxo de vídeo é transmitido a partir de uma fonte geradora para todos os participantes do sistema, seguindo a estrutura lógica da árvore formada. Em uma estrutura baseada em malha, não existe um fluxo contínuo transmitido nestes mesmos moldes. Nesses sistemas, a fonte do vídeo (servidor) faz a codificação e a divisão do vídeo, criando os pequenos pedaços chamados *chunks*. Cada *chunk* contém dado para um pequeno intervalo de tempo de visualização. Por exemplo, as aplicações atuais transmitem dados a uma taxa aproximada de 6 *chunks* por segundo de vídeo [20]. Esses *chunks* são numerados em uma sequência temporal, para que os participantes possam identificar e executar o vídeo correspondente de forma apropriada. Os pedaços do fluxo são disseminados a partir do servidor para diversos participantes da rede, que os disseminam para seus companheiros, e assim por diante. Como os *chunks* tomam diferentes caminhos para atingir os diversos pontos da rede, eles chegam a um usuário fora de ordem e, para uma execução contínua do vídeo, os participantes guardam os *chunks* em um armazenamento temporário de memória, onde são ordenados antes de sua apresentação. Dependendo do tipo de aplicação, o armazenamento pode variar de segundos a minutos. Em uma sessão de vídeo ao vivo, que é o período em que um fluxo de mídia é transmitido, a sequência de identificação dos *chunks* cresce enquanto o vídeo é disseminado.

Os dados são trocados principalmente através de duas estratégias: requisitando ou enviando (*pull* e *push*). Em um sistema do tipo *mesh-push* (malha e requisição), um usuário envia os dados que recebe aos seus vizinhos que provavelmente ainda não os obtiveram. Não há uma relação clara de pai-filho neste esquema e o envio dos dados é estabelecido por interações passadas entre os participantes, onde indicam quais são os dados desejados. Um participante pode estabelecer parcerias com diversos outros e, anunciar a necessidade por dados a todos estes. Por consequência, pode existir envio de dados redundantes na rede, pois mais de um dos parceiros pode responder por um pedido. Para tratar esse problema deve existir um planejamento entre os participantes do sistema, com escalonamento das transferências dos dados [103].

Caso seja usado um sistema *mesh-pull*, os participantes, periodicamente, trocam

entre si um mapa de *chunks*. Este mapa tem informações dos *chunks* disponíveis localmente por um participante. Contém também informações sobre os dados faltantes. Ao obter os mapas de seus vizinhos, um participante decide como escalonar o pedido de *chunks* (e a qual vizinho enviar o pedido). As transmissões redundantes são evitadas, uma vez que os participantes solicitam *chunks* a um único parceiro. Porém, as frequentes trocas de mapas de *chunks* e mensagens por pedidos aumentam a sobrecarga do protocolo e podem introduzir novos atrasos ao sistema.

A Figura 2.5 ilustra a troca de *chunks* em uma aplicação com estrutura baseada em malha. Por esta figura, o nodo 2 gera seu mapa, indicando quais *chunks* ele tem disponível em seu armazenamento temporário. Ele troca este mapa com os participantes 1 e, como resposta, o nodo 1 envia o seu mapa. Observe que o nodo 1 possui uma lista com os diversos mapas de seus parceiros. Os pedaços de vídeo faltantes no nodo 2, e que o nodo 1 possui, serão requisitados. Finalmente, o nodo 1 responde às requisições pelo nodo 2.

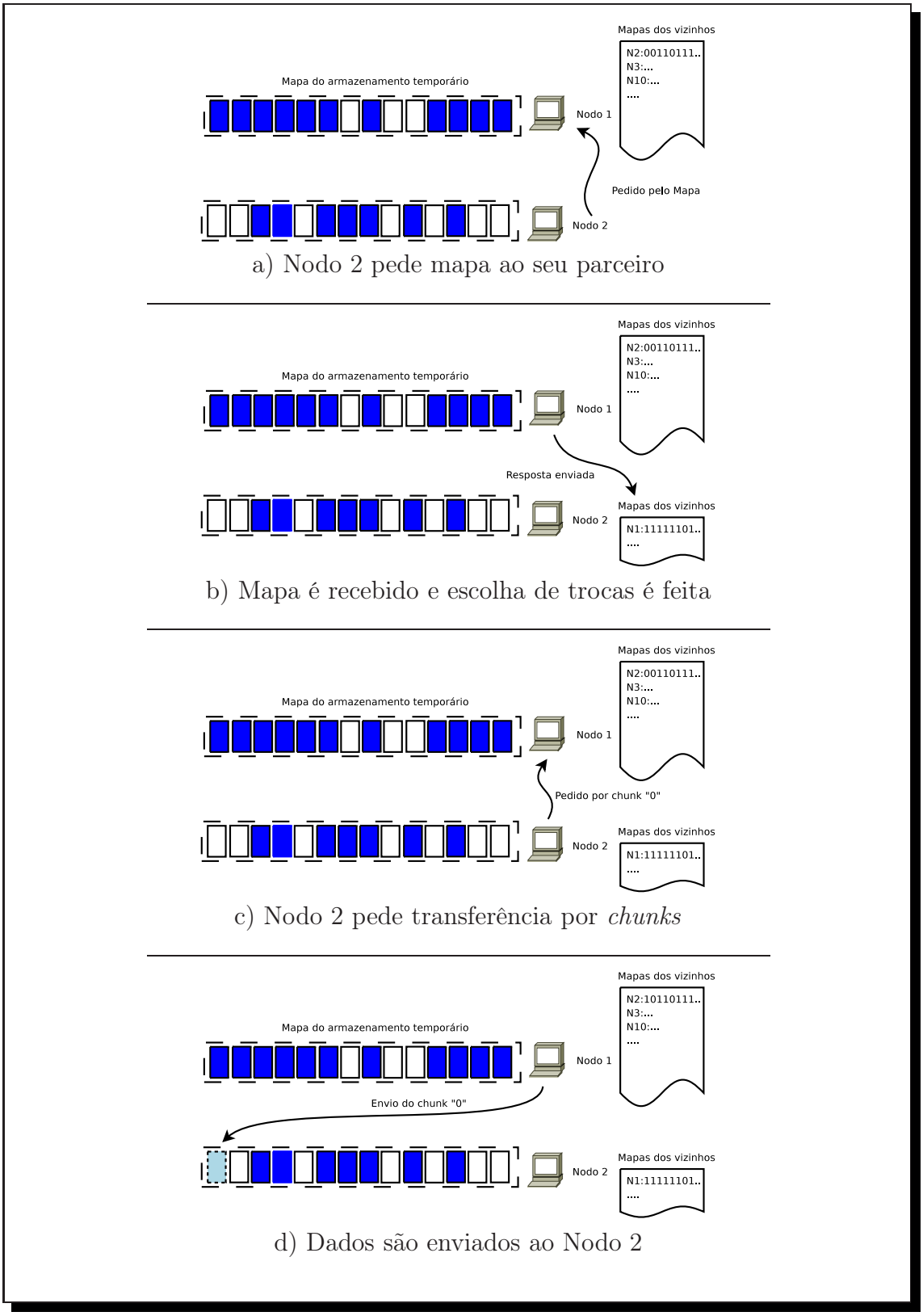


Figura 2.5: Troca de dados na aplicação baseada em malha.

2.2.3 Estrutura Híbrida

Uma estrutura híbrida para transmissões ao vivo em P2P pode ser caracterizada de duas formas. Na primeira, a arquitetura da rede é um misto entre uma arquitetura baseada em árvores e uma arquitetura baseada em malhas. Na segunda, o método de transmissão de dados entre os participantes é um misto entre um sistema P2P, orientado por pedidos explícitos por dados, e um encaminhamento automático dos dados da mídia. Em ambos os casos, há uma tentativa de se obter os benefícios de cada uma das propostas e isolar os pontos fracos das mesmas.

2.2.3.1 Híbrido de Árvore-Malha

Em uma rede sobreposta P2P baseada em árvore, os participantes da rede são organizados de forma hierárquica. Assim, a transmissão ao vivo flui dos níveis mais altos na hierarquia (do participante que está codificando o vídeo) para os níveis mais baixos. Os participantes mais próximos à fonte apresentam menores latências no vídeo assistido e menos problemas com relação a rupturas na hierarquia da árvore. Além disso, com o uso de uma única árvore, os participantes no nível mais baixo (folhas), não contribuem com o sistema. Isso pode diminuir a escalabilidade do sistema de transmissão ao vivo.

As estruturas baseadas em malha contornam o problema de rupturas na árvore. Nesse modelo, os participantes do sistema realizam parcerias e trocam dados entre si. Não há hierarquias e assim, todos os participantes colaboram com o sistema o que torna a utilização dos recursos do sistema mais eficiente. Entretanto, como não há uma estrutura bem definida, a recepção dos dados da transmissão ao vivo está sujeita a atrasos e imprevisibilidade [37].

Uma abordagem de construção híbrida da rede sobreposta adota partes da rede como uma árvore, e outras partes como uma rede em malha. Os participantes do sistema podem participar de ambas as estruturas. Sistemas como [38] adotam estratégias de alocação dos participantes na árvore e, na rede em malha formada, adotam estratégias para otimizar o agendamento de entrega de dados. Alguns dos critérios utilizados para alocar os participantes na árvore são estabilidade do participante na rede e proximidade entre os participantes na rede física.

Mais precisamente, o Anysee2 [38] estrutura seus participantes em uma rede de controle e em uma rede de troca de dados. A rede de controles é baseada em uma árvore, enquanto a rede de troca de dados é baseado em uma malha.

2.2.3.2 Híbrido por Encaminhamento Automático / Pedidos Explícitos (Push-Pull)

O método híbrido para obtenção de dados utiliza duas formas em conjunto para encaminhar/receber a mídia transmitida: O encaminhamento automático da mídia (utilizado em uma estrutura de árvores) e pedidos explícitos pelos dados (utilizado em uma estrutura em malha). Nesse caso, abordagens Push (encaminhamento automático) e Pull (pedido explícita), são utilizadas em uma rede P2P não estruturada. Dessa forma, esses sistemas quase sempre apresentam um protocolo/estrutura simples, sem a necessidade de coordenação e hierarquia entre participantes. Isso torna o sistema naturalmente resistente à dinâmica (*churn*) dos participantes e a outros imprevistos.

Existem alguns mecanismos propostos com a combinação do *"push-pull"* [52, 104] Esses mecanismos usam o *"push"* para espalhar os dados rapidamente e o *"pull"* para preencher as lacunas dos dados recebidos. Nesses dois trabalhos supracitados, ambos os mecanismos coexistem, não havendo uma alternância entre eles.

O protocolo proposto em [51] alterna as operações de *"push"* e *"pull"*. Cada participante é autônomo e independente, sem a necessidade de sincronia com outros participantes. Durante a operação de *"push"*, o participante envia dados para algum ou alguns de seus parceiros. Durante a operação de *"pull"*, o participante busca por dados que ele necessita localmente.

A utilização do mecanismo de *"push-pull"*, como mostrado em [46], pode levar a uma redução da sobrecarga do tráfico da rede. Os resultados nesse trabalho mostram que, em comparação com um sistema do tipo *"mesh-pull"* e com o GridMedia [108], houve uma redução da sobrecarga de rede de 33% e 37% respectivamente. Além disso, o sistema com a abordagem híbrida alcançou resultados com latência e taxa de execução do vídeo melhores que os sistemas comparados.

2.3 Ataques aos Sistemas P2P

2.3.1 Ataques e Comportamentos Indesejados em Transmissão ao Vivo

Com o aumento da popularidade das aplicações de envio de mídia ao vivo em P2P, vários estudos foram realizados com o objetivo de encontrar maneiras de organizar e coordenar o funcionamento de tais aplicações [35, 37, 69, 89, 103, 105]. Essas diversas maneiras de organização do sistema, e protocolos de comunicação, têm como principal objetivo alcançar uma alta escalabilidade quanto ao número de participantes do sistema. Além disso, essas abordagens visam à qualidade do vídeo transmitido, com baixa latência e pequeno número de falhas.

Mesmo com esse crescimento, durante o planejamento e o desenvolvimento desses sistemas de transmissão ao vivo em P2P e de seus protocolos, não são levados em consideração o comportamento de seus participantes. Geralmente, assume-se que eles (os participantes) terão um comportamento altruísta e não malicioso. Dessa forma, os objetivos desejados na criação de um sistema P2P para transmissão ao vivo podem não ser alcançados em razão de atitudes maliciosas e/ou oportunistas.

Porém, alguns sistemas semelhantes dispensam atenção a esses comportamentos indesejados. Por exemplo, existe uma série de trabalhos relacionados a ataques e comportamentos oportunistas em aplicações de compartilhamento de arquivos em P2P [14, 15, 19, 91]. Mas, mesmo havendo semelhanças entre compartilhamento de arquivos e transmissão ao vivo em P2P, não há garantias de sucesso ao adotar as soluções de um sistema diretamente no outro. Há um risco em acreditar que conclusões válidas para uma aplicação baseada em P2P sejam compartilhadas com outras aplicações P2P [90], pois, as premissas entre desses sistemas se diferem, assim como os requisitos de qualidade de serviço.

Por esses motivos, fica evidente a importância de medidas de identificação e combate ao comportamento malicioso/oportunista nos sistemas de transmissão ao vivo em P2P. Mais ainda, o tratamento dado a esses comportamentos indesejados deve ser condizente com o caractere ao vivo da aplicação de transmissão. As medidas devem surtir efeito em pouco tempo e com grande eficiência para que a aplicação não seja comprometida, perdendo qualidade e aumentando a latência.

Neste trabalho, somente os comportamentos indesejados que podem ser monitorados pelo comportamento passado dos participantes do sistema são levados em consideração. Destaca-se nesse grupo o envio de conteúdo poluído (poluição de conteúdo). Este ataque pode ter um impacto negativo forte em uma transmissão

ao vivo. Nele, um usuário malicioso (ou vários) altera o conteúdo que está sendo transmitido, repassando aos demais participantes uma mídia falsa ou inválida.

Como a grande maioria dos sistemas para transmissão ao vivo em P2P assume um comportamento altruísta e não malicioso de seus participantes [20], esses sistemas não consideram que o conteúdo transmitido possa ser alterado ou forjado. Assim, a ocorrência de ataques de poluição pode ser facilitada pela falta de medidas de combate no sistema. Mais ainda, conforme observado em [20], esse ataque pode levar ao abandono do sistema de vídeo por grande parte de seus usuários, por insatisfação quanto à mídia recebida.

Entre outras ações maliciosas e comportamentos indesejados, destacam-se:

1. **Ataque de eclipse**: em uma rede sobreposta, se um atacante controla uma fração dos parceiros de um participante legítimo, ele pode ocultar este participante legítimo da rede P2P. Por exemplo, o atacante pode refazer um roteamento das mensagens de/para esse participante; pode inutilizar as mensagens do participante e, até mesmo, se anunciar como o próprio participante legítimo [80]. Caso este atacante também dissemine conteúdo inválido, todo o fluxo de vídeo da rede pode ser completamente sequestrado por ele e assim, os demais usuários irão assistir um vídeo falso. Um atacante que tenta realizar um ataque de eclipse, geralmente apresenta um alto grau de entrada e saída de dados, se comparado com os valores médios encontrados no sistema [40, 80]

2. **Ataques de consumo de recursos**: nesse ataque, um participante malicioso requisita o máximo de dados possível de seus parceiros. O atacante não necessita desses dados, mas ele tenta exaurir os recursos dos demais participantes. Para tentar detectar este comportamento, os participantes do sistema podem relatar a quantidade de recurso que seus parceiros consomem a uma entidade centralizadora. Um participante que consome muitos recursos de vários outros pode estar realizando um ataque desse tipo, pois, sua intenção é consumir recursos de seus parceiros e fazer com que eles não consigam compartilhar mais.

3. **Pote de mel**: um atacante pode atrair vários parceiros com promessas de servi-los. Esses atacantes geralmente anunciam-se com muitos recursos e com uma grande quantidade de dados disponível. Assim que um parceiro requisita algum serviço, o atacante nega a requisição realizada. Tal comportamento gera atrasos e rupturas no fluxo de vídeo dos participantes que foram atraídos pelo pote de mel. Para tentar evitar que os participantes sejam atraídos por potes de mel, os participantes podem relatar entre suas parcerias a quantidade de dados

servida e negada pelos demais parceiros. Dessa forma, eles podem tentar detectar esse tipo de comportamento malicioso.

Além dos ataques propriamente ditos, uma série de comportamentos não desejados são observados em sistemas P2P de transmissão de vídeo ao vivo. Estes padrões de comportamentos geralmente não visam atacar o sistema. Porém, eles contribuem para a queda de qualidade do serviço provido. Entre estes comportamentos, se destacam:

1. **Alta Dinâmica dos Participantes (alto *churn*)**: as entradas e abandonos frequentes de participantes do sistema P2P de transmissão ao vivo podem levar a rupturas na estrutura da rede e a uma baixa qualidade do fluxo de vídeo. E mesmo em um sistema não estruturado, como os sistemas P2P baseados em malha, a alta dinâmica pode levar a constantes perdas de parcerias. Dessa forma, a realização de parcerias com participantes propensos a um pequeno tempo de vida no sistema pode não ser vantajoso. Por esse motivo, os participantes podem comunicar-se entre si relatando sobre a entrada e o abandono de suas parcerias. Desta forma, eles (os participantes) podem tentar computar o tempo médio de permanência no sistema de um determinado participante, e então, decidir se é vantajoso ou não fazer a parceria com ele.

2. **Egoísmo (*free riding*)**: um participante do sistema pode decidir a não compartilhar os dados da mídia que é transmitida. Quando um participante não se comporta de maneira altruísta, ou seja, ele não compartilha os dados, o sistema pode sofrer com a escalabilidade no número de usuários e a qualidade percebida do serviço pode cair (por falta de trechos de vídeo, por exemplo). Maneiras para evitar egoísmo, como o *"tit-for-tat"* adotado no Bittorrent, podem não ser eficientes em sistemas de transmissão ao vivo uma vez que os participantes em um sistema ao vivo procuram por uma faixa muito restrita de dados. Mesmo assim, para evitar parceiros egoístas, os participantes devem monitorar seus parceiros e trocar informações com os demais participantes do sistema sobre o comportamento de suas parcerias.

Os ataques aos sistemas de transmissão ao vivo em P2P podem ser realizados tanto sobre os dados transmitidos quanto às informações de controle do protocolo do sistema. Em ataques realizados sobre as mensagens de controle, os atacantes podem alterar informações importantes sobre os participantes do sistema, inclusive sobre si mesmos. Ao alterar essas informações, eles podem repassar uma informação falsa

sobre si, atraindo mais vítimas ou até mesmo podem gerar problemas aos demais participantes do sistema por conta de informações falsas divulgadas.

Quanto aos ataques realizados sobre os dados da mídia, os atacantes podem ignorar pedidos, corromper os dados, atrasar a entrega, duplicar o envio ou mesmo poluir os dados que compõem o fluxo de vídeo. As consequências são as mais diversas como: queda na qualidade do fluxo ao vivo, diminuição do número de participantes do sistema que conseguem o fluxo de vídeo correto, e recepção de informações falsas.

Alguns ataques são específicos para a topologia de rede P2P utilizada. Por exemplo, um participante pode alterar informações relativas ao seu tempo de retransmissão de dados em uma estrutura baseada em árvore ou mesmo se conectar a muitos outros participantes em uma estrutura baseada em malha. Quando um nodo altera informações sobre seu tempo de retransmissão em topologias baseadas em árvore, ou mente sobre sua capacidade de processamento e banda de rede, ele pode atrair muitos outros usuários a fazer uma conexão ou parceria com ele. Desse ponto em diante, o atacante passa a agir de forma maliciosa, negando pedidos, consumindo recursos dos vizinhos (por exemplo, as conexões existentes) ou mesmo ocultando os seus parceiros da parte íntegra da rede (eclipse da rede). Alguns sistemas apresentam uma estrutura vulnerável a este tipo de comportamento, como é o caso do CollStreaming [99], é organizado em uma estrutura de múltiplas árvores.

Os ataques de negação de serviço são facilmente ilustrados. Por exemplo, um participante malicioso em uma estrutura de árvore pode ignorar pacotes de dados que ele deve encaminhar, causando uma interrupção no fluxo de vídeo para todos os participantes abaixo dele na hierarquia da árvore. Mais ainda, eles podem alterar ou forjar dados e desta forma, seus vizinhos ou filhos na estrutura de árvore terão um fluxo de vídeo com conteúdo indesejado (poluído). Além disso, os atacantes podem simplesmente repassar o dado com um atraso propositalmente grande. Isto torna a qualidade do fluxo de vídeo baixa, com uma latência alta. Muitas vezes, os dados do fluxo de vídeo se tornam inúteis por chegarem fora de tempo esperado. Finalmente, os atacantes podem inundar a rede com dados falsos ou duplicados, causando uma sobrecarga nos recursos necessários para o bom funcionamento da aplicação.

Na maior parte dos ataques descritos, os atacantes podem combinar diversas técnicas para atingir seus objetivos. Podem combinar ataques à estrutura e controle da rede P2P, e também, aos dados transmitidos. Além disso, eles podem se organizar e gerar ataques sincronizados, tornando a detecção e o combate mais desafiadores.

2.3.2 Ataques de Poluição

Os ataques de poluição, ou simplesmente poluição de dados, são os ataques onde os participantes maliciosos alteram, forjam ou simplesmente inutilizam os dados que são transmitidos pela rede P2P. Esses ataques são comuns em sistemas de compartilhamento de arquivos P2P, motivados principalmente pela tentativa de diminuir a proporção de arquivos verdadeiros [10]. Isso torna a obtenção desses arquivos mais difícil e, por consequência, faz com que os usuários fiquem insatisfeitos com os serviços da aplicação de compartilhamento.

Alguns autores diferenciam dois tipos de comportamento malicioso que alteram o conteúdo compartilhado. No primeiro deles, há uma intenção de alterar o conteúdo e espalhar cópias falsas do mesmo. Esse comportamento é denominado como envenenamento (*poisoning*) [10]. O segundo comportamento é um processo aleatório, onde o conteúdo é alterado de forma não intencional. O segundo comportamento é denominado como poluição por esses mesmo autores. Há uma crença que a poluição possa ser encontrada em quaisquer arquivos compartilhados da rede P2P, enquanto o envenenamento é direcionado em certos arquivos. Apesar dessa diferenciação, os trabalhos atuais não as levam em consideração de forma acentuada e tratam todo comportamento que altera ou forja dados como um ataque de poluição.

A prevalência desse comportamento malicioso em sistemas de compartilhamento de arquivos P2P está intimamente ligada aos problemas de infração às leis de direitos autorais [10, 48]. Medições como as apresentadas em [48] mostram que em redes de compartilhamento como a Kazaa [43], as músicas populares mais novas chegam a ter 50% de suas versões poluídas.

Esses ataques são especialmente desafiadores em transmissão ao vivo em P2P. Isto porque, os mecanismos utilizados para lidar com conteúdo poluído em compartilhamento de arquivos P2P não podem ser diretamente aplicados no contexto de transmissão ao vivo [7]. No compartilhamento de arquivos, os arquivos poluídos podem ser apagados ou retirados do sistema pelos seus usuários enquanto em uma transmissão ao vivo, a detecção e o tratamento dos dados poluídos dificilmente não impactam na visualização da mídia devido às fortes restrições de tempo [101].

Mais ainda, as maneiras tradicionais para lidar com conteúdo poluído como lista negra [49], assinatura baseadas em hash [96] e criptografia dos dados [64] podem aumentar a sobrecarga de comunicação e atrasar a execução da mídia transmitida [101]. Esses efeitos são negativos em uma aplicação com fortes restrições de tempo e, onde seus usuários podem ter restrições de recursos como banda de rede.

Além do impacto na interrupção da execução da mídia e desperdício de banda

de rede com conteúdo indesejado, há uma série de outros impactos negativos ao sistema atacado por poluição de conteúdo. Por exemplo, o fornecedor do serviço de transmissão ao vivo terá uma má reputação quanto à qualidade de seus serviços. Os usuários se tornam insatisfeitos e abandonam o sistema, deixando de gerar recursos (e.g. pagamento pelo serviço).

Há uma série de facilitadores para ataques de poluição nos sistemas de transmissão ao vivo. Por exemplo, a maior parte das aplicações de transmissão ao vivo em P2P usam dois tipos de algoritmos de seleção de dados a serem requisitados. Esses algoritmos de fato buscam por dois tipos de trechos de mídia: o trecho com a marcação de tempo (*timestamp*) mais antiga e o trecho com a marcação mais recente. Os trechos com marcação mais antiga se encontram próximos ao seu fim de validade, na eminência de serem executados. Os trechos com marcação mais recentes acabaram de ser criados e por consequência, são mais raros. Com conhecimento desses protocolos, os poluidores geram informações falsas, despertando interesse dos participantes na rede P2P. Isso faz com que os ataques sejam mais rápidos e com grande alcance. Além disso, as aplicações mais populares não usam criptografia nas suas mensagens de protocolo [20].

Abaixo são apresentadas uma série de fraquezas das soluções tradicionais para tratar o problema de ataques em sistemas de transmissão ao vivo em P2P:

- A lista negra [49] não é eficiente em transmissões ao vivo (conforme será mostrado no capítulo 6). Não há um conceito explícito de identificação nas aplicações existentes. Os participantes são reconhecidos apenas por seus endereços IP, que pode ser facilmente falsificado. Caso um conceito de identificador global seja implementado, os poluidores podem simplesmente criar novas identidades e assim, não serem identificados facilmente.

- Uma infra-estrutura com chaves públicas usa um mecanismo de criptografia para evitar a modificação do fluxo [64]. Esse mecanismo não é adequado ao ambiente de transmissão ao vivo em P2P. Primeiro, porque é difícil estabelecer parceiros confiáveis em um ambiente P2P [64]. Segundo, a construção de tal infraestrutura e da criptografia em si é incompatível com os requisitos de simplicidade do sistema ou demasiadamente custosa para os sistemas de transmissão ao vivo em P2P existentes. Finalmente, mesmo que um dado seja verificado ao ser capturado da rede e seja descartado em caso de poluição, o custo de transmissão com dados indesejados continua existindo e, a sobrecarga da rede continua alta [7].

- Uma medida de defesa é a utilização de um esquema de assinatura baseada em *hash*. Nesse esquema de assinatura, o valor *hash* de um *chunk* é computado

e espalhado pela rede. Os participantes do sistema checam esse valor antes de retransmitir ou utilizar o *chunk*. Essa medida permite que os participantes não retransmitam conteúdo poluído ingenuamente e que se tornem poluidores passivos. Porém, dependendo do número de poluidores existentes na rede e como eles se organizam, a sobrecarga imposta por pedidos de retransmissão pode novamente tornar-se custosa [6, 7].

Além do comportamento malicioso com ataques de poluição, pode existir uma série de outros comportamentos oportunistas e indesejados dos participantes do sistema de transmissão ao vivo em P2P. Nas seções que seguem, são apresentados alguns desses comportamentos, assim como o modelo adotado desses comportamentos nas simulações realizadas nesse trabalho.

2.3.3 Comportamentos Oportunistas

Enquanto muitos usuários no sistema de transmissão ao vivo em P2P se dispõem a compartilhar dados com os demais participantes, pode existir uma parcela de usuários que não contribuem com o sistema. Esses usuários são denominados *"free-riders"* [9]. Esses usuários não irão contribuir com o sistema a menos que medidas de incentivo ou penalizações sejam tomadas [60]. No contexto de aplicações P2P, até os usuários que não contribuem acima de um nível aceitável são considerados *"free-riders"*.

A medida mais comum para classificar os participantes como *"free-riders"* ou não é a taxa de compartilhamento do participante. Essa medida é definida como a relação entre o total de dados servidos pelo participante e o quanto de dados ele captura da rede P2P. Alguns sistemas P2P, como o Bittorrent [5], adotam estratégias que forçam o compartilhamento e, dependendo da taxa de compartilhamento, os participantes conseguem melhores parcerias ou até mesmo são banidos. Dessa forma, os participantes acabam compartilhando dados enquanto eles capturam os dados da rede. Caso contrário, eles não conseguirão completar a sua tarefa.

Abaixo são relacionados algumas medidas que podem coibir a existência de *"free-riders"* em um sistema P2P. Cada medida é descrita brevemente e são apresentados os argumentos prós e contras essa medida.

(A) Abordagem com base Monetária:

Uma abordagem com base monetária tarifa os participantes pelos serviços que eles recebem [41]. Um mecanismo armazena informações sobre os participantes para uma futura tarifação. Como incentivo para o compartilhamento de dados, os participantes recebem descontos em suas transações de acordo com a sua taxa de compartilhamento.

Apesar do apelo financeiro ser um forte argumento para que os participantes compartilhem dados, a manutenção de uma unidade ou módulo para gerar a tarifação de forma justa pode ser custosa para o sistema. Além disso, esses pontos de tarifação geralmente são unidades centralizadoras, o que pode levar o sistema a problemas com ataques ou contenção de recursos.

(B) Abordagem com base em Reciprocidade:
Uma abordagem baseada em reciprocidade tem como principal idéia o compartilhamento entre parceiros que são mutuamente altruístas. Em outras palavras, cada participante monitora o comportamento de seus parceiros e avalia o seu nível de contribuição. Caso esse nível seja acima de um limite pré determinado, o participante realiza interações com esse parceiro (enviando e pedindo dados). Caso contrário, o participante evita esse parceiro, não respondendo suas requisições [41].

Essa abordagem geralmente não mantém um histórico de longo prazo. Ou seja, as decisões tomadas e os monitoramentos feitos levam em consideração apenas a seção corrente do participante. Dessa forma, um usuário anteriormente julgado como um *"free-rider"* pode ser visto como um parceiro justo em sessões futuras. Uma das grandes vantagens dessa abordagem é que ela preserva o anonimato dos participantes, uma vez que a identificação entre sessões dos participantes não é crucial para o funcionamento desse mecanismo.

(C) Abordagem com base em Reputação:
Na abordagem baseada em reputação, o sistema constrói e mantém um repositório com informações sobre os participantes. Assim, os participantes que são bem reputados recebem melhores serviços por parte do sistema P2P [41]. Geralmente, a reputação de um participante é construída a partir de relatos de seus parceiros com informações sobre as interações com o participante em questão. Por exemplo, um participante que foi benevolente e atendeu bem seus parceiros terá uma boa reputação, pois, eles irão fazer bons relatos. Em contra partida, participantes pouco altruístas (e.g. *"free-riders"*) terão relatos ruins de seus parceiros, o que os tornará participantes com baixa reputação.

Os sistemas de reputação geralmente armazenam informações a longo prazo. Assim, as reputações dos participantes extrapolam a sessão atual e, por consequência, marcam um usuário com seu comportamento passado, mesmo que ele resolva mudar de comportamento. Em outras palavras, uma vez reputado como um mau parceiro, a conversão para um bom parceiro é difícil de ser computada pelo sistema.

Outros Comportamentos Indesejados

Um ataque qualquer a um sistema P2P pode ser combinado com outros tipos de comportamentos indesejados. Por exemplo, os atacantes podem combinar entre si para gerar um ataque coordenado ao sistema. Eles podem articular para causar um maior dano ao sistema de transmissão ao vivo P2P durante um ataque. Assim, eles realizam um conluio, que é um acordo entre os participantes maliciosos para trapacear o sistema P2P, alcançando um ataque com maior impacto [7, 31]. Ataques de poluição combinados com conluio dos nodos maliciosos podem causar grandes danos aos sistemas de transmissão ao vivo em P2P [7].

Em muitos sistemas P2P, a identificação de um participante é dada no momento que ele se junta à rede sobreposta. A identificação do mundo real do participante não é amarrada a sua identificação na rede. Esse mecanismo facilita o processo de identificação de um participante em uma sessão e permite que as redes cresçam rapidamente, uma vez que os novatos podem facilmente se juntar ao sistema [21]. Entretanto, participantes maliciosos podem aproveitar disso para repetidamente alterar suas identidades e assim, ter as vantagens de ser um novo participante do sistema. Essa lavagem de identidade é conhecida como *"whitewashing"* e de fato é utilizada por atacantes para escapar de mecanismos de proteção do sistema atacado.

Algumas técnicas tentam combater a lavagem de identidade. Por exemplo, pode-se impor um alto custo no processo de se obter uma nova identidade para todos os novatos do sistema. Esse alto custo associado à obtenção de novas identidades pode desestimular a constante mudança de identificação que os nodos maliciosos fazem. Uma outra maneira de combater a lavagem de identidade é forçar a criação de uma identidade insubstituível. Essa identidade única seria obtida, por exemplo, a partir de uma autoridade certificadora centralizada [41]

2.3.4 Ataques e Poluição de Dados aos Sistemas de Transmissão ao Vivo em P2P

Na literatura existe uma série de trabalhos que tratam principalmente os comportamentos oportunistas dos usuários de um sistema de transmissão ao vivo em P2P. Trabalhos como [13, 40, 85] apresentam propostas para reputar os participantes do sistema, banindo os participantes que contribuem pouco. Neste contexto, um sistema de reputação consiste em um sistema capaz de avaliar a atuação de um participante na rede P2P e, através de seu comportamento passado, atribuir-lhe uma nota. Por exemplo, os participantes com baixa relação *upload/download* nas suas

interações passadas serão avaliados de forma punitiva. Por consequência, terão notas ruins de reputação e assim, eles terão uma menor prioridade no sistema em relação aos participantes que possuem maior relação *upload/download*. Dessa forma, esses trabalhos apresentam propostas que podem incentivar um comportamento altruísta dos participantes da rede P2P de transmissão ao vivo.

Com relação ao conteúdo poluído, há uma série de trabalhos desenvolvidos no contexto de compartilhamento de arquivos em P2P. Sistemas como Credence [91] apresentam uma abordagem distribuída, na qual os participantes da rede assinam reputação aos *objetos* descarregados do sistema (arquivos baixados). Além de uma nota que descreve se um usuário considerou um objeto como poluído ou não, há uma reputação relativa aos usuários participantes do sistema. Por exemplo, a proposta de reputação em sistema de compartilhamento de arquivos denominada Scruber [15], consegue identificar por meio de reputação os participantes maliciosos que ativamente disseminam conteúdo poluído. Um efeito colateral da adoção de tais medidas é uma diminuição da propagação de conteúdo poluído de forma ingênua. Caso algum participante, mesmo que não seja poluidor, dissemine conteúdo poluído, ele terá sua reputação penalizada. Assim, os participantes honestos do sistema são incentivados a verificar o conteúdo que eles têm e descartar o quanto antes os arquivos poluídos.

Entre os primeiros trabalhos que tratam poluição de conteúdo em sistemas de transmissão ao vivo em P2P estão [28–31]. Nesses trabalhos é apresentada uma proposta de um sistema de transmissão ao vivo em P2P resistente a alteração de conteúdo e intrusão de participantes maliciosos. Também são apresentadas comparações entre quatro alternativas para verificar a integridade do vídeo distribuído no sistema de mídia contínua ao vivo em P2P. Nesse sentido, os autores mostram que é possível realizar a verificação do conteúdo recebido da rede e assim, evitar a execução de trechos de mídia poluída.

Em [20] é apresentado um experimento no qual um poluidor ativo é colocado em um sistema real. Os resultados obtidos indicam que ataques de poluição podem danificar por completo um sistema ao vivo de vídeo em P2P. Por exemplo, em um dos experimentos, o sistema de transmissão ao vivo teve uma forte evasão dos participantes que estavam insatisfeitos com o serviço oferecido. O número de participantes caiu para cerca de 10% da quantidade de participantes existentes antes ao ataque (de 1000 para cerca de 100 participantes). Nesse mesmo trabalho são sugeridas algumas abordagens para evitar o consumo de dados corrompidos. Os autores mostram que é possível verificar a integridade dos dados da mídia transmitida com um baixo custo adicional. Utilizando os esquemas propostos por eles há um custo de processamento em $O(n)$ para assinar os blocos de dados a serem transmitidos, onde n é a quantidade de *chunks*

contida em cada bloco. A verificação a ser realizada pelo participante é de $O(1)$. O custo adicional de banda de rede é cerca de 5% da banda de rede necessária para o envio do fluxo original.

Apesar das propostas existentes para evitar conteúdo poluído em um sistema de transmissão ao vivo em P2P, não são apresentadas evidências que o assinamento dos dados transmitidos seja uma medida eficiente para o combate aos poluidores. Além disso, o experimento realizado para verificar a abrangência de um ataque aos sistemas de transmissão ao vivo em P2P não apresenta uma visão generalizada da aplicação. São mostrados apenas alguns dados pontuais, de alguns poucos participantes.

2.3.5 Verificação do Conteúdo Distribuído

Nesta seção são apresentadas algumas soluções propostas pela literatura para realizar a verificação do conteúdo distribuído ao vivo em arquitetura P2P. As soluções propostas têm como principal objetivo possibilitar que um participante do sistema tome conhecimento se o dado que ele acaba de obter é verdadeiro ou falso (poluído). Assim, não há uma distinção entre a entrega de um dado intencionalmente alterado ou uma porção de dados corrompida durante a transmissão. Em ambos os casos há um prejuízo claro para o sistema e, portanto, ambos podem ser tratados como poluição.

Vários protocolos para verificação de dados foram propostos para distribuição de conteúdo em *multicast*. O esquema originalmente utilizado para comunicação ponto a ponto (que adiciona um código às mensagens), não são bem aplicáveis ao ambiente *multicast*. Isso porque, o código adicionado às mensagens são computados utilizando uma chave compartilhada e, se todos compartilham a mesma chave em um ambiente *multicast*, qualquer participante pode forjar dados. Por outro lado, uma abordagem ingênua de assinar cada pacote de dados usando uma criptografia assimétrica induz uma alta sobrecarga ao sistema [28, 30, 31], tornando essa abordagem impraticável.

A seguir serão abordadas algumas maneiras de se verificar a integridade dos dados distribuídos pelo sistema de transmissão ao vivo em P2P. Em todos os casos abaixo, os participantes do sistema devem realizar uma verificação dos dados antes de consumi-los e compartilhá-los. Caso eles não façam a verificação, eles correm risco de repassar conteúdo poluído de forma ingênua. Assim, esse participante se tornará um poluidor passivo no sistema e deverá ser tratado da mesma forma que um poluidor. Dessa forma, a adoção de medidas punitivas pode incentivar que todos os participantes do sistema realizem o processo de verificação de dados antes de compartilhá-los.

(A) Verificação Hash:

Assinar um pacote consiste no processo de computar um valor *"hash"* sobre seu conteúdo utilizando uma função hash. Após esse processo, o valor calculado é finalmente assinado com a chave privada do servidor da transmissão ao vivo. Quando algum participante recebe um pacote de dados, ele é capaz de recomputar o valor hash sobre o conteúdo recebido e comparar com o valor assinado. Esse processo é seguro, pois, o valor assinado só pode ser decifrado com a utilização da chave do servidor de transmissão ao vivo.

Esse esquema (assinar os dados) é utilizado em vários sistemas como o BitTorrent [20]. Um participante, antes de tentar obter os dados de um arquivo compartilhado, obtém o arquivo do tipo *"torrent"*. Esse arquivo possui os valores hash de todos os *chunks* de dados do arquivo a ser recuperado da rede P2P. Quando o participante recebe qualquer *chunk* de seus parceiros, ele verifica se o *hash* calculado confere com o valor previamente computado. Caso positivo, o dado está íntegro, caso contrário, o dado é descartado e o processo de busca por esse dado recomeça.

A abordagem mais simples para a verificação de *chunks* em um sistema de transmissão ao vivo em P2P é fazer um assinamento de cada *chunk* antes de começar a sua difusão pela rede [20]. A partir desse momento, os participantes da rede pegam o valor *"hash"* de cada *chunk* transmitido para posterior verificação.

Essa abordagem é rápida em termos de execução, pois, não há a necessidade de agrupamento de vários *chunks* (o que iria atrasar a execução). Porém, ela introduz um custo computacional alto, pois, para cada *chunk* gerado, o servidor terá que computar um valor hash e para cada *chunk* recebido por um participante do sistema, ele terá que calcular um valor hash, decifrar o valor que a fonte enviou e realizar a comparação.

Uma maneira de reduzir a carga de trabalho da fonte seria utilizar a própria rede P2P para distribuir os valores de hash que são computados para cada *chunk*. Porém, essa abordagem permite que um participante malicioso altere o dado que é transmitido pela fonte e gere novos valores hash para esse dado forjado. Quando os demais participantes recebem o dado forjado, ele pode fazer a sua verificação com o hash também forjado e isso o levará à crença de ter obtido um dado íntegro.

Em canais com uma alta taxa de codificação do vídeo, a taxa de *chunks* é muito alta. Isso significa que o número de operações por segundo é igualmente alta, o que torna o sistema inviável para dispositivos com baixa capacidade de processamento. Neste caso, dispositivos portáveis como telefones celulares podem não ter capacidade para participar da rede de transmissão ao vivo em P2P.

(B) *Linear digests*:

Para evitar que cada *chunk* seja verificado, pode-se criar grupos de n *chunks* e esse

grupo é assinado pela fonte, criando uma mensagem especial sobre o assinamento desse grupo. A mensagem com as assinaturas devem ser enviadas aos participantes antes da disseminação do conteúdo que ela representa. Isso pode implicar em um atraso na disseminação do conteúdo ao vivo, pois, a fonte terá que gastar um certo período de tempo enquanto agrupa os *chunk*. Uma das principais vantagens é que essa abordagem resulta em uma sobrecarga menor da rede.

(C) Autenticação baseada em grafo:
Em uma autenticação baseada em grafos, o servidor assina apenas um *chunk* e os consecutivos são ligados ao assinado através de cadeias de valores *hash*. Caso fosse utilizada apenas uma cadeia de verificação, o esquema de verificação estaria sujeito a erros devido a perdas de pacotes. Por isso, invés de se criar um caminho único de verificação, é criado um grafo. Nesse grafo os *chunks* são representados por vértices e uma aresta direcionada entre dois pontos no grafo, P_i e P_j, representa que P_j contém o valor hash do *chunk* P_i. Um *chunk* correspondente a um nodo pode ser verificado se existe algum caminho verificado anteriormente entre a fonte e o nodo no grafo.

(D) Encadeamento Merkle-Tree:
Nessa abordagem, é construída uma árvore de autenticação a partir da fonte de cada bloco. As folhas correspondem aos valores de hash de cada *chunk* no bloco. Outros nodos são construídos como hashes de seus filhos. A assinatura na raiz se torna a assinatura do bloco. A informação de autenticação de cada *chunk* é: a assinatura do bloco, a posição do *chunk* no bloco e os "irmãos" de cada nodo no caminho que vai da folha correspondente ao *chunk* até a raiz da árvore de autenticação. Quando um participante recebe um *chunk* no bloco e sua informação de autenticação correspondente, ele calcula o hash da raiz e então verifica a assinatura do bloco com o valor calculado.

Para um bloco com n *chunks*, a fonte necessita realizar $2n + 1$ operações de hash e uma operação de assinatura. O participante que recupera as informações deve realizar $2n+1$ operações de hash e uma verificação de autenticidade do bloco. O atraso imposto na rede é equivalente ao processamento de n *chunks* no servidor e o processo nos participantes tem custo de 1 *chunk*.

2.3.6 Lista Negra Centralizada

Na literatura, há a recomendação de realizar uma verificação da integridade dos dados para combater a disseminação do conteúdo poluído. Junto a isso, também é

recomendado o uso de estruturas como uma lista negra centralizada [20,28,30,31]. Em uma abordagem de lista negra, o identificador de um participante considerado malicioso é colocado em uma lista. Essa lista pode ser acessada pelos demais membros do sistema P2P a qualquer momento. Quando um participante recebe um convite de uma nova parceria, ele pode verificar se o parceiro candidato se encontra na lista negra e, caso positivo, ele pode evitar o contato com um possível poluidor. O participante também pode acessar a lista negra periodicamente para verificar se seus atuais parceiros foram caracterizados como participantes maliciosos.

No sistema de lista negra deste trabalho, os participantes monitoram o comportamento passado de seus parceiros e avaliam o desempenho deles [40, 92, 100]. Periodicamente os participantes reportam o que foi observado a uma entidade centralizada que resume a reputação das observações recebidas.

Uma ação comum da entidade centralizadora é ponderar as reputações recebidas para evitar distorções nas reputações recebidas. Por exemplo, em [40], quando o servidor recebe uma reputação sobre um determinado participante, ele ponderada essa reputação com relação à média das demais reputações recebidas anteriormente. Assim, reputações com valores próximos aos valores médios terão mais importância (maiores pesos) que reputações que diferem da média.

Em outros casos, como em [92,100], a reputação é ponderada pela credibilidade do participante que a enviou. Ou seja, quem tem maior credibilidade no sistema terá um peso maior em suas opiniões. Nesse caso, a credibilidade de um participante pode ser a sua própria reputação. Dessa forma, quanto maior a reputação de um participante, maior o peso de sua opinião sobre seus parceiros.

Mais detalhadamente, a informação sobre a reputação é um valor entre 0 e 1. Os participantes que oferecem dados corretos aos seus parceiros obtêm incrementos em suas notas de reputação. Caso os dados oferecidos tenham conteúdo inválido, as notas sofrerão perdas. Dessa forma, os participantes do sistema escolhem suas parcerias a partir do valor da nota de reputação do candidato a parceiro. Assim, uma nota em um servidor pode determinar quais são os participantes que originam o conteúdo poluído.

No sistema proposto nesta seção, um participante p_i envia os dados relativos à reputação sobre cada um de seus parceiros ao servidor de lista negra LN. O servidor LN pode ser o próprio servidor do conteúdo ao vivo, mas por questões de desempenho, segurança e anonimato da fonte geradora de conteúdo, LN geralmente é uma entidade confiável à parte do sistema P2P. A interação entre p_i e LN é realizada a cada intervalo de tempo Tr_i. Nesse intervalo, além de enviar as notas de seus parceiros, p_i também consulta LN para atualizar os valores da sua lista de reputações local R_i.

Assim, p_i reporta o comportamento observado de p_j ($p_j \in LP_i$) enviando uma

nota ao LN e, ao consultar LN sobre o parceiro p_j, ele obtém a nota consensual do sistema sobre p_j. O participante p_i decide realizar uma nova parceria ou novas trocas de *chunks* com um nodo p_j se a reputação de p_j for maior que um limite mínimo aceitável estabelecido pelo nodo p_i ($limite_i \leq R_i[p_j]$).

A reputação do parceiro p_j é assinada por p_i conforme a equação 2.1. A cada intervalo Tr_i, o participante p_i verifica a sua interação com p_j. Neste intervalo, p_i requisita r *chunks* à p_j ($r > 0$). Entre os dados requisitados, n *chunks* são considerados poluídos por p_i ($0 \leq n \leq r$). A relação n/r determina se um fluxo enviado de p_j para p_i teve um grau de poluição aceitável ($n/r \geq limiteNR_i$) ou não ($n/r < limiteNR_i$), onde $limiteNR_i$ é um valor de reputação válido para os parceiros de p_i. Se o grau de poluição for aceitável, p_i incrementará a reputação $R_i[p_j]$. Caso contrário, p_i decrementará a reputação local de p_j.

$$R_i[p_j] = \begin{cases} max(0, R_i[p_j] - \alpha_{p_i} * (1 + n/r)^{y_i}) & \text{se } n/r > \text{valor } limiteNR_i \\ min(1, R_i[p_j] + \alpha_{g_i} * (1 - n/r) & \text{caso contrário} \end{cases} \tag{2.1}$$

A penalidade ou a recompensa dada à reputação $R_i[p_j]$ são ajustadas pelos fatores α_{p_i} e α_{g_i} respectivamente, onde ($\alpha_{p_i} \geq \alpha_{g_i}$). Mais ainda, a reputação $R_i[p_j]$ sofre uma perda proporcional à relação $(1 + n/r)$ elevado a um fator y_i; o que pode levar a perdas exponenciais de reputação. O ganho de reputação segue à proporção $(1 - n/r)$ sem nenhum fator de ganho além de α_{g_i}. Com tal esquema de perdas e ganhos, um participante perde reputação de forma muito mais rápida que ele ganha. Dessa maneira, o sistema de lista negra centralizada pode identificar um poluidor muito rapidamente.

Finalmente, a nota reportada por um nodo p_i ao servidor de lista negra LN é ponderada pela sua própria reputação ($R_{LN}[p_i]$). Consequentemente, a reputação final de p_j é a média ponderada de todas as reputações já enviadas ao servidor de lista negra LN conforme a equação 2.2. Por esse esquema, participantes bem reputados no sistema P2P apresentam maior impacto na nota de um outro participante. Assim, os prováveis participantes maliciosos apresentam pouco impacto na reputação de um outro participante. Esse mesmo processo de ponderação é realizado em trabalhos de combate aos participantes egoístas de um sistema P2P [40, 92].

$$R_{LN}[p_j] = \frac{\sum_{p_k \in LPR_j} R_k[p_j] * R_{LN}[p_k]}{\sum_{p_k \in LPR_j} R_{LN}[p_k]} \tag{2.2}$$

A tabela 2.1 apresenta um resumo dos parâmetros utilizados para o modelo de lista negra centralizada adotada nesse trabalho.

Como expressado anteriormente, o participante p_i decide realizar ou continuar a parceria com p_j de acordo com o valor atualizado de $R_i[p_j]$. Caso $R_i[p_j] \geq Rmínimo_i$, o parceiro p_j é considerado confiável e p_i realiza a parceria ou continua suas interações com p_j. Caso $R_i[p_j] < Rmínimo_i$, p_j é considerado um poluidor e p_i não concretiza a nova parceria ou interrompe a já existente.

Tabela 2.1: Resumo dos elementos do modelo de lista negra centralizada.

Parâmetro	Descrição
p_i	participante i do sistema
LP_i	conjunto de parceiros de i
LP_i	conjunto de parceiros candidatos de i
LN	servidor de lista negra
Tr_i	intervalo de tempo entre interações de p_i e LN
R_i	conjunto de reputações que p_i tem de seus parceiros
$R_i[p_j]$	reputação de p_j visto em p_i
$R_{LN}[p_i]$	reputação de p_j visto no servidor
$limite_i$	limite da nota de reputação para tomar uma decisão
r	total de *chunks* requisitados em um intervalo
n	total de *chunks* poluídos em um intervalo
$limiteNR_i$	limite aceitável de dados poluídos (relação n/r)
α_{p_i}	Fator de penalização da reputação
α_{g_i}	Fator de premiação de reputação
y_i	Fator para acelerar exponencialmente a penalização

2.4 Resumo do Capítulo

Neste capítulo, foram apresentadas as principais estruturas para organizar uma rede P2P. Inicialmente, foram abordadas redes P2P de compartilhamento de recursos, suas principais arquiteturas e mecanismos de busca. Em seguida, foram abordados os conceitos de redes P2P para transmissão de mídia ao vivo. Foram detalhadas as principais estruturas utilizadas para esse fim, seus benefícios e suas limitações.

Verifica-se que os projetos existentes para transmissão de fluxo ao vivo em P2P são capazes de transmitir um conteúdo para uma grande população. Os custos envolvidos no servidor são baixos e, praticamente não há necessidade de estruturas específicas ou custosas. Entretanto, existem limitações às soluções existentes.

Os usuários de um serviço de P2P ao vivo na Internet experimentam problemas, normalmente, não encontrados nos serviços convencionais. Geralmente, existe uma latência grande e atrasos imprevisíveis na sessão de vídeo. Mais ainda, a diferença de tempo do ponto de visualização da mídia entre os usuários é grande. Alguns usuários assistem a transmissão ao vivo com minutos de atraso em relação ao restante da rede. Esta situação não é aceitável quando se assiste alguma transmissão com forte apelo ao vivo, por exemplo, um jogo de futebol ou as finais das olimpíadas.

Grande parte dos canais de vídeo disponível, neste tipo de aplicação, fornece um fluxo com vídeos de baixa resolução. Isto ocorre, em grande parte das vezes, devido às restrições de largura de banda nos usuários finais. Os fluxos providos, muitas vezes, são de baixa qualidade e instáveis; principalmente porque o número de usuários em muitos casos é relativamente baixo. Isto se torna um grande desafio, especialmente quando se deseja servir a cauda pesada de canais pouco populares [50].

Entre as duas abordagens principais utilizadas para estruturação dos sistemas de P2P de transmissão de fluxo ao vivo, a baseada em malha e orientada a pedidos de dados mostra-se atualmente a mais popular. Essa abordagem mostra um grande número de vantagens sobre sistemas baseado em árvores, principalmente sob o ponto de vista de dinâmica dos participantes do sistema. Enquanto uma abordagem em árvore sofre constantemente com as rupturas causadas por entrada e saída dos seus participantes, a abordagem baseada em malha lida com esta situação de maneira simples e flexível. Porém, a abordagem em malha pode se mostrar muito mais instável e com uma sobrecarga adicional maior devido ao grande número de mensagens de controle trocada entre seus usuários.

Novas abordagens para organizar os participantes da aplicação P2P estão sendo propostas. Geralmente, essas novas abordagens utilizam estruturas híbridas, com mistura baseada em árvores e malha. Trabalhos como [37,47,69] apresentam estruturas

mistas, com a intenção de serem robustas à dinâmica dos participantes e serem mais estáveis e previsíveis. Os resultados atuais, como o exposto em [46], mostram que a utilização do mecanismo de *"push-pull"* pode levar a uma redução da sobrecarga do tráfico da rede, menores latências e melhores taxas da mídia recebida.

Também foram apresentados alguns dos principais problemas com relação aos ataques em Sistemas de Transmissão ao Vivo em P2P. Foram formalizados os problemas tratados nessa tese, como poluição de dados e conluio dos participantes maliciosos. Além disso, foram apresentadas e discutidas soluções existentes para verificar a integridade dos dados que são transmitidos pela rede P2P em questão.

Com relação aos sistemas de transmissão ao vivo em P2P, há vários estudos realizados com o objetivo de encontrar maneiras de organizar e coordenar o funcionamento de tais aplicações [35, 37, 69, 89, 103, 103, 105]. Porém, não há uma atenção dispensada na mesma proporção aos ataques a esses sistemas. Mesmo que exista um série de trabalhos em sistemas semelhantes, como aplicações de compartilhamento de arquivos em P2P [14, 15, 19, 91], não há garantias de sucesso ao adotar as soluções de um sistema diretamente no outro.

Como apresentado, os ataques aos sistemas de transmissão ao vivo em P2P podem ser realizados tanto sobre os dados transmitidos quanto às informações de controle do protocolo do sistema. Em ataques realizados sobre as mensagens de controle, os atacantes podem alterar informações importantes sobre os participantes do sistema, inclusive sobre si mesmos. Ao alterar essas informações, eles podem repassar uma informação falsa sobre si, atraindo mais vítimas. Quanto aos ataques realizados sobre os dados da mídia, os atacantes podem ignorar pedidos, corromper os dados ou mesmo poluir os dados que compõem o fluxo de vídeo. Pode ocorrer queda na qualidade do fluxo ao vivo, diminuição do número de participantes do sistema que conseguem o fluxo de vídeo correto, e até mesmo recepção de informações falsas.

Finalmente, foram apresentadas propostas encontradas na literatura para realizar a verificação do conteúdo distribuído ao vivo em arquitetura P2P. As soluções propostas têm como principal objetivo possibilitar que um participante do sistema tome conhecimento se o dado que ele acaba de obter é verdadeiro ou falso (poluído). Essas soluções, na maior parte, são baseadas em marcação de pacotes com seus códigos *hash*. Com a adoção da técnica apropriada, consegue-se verificar a integridade dos dados transmitidos pela rede com uma sobrecarga relativamente baixa, tanto no processamento quanto na banda de rede consumida.

Capítulo 3

Sistema de Transmissão ao Vivo em P2P

Este capítulo apresenta a descrição e o funcionamento de uma aplicação de envio de mídia contínua ao vivo em P2P. A descrição apresentada utiliza como base as principais aplicações existentes atualmente. Por exemplo, Sopcast [81], o PPLive [33, 66] e o GridMedia [103, 108]. As aplicações seguem o modelo de rede P2P em malha com pedidos explícitos, apresentado na seção 2.2.2. A Figura 3.1 ilustra um exemplo do sistema de transmissão ao vivo em P2P que será detalhado nessa seção.

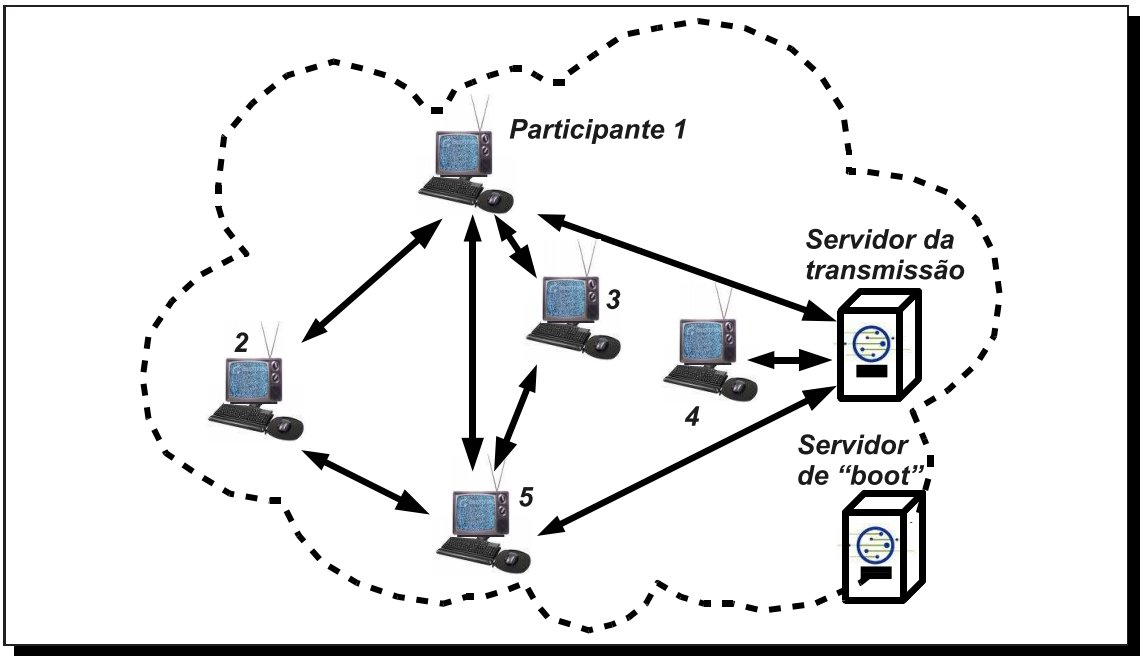


Figura 3.1: Modelo de sistema utilizado.

As principais entidades envolvidas neste sistema são as seguintes:

- Servidor da transmissão ao vivo: O servidor de transmissão é um participante especial do sistema P2P. Ele captura e codifica o vídeo que será transmitido pela rede. O servidor é a fonte inicial dos dados de vídeo da rede.
- Servidor de *boot*. O servidor de *boot* (ou *bootstrap*) é uma entidade centralizadora por meio do qual os demais participantes do sistema encontram seus parceiros iniciais para entrar na rede P2P. Todo participante se registra nesse servidor para fazer parte da lista dos participantes do sistema. Quando um novo participante se registra, o servidor de boot envia a ele uma lista com alguns parceiros candidatos. Quando um participante antigo deseja realizar mais parcerias, este pede ao servidor de *boot* uma lista com alguns participantes para tentar novos contatos.
- Participantes (clientes/nós/*peers*). São os usuários do sistema P2P de transmissão ao vivo. Cada participante está em contato com um subconjunto de todos os participantes do sistema. Não há hierarquia entre esses participantes, e qualquer um pode servir dados de vídeo e também realizar pedidos por dados aos seus parceiros.

O modelo analítico presente neste capítulo servirá de base para os demais experimentos e simulações realizados nesta tese. Além disso, o presente modelo também serve como base para criação de novos protocolos de envio de mídia contínua ao vivo em arquiteturas P2P.

3.1 Sistema P2P para Transmissão ao Vivo

Os sistemas de envio de mídia contínua ao vivo em P2P são sistemas compostos por participantes que colaboram entre si para a disseminação do conteúdo gerado por um servidor. Esses participantes se organizam em uma rede virtual, sobreposta à rede de computadores real. A organização dessa rede baseia-se, geralmente, em dois modelos. Um deles é uma estrutura como uma árvore, o outro é uma malha computacional.

Nesses sistemas de transmissão ao vivo há um par denominado servidor S. Ele gera todo o conteúdo que será disseminado pela rede. Assim, o objetivo dos demais participantes do sistema é receber a mídia que é gerada por S (e dividida em pedaços), e repassar para seus parceiros esses pedaços de mídia. Dessa forma, qualquer participante do sistema poderá enviar e receber dados para/de os demais participantes.

Mais detalhadamente, os sistemas par-a-par para envio de mídia contínua ao vivo usam recursos do conjunto $P = p_1, p_2, ..., p_n$ de seus participantes para repassar o conteúdo que é transmitido pelo servidor S. Cada participante p_i é livre para entrar e sair do sistema a qualquer momento. Tal comportamento diferencia a aplicação de transmissão ao vivo em P2P de aplicações baseadas em IP *multicast*, pois, em IP *multicast* a estrutura formada é pouco dinâmica.

Os sistemas mais populares de envio de mídia ao vivo em P2P utilizam uma rede sobreposta baseada em malha. Uma rede sobreposta é uma organização virtual dos participantes, que pode ser completamente diferente da organização física existente. Mais ainda, no modelo de malha, a rede não é estruturada de forma rígida e as parcerias no sistema são formadas aleatoriamente. As interações e trocas de dados entre os participantes p_i e p_j, com $i \neq j$, são normalmente orientadas pelos pedidos de dados e informações entre p_i e p_j. Assim, esse tipo de rede sobreposta (baseado em malha) é utilizada para aliviar os efeitos de entrada e saída dos participantes na rede [108]. Esta de abordagem é também utilizada para construir um sistema resistente a falhas e com visualização, sem interrupções da mídia contínua.

O funcionamento desse tipo de sistema acontece da seguinte forma: inicialmente, um participante p_i contata um mecanismo centralizado de inicialização. Esse mecanismo de inicialização normalmente está à parte da rede P2P. Esse mecanismo é denominado bootstrap B ou rastreador. Na inicialização de p_i, B envia um subconjunto dos nodos participantes do sistema para o participante p_i. O subconjunto é a lista inicial de parceiros candidatos do participante p_i e é denominada LPC_i ($LPC_i \subseteq P$ e $LPC_i \neq \emptyset$). Além de se registrar e pegar uma lista de candidatos a parceiros, o novo participante sincroniza a posição atual da mídia ao vivo, com a posição informada em B [26]. Assim, p_i tem uma referência do ponto da mídia ao vivo que está sendo gerada pelo servidor S e saberá a partir de qual ponto deverá solicitar os dados e assistir a mídia ao vivo.

O novo participante p_i seleciona, aleatoriamente, uma quantidade n de participantes de LPC_i como parceiros candidatos. Eles formarão o conjunto de parceiros de p_i, denominado LP_i ($LP_i \subseteq LPC_i$). Os conjuntos LPC_i e LP_i são dinâmicos, pois, cada nodo p_j está livre para abandonar o sistema de mídia contínua ao vivo. Quando $p_j \in LP_i$ e o nodo p_i detecta a inatividade deste parceiro, p_i remove p_j de LPC_i e LP_i e seleciona um novo elemento de LPC_i para criar uma nova parceria.

O participante p_i sempre tenta manter sua LPC_i com um número de candidatos acima de um limite $limite_i$. O valor de $limite_i$ pode ser dado pela capacidade de cada nodo, como banda de rede ou número de conexões disponível. Assim, quando $|LPC_i| < limite_i$, p_i recorre à B para obter novos elementos para LPC_i.

Cada participante p_i também contém um mapa de *chunks* de tamanho m, denominado cm_i. Esse mapa sinaliza os *chunks* da mídia que ele contém ou necessita. Ou seja, o mapa cm_i representa um trecho contínuo da mídia transmitida ao vivo pelo sistema P2P que será reproduzida pelo participante p_i.

Inicialmente, cada posição do mapa é marcada como *"desejada"*, ou seja, $cm_i[x] = desejada$, onde $x = [0..m]$. Periodicamente, p_i requisita cm_j a cada um de seus parceiros p_j ($p_j \in LP_i$). Dessa forma, p_i verifica quais parceiros podem satisfazer a sua necessidade por determinado *chunk* c_t. Quando p_i recebe um *chunk* x qualquer, p_i faz $cm_i[x] = disponivel$.

Periodicamente, p_i verifica quais *chunks* c_t ele necessita($cm_i[c_t] = desejada$) e verifica entre seus parceiros p_j ($p_j \in LP_i$), quais possuem o *chunk* c_t ($cm_j[c_t] = disponivel$). O parceiro p_j que contém o *chunk* c_t e que possui maior disponibilidade de recursos é escolhido por p_i para a realização do pedido de *chunk*. Quando p_i recebe o *chunk* c_t de p_j, p_i faz $cm_i[c_t] = disponivel$. Os processos de escolha do *chunk* a ser requisitado e a escolha do parceiro serão detalhados nas seções seguintes.

A tabela 3.1 apresenta um resumo dos parâmetros utilizados para descrever um sistema P2P de transmissão ao vivo.

Tabela 3.1: Resumo dos parâmetros de um sistema P2P de transmissão ao vivo.

Parâmetro	Descrição
S	servidor de mídia contínua
$P = p_1, p_2, ..., p_n$	conjunto dos participantes do sistema
B	*bootstrap* ou rastreador do sistema
p_i	participante i do sistema
LP_i	conjunto de parceiros de i
LPC_i	conjunto de parceiros candidatos de i
$limite_i$	número de candidatos mínimo de p_i

3.2 Geração do Conteúdo da Transmissão

Na aplicação de envio de mídia contínua ao vivo em P2P, o servidor S é responsável pela aquisição e codificação da mídia em um formato apropriado para a transmissão. O servidor S gera o conteúdo da transmissão e o divide em *chunks*. Cada novo *chunk* c_i é armazenado na área de memória apropriada de S (*buffer*). Assim, S marca $cm_S[c_i]$ como disponível no seu mapa de *chunks* ($cm_S[c_t] = disponivel$).

Os parceiros do servidor S atualizarão as informações sobre ele, a partir da troca dos mapas de *chunks*, e perceberão a existência de novos dados. Quando eles necessitarem dos *chunks* produzidos por S farão requisições e, o servidor S começará a disseminar os dados da mídia. Outros participantes do sistema irão encontrar os novos dados produzidos por S, quando algum de seus parceiros os receberem, seja por um pedido direto a S (se parceiro do servidor), ou por outro participante do sistema.

Nas aplicações mais populares de envio de mídia contínua em P2P, o servidor gera os dados da mídia contínua ao vivo a uma taxa aproximada de 6 *chunks* por segundo. Normalmente, um vídeo é codificado a aproximadamente 300kbps e assim, cada *chunk* tem cerca de 6 KB de dados.

A tabela 3.2 apresenta um resumo dos parâmetros utilizados para descrever a geração de conteúdo da transmissão ao vivo em P2P.

Tabela 3.2: Resumo dos parâmetros utilizados na geração de conteúdo.

Parâmetro	Descrição
S	Servidor que gera o conteúdo da transmissão
cm_i	mapa de *chunks* de p_i
$cm_i[x] = desejada$, onde $x = [0..m]$	conteúdo inicial do mapa de *chunks* de p_i

3.3 Realização de Parcerias

Um nodo p_i realiza parcerias logo após seu primeiro contato com o servidor de *boostrap* BS. A partir do estabelecimento das parcerias iniciais, o nodo efetivamente começa a participar do sistema de envio de mídia contínua ao vivo em P2P. Além deste momento inicial de estabelecimento de parcerias, um nodo pode ser contatado por um outro participante p_j, requisitando sua parceria, ou p_i pode tentar novas parcerias para aumentar sua conectividade.

Tanto no estabelecimento inicial de parcerias, quanto na descoberta de novos parceiros para aumento da conectividade, o nodo p_i recorre à lista de parceiros candidatos LPC_i para selecionar um nodo, ao qual vai enviar uma requisição de parcerias. Vários critérios podem ser utilizados para a escolha do candidato p_k, como uma escolha aleatória entre os nodos pertencentes a LPC_i, ou pelos recursos disponíveis em p_k informados por BS.

Uma vez selecionado o candidato, p_i envia uma mensagem de pedido de parceria ao candidato p_k selecionado de LPC_i. Caso p_k tenha recursos disponíveis (e.g.: banda

de rede, conexões disponíveis na aplicação), adiciona p_i à LP_k e responde a solicitação de p_i. Quando p_i recebe a resposta de p_k, ele adiciona p_k à LPC_i, a nova parceria estará formalmente estabelecida. Caso não seja de interesse de p_k o estabelecimento da nova parceria (e.g.: falta de recursos), p_k ignora o pedido de nova parceria. O nodo p_i espera por um período de tempo pela resposta de p_k e, caso não a receba, p_i retira o candidato p_k de sua LPC e repete o processo com um novo candidato selecionado.

No momento que um nodo p_i adiciona um novo parceiro p_k à LPC, p_i cria um temporizador T_{ik}; que é acionado em intervalos de tempo tp_i. A cada acionamento do temporizador T_{ik}, o nodo p_i envia uma mensagem ao parceiro p_k para verificar seu estado na aplicação. O objetivo desta mensagem é verificar se a parceria está ativa e trocar informações, como os mapas de *chunks* de ambos os nodos.

O nodo p_i espera a resposta de p_k acerca da mensagem disparada por um intervalo de tempo tr_i. Caso p_k não responda, p_i remove p_k de sua LP. Caso p_i receba a resposta de p_k, p_i atualiza os dados relativos a p_k, como por exemplo, o mapa de *chunks* cm_k.

A tabela 3.3 apresenta um resumo dos parâmetros utilizados para descrever a realização de parcerias no sistema de transmissão ao vivo em P2P.

Tabela 3.3: Resumo dos parâmetros utilizados na realização de parcerias.

Parâmetro	Descrição
T_{ik}	temporizador de *Ping* entre p_i e p_k
tp_i	intervalo de tempo para acionar T_{ik}
tr_i	tempo máximo de espera de resposta de *Ping*

3.4 Armazenamento e Consumo de Dados

Os participantes do sistema P2P de transmissão ao vivo devem conseguir obter a mídia da rede a uma taxa apropriada. Caso não o consigam, a execução da mídia terá falhas e a sensação de qualidade poderá ser ruim. Mais ainda, os participantes devem obter os dados da mídia o mais rápido possível, uma vez que a aplicação de transmissão ao vivo exige uma baixa diferença de tempo entre a criação do trecho de mídia e a sua exibição nos participantes do sistema P2P.

Caso a latência seja alta, os participantes irão experimentar um atraso indesejável e, no pior dos casos, a informação será exibida muito tempo depois do fato ocorrido. Por exemplo, um gol em uma partida de futebol poderá ser comemorado pelo vizinho e muito tempo depois, o participante o verá em seu equipamento P2P. Por esses motivos,

os participantes devem obter a mídia a uma taxa de c *chunks* por segundo. Essa é a mesma taxa de produção da mídia ao vivo pelo servidor S do sistema.

Cada participante do sistema apresenta uma área de armazenamento temporário B_i (*Buffer*), onde são armazenados os dados da mídia recebidos da rede P2P. Esse *Buffer* pode guardar uma quantidade pré-determinada de *chunks*. Inicialmente, cada posição j de B_i ($B_i[j]$) é inicializada com conteúdo vazio e, à medida que p_i recebe os *chunks* de seus parceiros, cada posição vai sendo preenchida. Nesse esquema de armazenamento temporário, o participante p_i tem por objetivo manter o *buffer* B_i preenchido de forma que se garanta uma contínua exibição da mídia ao vivo, mesmo se ele perder conexão temporariamente com seus parceiros.

Inicialmente, o buffer B_i pode ser representado por uma estrutura do tipo *"buffer circular"*. Dois processos trabalham em conjunto para manter o buffer B_i preenchido e a execução da mídia constante (produtor/consumidor). Assim, a posição de B_i a ser consumida é $B_i[0]$, e a posição mais recentemente a ser preenchida será $B_i[b]$ (b é a última posição de um buffer com pelo menos $b + 1$ posições).

Para manter uma visualização constante e sem interrupções, a aplicação deve obter alguns dados à frente do momento de exibição. Então, caso ocorra um problema temporário na rede P2P, há alguns dados armazenados no buffer. A cada intervalo de tempo, o nodo p_i verifica quais *chunks* devem ser consumidos em uma janela de tempo futura. Os *chunks* pertencentes a essa janela de interesse deverão ser recolhidos da rede enquanto p_i executa os dados relativos aos *chunks* anteriores a essa janela.

O tamanho da janela de interesse deve ser pequeno para não possibilitar a exibição da mídia com atraso demasiado (espera longa para recolher os dados da rede). Porém, janelas de interesse muito curtas podem gerar uma série de perdas na exibição, pois, pode ocorrer que um determinado *chunks* não tenha sido recolhido da rede e o momento de sua exibição tenha chegado.

A figura 3.2 exemplifica o mecanismo de consumo da mídia por um participante do sistema. Nessa figura, considera-se que a taxa de criação de *chunks* é de 1 unidade por intervalo de tempo. Desta forma, a cada unidade de tempo, o participante deve tentar obter o próximo *chunk* criado. Assim, a cada intervalo de tempo, o participante desloca a sua janela de interesse para o próximo *chunk* indicado em seu mapa.

No primeiro momento da figura 3.2, esse participante irá consumir o *chunk* à esquerda da janela de interesse. No segundo momento, a janela de interesse é deslocada para a direita e esse participante não tem o respectivo *chunk* para consumo (poderá haver uma falha na exibição). Neste mesmo instante, o participante consegue obter o *chunk* mais recente de seu interesse. Finalmente, o processo continua e o participante consome o próximo *chunk*.

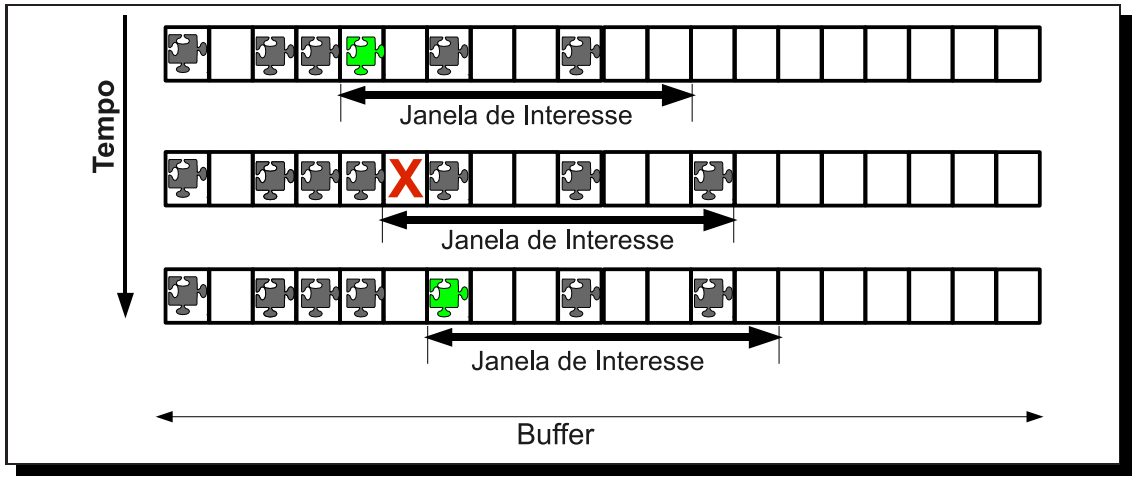


Figura 3.2: Mecanismo de consumo da mídia ao vivo.

3.5 Estratégias de Seleção de *Chunks*

A estratégia de seleção do *chunk* a ser requisitado da rede pode influenciar o desempenho e a eficiência da aplicação de mídia contínua ao vivo em P2P. Uma estratégia tenta estabelecer qual dos *chunks*, entre os vários necessários, deve ser requisitado em um determinado momento. Essas estratégias de seleção tentam manter a continuidade da exibição da mídia ao vivo para o participante do sistema, e também difundir, o mais rápido possível, um trecho de mídia que acaba de ser gerado.

Inicialmente serão apresentadas duas estratégias de seleção de *chunks* comumente utilizadas em aplicações P2P. Primeiro, será discutida a estratégia *"Mais Raro Primeiro"*, que é adotada em protocolos de aplicações de compartilhamento de arquivos em P2P, como o Bittorrent [5], e em envio de mídia ao vivo em P2P, como o CoolStreaming [99]. A segunda estratégia a ser analisada e discutida será a estratégia *"Gulosa"*, onde os participantes privilegiam a escolha de *chunks* que estão próximos ao fim de suas janelas de visualização.

3.5.1 Estratégia "Mais Raro Primeiro"

Na estratégia *"Mais Raro Primeiro"*, um participante irá requisitar o *chunk* que está menos replicado pelo sistema P2P de transmissão ao vivo. Do ponto de vista de um participante p_i, este *chunk* será o que está menos replicado entre seus parceiros.

Para exemplificar essa estratégia, considere o *buffer* B_i do participante p_i. A posição $B_i[0]$, que acaba de ser criada pelo processo da janela deslizante e está vazia, será a mais rara. A probabilidade de encontrar um parceiro que tenha esse dado disponível cresce com o tempo. Assim, no próximo intervalo de tempo, o *chunk* da

posição $B_i[0]$ irá para a posição $B_i[1]$, no intervalo consecutivo será movido para a posição $B_i[2]$, e assim por diante. Dessa forma, percebe-se claramente que, o *chunk* mais raro a ser buscado é o que acaba de ser criado, ou seja, $B_i[0]$. Dessa forma, a estratégia *"Mais Raro Primeiro"* seleciona os *chunks* em ordem crescente.

3.5.2 Estratégia "Gulosa"

A estratégia *"Gulosa"* tem como objetivo preencher os espaços do *buffer* que estão próximos de seu prazo final de visualização. Assim, nessa estratégia, um participante p_i irá selecionar o *chunk* mais próximo à posição de visualização no seu buffer B_i. Por motivos de simplificação, pode-se considerar o *chunk* final do buffer ($B_i[b]$) como sendo o elemento de próximo prazo final para visualização. Dessa forma, p_i irá selecionar o *chunk* $B_i[b]$, caso este não esteja preenchido em seu *buffer*, depois o *chunk* $B_i[n-1]$ e assim por diante. Desse modo, os participantes do sistema tendem a ter armazenado, localmente, os dados mais antigos produzidos pelo servidor S.

3.6 Seleção de Parceiros para Troca de Dados

Um participante p_i do sistema de transmissão ao vivo em P2P, além de selecionar quais *chunks* ele deve buscar em um determinado momento, deve escolher de qual parceiro realizar o pedido. Vários critérios podem ser utilizados para definir essa escolha, entre eles, critérios baseados em disponibilidade de recursos, proximidade temporal e até mesmo proximidade geográfica.

O modelo apresentado realiza uma seleção de parceiros em duas etapas. Por simplificação, adota-se a disponibilidade de recursos como critério de escolha de um parceiro para realização do pedido por um *chunk* específico. Essa mesma maneira de seleção pode ser realizada com quaisquer outros critérios de escolha.

Na primeira etapa, um participante p_i seleciona o parceiro p_k entre todos os seus parceiros, que apresenta maior disponibilidade de recursos naquele momento. Os recursos de um participante podem ser conhecidos por duas maneiras: por informação direta entre os parceiros, ou por requisição e envio de informações a uma unidade centralizadora. No caso apresentado, os parceiros trocam essas informações entre si e o recurso monitorado é a banda de rede disponível para o compartilhamento.

A partir da seleção de p_k, o participante p_i envia uma requisição pelo *chunk* de interesse c_j. O participante p_i cria um temporizador para esse pedido. Quando p_k recebe um pedido por dados vindo de um parceiro ativo p_i ($p_i \in LP_k$), p_k cria uma mensagem de resposta contendo o *chunk* c_j requisitado e o envia ao nodo p_i. Caso o

pedido seja enviado por um parceiro não ativo ($p_i \notin LP_k$), p_k ignora o pedido por c_j, mas adiciona p_i à lista de parceiros candidatos LPC_k.

Se o pedido pelo *chunk* c_j for respondido durante o intervalo de tempo esperado, p_i coloca o novo dado em seu armazenamento temporário e o assinala como disponível em seu mapa de *chunks* ($cm_i[c_j] = disponivel$).

A segunda etapa ocorre, caso o pedido não seja respondido até o final do tempo esperado, marcado pelo temporizador que p_i utiliza. Nesse caso, p_i refaz o processo de seleção de parceiros para pedidos e escolhe um novo parceiro p_l (onde $l \neq k$). Esse processo pode ser repetido até que o *chunk* c_i seja recebido corretamente, ou que o tempo de interesse nesse *chunk* seja superado.

3.7 Resumo do Capítulo

Neste capítulo, foram apresentados os principais conceitos de funcionamento do modelo de sistema de transmissão ao vivo em P2P adotado nesta tese. Descreveu-se a estrutura da rede P2P, o mecanismo de entrada na rede utilizado, a geração de conteúdo e a realização de pedidos e troca de dados na rede. Dessa forma, cobriram-se os principais aspectos funcionais das futuras implementações adotadas.

O sistema P2P de transmissão ao vivo adotado é baseado em uma arquitetura em malha, sem organização rígida dos participantes do sistema. Os participantes podem entrar e sair a qualquer momento, e realizam parcerias com um subconjunto de outros participantes. Os parceiros trocam informações entre si para colaborar uns com os outros e assim, obter a mídia que está sendo transmitida ao vivo.

Para se juntar ao sistema P2P de transmissão ao vivo, um participante se registra em um servidor separado da estrutura P2P. Esse servidor, denominado *bootstrap*, armazena as informações de todos os participantes ativos do sistema. O novo participante recebe como resposta desse registro, uma lista com outros participantes do sistema. Essa lista é utilizada para a tentativa inicial de estabelecimento de parcerias.

Entre os participantes do sistema há um especial: o servidor de mídia ao vivo. Ele captura o vídeo a ser transmitido, codifica para um formato apropriado e disponibiliza esses dados para toda a rede P2P. Esse servidor atua da mesma forma que todos os participantes do sistema, mas não requisita dados de seus parceiros.

Os participantes têm um armazenamento local, onde guardam os dados do vídeo para uma execução contínua. Eles devem verificar, periodicamente, quais os pedaços de dados eles necessitam. Há maneiras apropriadas de selecionar e de se fazer uma requisição por dados específicos. Neste capítulo, foram apresentadas duas abordagens

denominadas *"Gulosa"* e *"Mais Raro Primeiro"*. Essas estratégias têm como objetivo, respectivamente, manter o fluxo da execução sem interrupções, e disseminar o conteúdo rapidamente pela rede P2P.

Caso mais de um parceiro possa contribuir com o dado necessário, um participante deve escolher a qual fará a solicitação. O mecanismo adotado baseia-se na disponibilidade de recursos de cada parceiro. Assim, será escolhido o parceiro com maior quantidade de recursos disponíveis (e.g. banda de rede, memória, etc.). Fazendo uma escolha apropriada, um participante selecionará quem poderá atendê-lo da melhor.

Capítulo 4

Caracterização do Comportamento dos Participantes de um Sistema de Transmissão ao Vivo em P2P

Apesar da crescente popularidade das aplicações de transmissão de mídia contínua ao vivo em P2P, a compreensão atual de aspectos importantes do comportamento dos participantes, como o intervalo de chegada entre eles e tempo de permanência na rede P2P, ainda é limitada. Este capítulo aborda esse problema, apresentando uma caracterização do comportamento dos participantes no SopCast, uma das aplicações mais populares de transmissão ao vivo em arquiteturas P2P. A análise apresentada inclui propriedades relativas à sessão dos participantes, tais como intervalos entre chegadas de participantes, tempo de permanência na rede, número de sessões e tempo de inatividade entre sessões. Também são apresentadas propriedades das parcerias entre os participantes, como o número de parceiros e suas durações. Além disso, o estudo apresentado compreende diferentes canais do SopCast, destacando diferenças nos padrões de comportamento dependendo do conteúdo a ser transmitido. As caracterizações apresentadas definem um modelo de participantes, que pode ser usado como base para criação de novos protocolos e geração de cargas sintéticas realistas para simulações de transmissão ao vivo em P2P.

Atualmente, existe uma grande quantidade de trabalhos com foco no desenvolvimento de protocolos e organização das redes P2P de mídia contínua ao vivo. Entretanto, apenas alguns estudos abordam a caracterização dos participantes desses sistemas [76]. Apesar do comportamento dos clientes poder afetar significativamente a transmissão de mídia [73, 84], a sua compreensão ainda é superficial. Assim, a caracterização desses comportamentos é um ponto de interesse crescente [76].

Trabalhos anteriores sobre esse tema [23,33,34,76–78,90,97,109] concentram seus estudos somente nas propriedades topológicas da rede P2P. Apenas alguns aspectos comportamentais dos participantes são caracterizados, como a taxa entre chegadas e o tempo de permanência no sistema. Logo, eles não provem uma análise aprofundada do comportamento dos clientes. Esses estudos assumem um comportamento homogêneo e independente dos participantes, e ignoram diferenças nos cenários de domínios distintos, como o canal transmitindo um evento de grande interesse ou um programa diário de notícias. Assim, não é claro se os resultados se confirmam em outros domínios. Dessa forma, torna-se necessária uma caracterização detalhada do comportamento dos participantes de uma transmissão ao vivo em P2P já que a dinâmica deles é apontada como um fator de impacto no desempenho de sistemas P2P [83].

Neste capítulo, é apresentada uma análise do comportamento dos participantes do SopCast, um importante sistema de transmissão ao vivo em P2P. Essa análise tem foco nas características do comportamento dinâmico dos usuários nessa rede P2P. O principal objetivo é prover dados para geração de cargas sintéticas mais realistas dos comportamentos dos clientes, que podem ser utilizados tanto na avaliação dos protocolos para transmissão de conteúdo contínuo ao vivo sobre P2P, quanto no desenvolvimento de novas aplicações P2P para transmissões ao vivo.

A presente caracterização baseou-se em um conjunto de coletas de tráfego sobre três canais diferentes do SopCast. Foram coletados dados em diferentes datas, entre novembro de 2008 e fevereiro de 2009, utilizando a rede de experimentos PlanetLab [11]. Foram caracterizadas as cargas de trabalho desses três canais, que podem ser classificados em dois domínios diferentes: o primeiro domínio é o de um típico canal de transmissão ao vivo, que consiste em uma emissora chinesa de televisão, apresentando notícias em alta qualidade de vídeo a uma audiência majoritariamente chinesa. O segundo domínio é representado por dois canais que distribuem conteúdo de entretenimento durante um grande evento de interesse, principalmente, para brasileiros (i.e. jogos decisivos para o campeonato brasileiro de futebol).

Conforme em [16], foi utilizada uma abordagem hierárquica para descrever a carga de trabalho em dois níveis: um nível de sessão do sistema e outro de parcerias interativas entre os participantes. Assim, caracteriza-se uma lista dos parâmetros em cada nível dos cenários, incluindo frequências de tempo de acesso, intervalos entre a chegada de novos participantes, número de parceiros e interações dinâmicas entre eles, também conhecidas como *churn* (entrada e saída da rede, abandono de parcerias, etc.). O conjunto de parâmetros pode ser utilizado para construir um modelo de usuários do sistema P2P. Uma atenção especial foi dada às variações temporais da transmissão, e as análises apresentadas são de dados selecionados em dias de carga alta do sistema

(mais usuários), onde se espera que as distribuições dos parâmetros mantenham-se relativamente estáveis.

Os principais pontos observados neste capítulo foram:

- As características dos participantes apresentam diferenças marcantes, dependendo do conteúdo transmitido.
- Os tempos de permanência dos participantes no sistema são modelados por distribuições diferentes dependendo da transmissão. As distribuições Gamma e Weibull representam as duas classificações de transmissão criadas na presente caracterização.
- Os tempos de inatividade entre sessões de usuários, quando existem, são bem representados por uma distribuição Exponencial em todas as transmissões observadas.
- As características de parceria, como número de parceiros, e tempo da parceria são representadas pelas mesmas distribuições independente da classificação da transmissão. Para o número de parceiros, foi encontrada uma distribuição Normal, e para o tempo relativo das parcerias, foi encontrada uma distribuição Gamma. Porém, os parâmetros para a distribuição Gamma dependem do tipo de transmissão.

Até o momento, o que se observa na literatura é:

O padrão de tráfego, vizinhança dos clientes e tempo de vida dos participantes (definido como tempo de residência ou *ON time*) são explorados em [76–78, 90]. Os autores em [77,78] analisaram o comportamento de quatro sistemas P2P de transmissão ao vivo durante um evento. Eles focaram em componentes da rede, como taxa de envio, taxa de recebimento, e distribuição de tráfego TCP ou UDP. Na presente caracterização, é observado um comportamento da distribuição de pacotes similar a [77, 78]. O tempo de vida dos clientes em [76] é bem descrito por uma distribuição de Weibull. Já em [90], o tempo de vida é modelado por uma distribuição geométrica. Os resultados aqui apresentados mostram que uma distribuição Gamma representa o tempo de vida nos canais típicos. Também foi percebido um comportamento diário de tráfego similar ao observado em [90].

Uma caracterização do padrão de chegada e saída dos clientes, a distribuição das popularidades dos canais e a localização geográfica dos participantes é apresentada em [33, 34]. Assim, como ocorre com a análise apresentada, os autores observaram que

os participantes têm um tempo de vida maior em canais populares. Em todos estes trabalhos, os autores não diferenciam sessão de parceria e sessão do participante.

Apesar desses estudos anteriores cobrirem diferentes características, a maior parte deles considera apenas poucos parâmetros específicos dos cenários estudados, ressaltando o comportamento do sistema e as propriedades topológicas da infra-estrutura P2P formada. A caracterização apresentada foca em diversos aspectos-chave para a geração de cenários sintéticos realistas, como tempo de permanência do cliente e seu tempo de parcerias, pois, estes parâmetros podem caracterizar um cliente em qualquer sistema P2P de transmissão ao vivo. Além disso, dada a diversidade de canais transmitidos ao vivo, não fica claro se os resultados anteriores se confirmam em diferentes domínios, como em um canal de entretenimento ou um de notícias.

Até onde se conhece, a única tentativa de comparar características de diferentes domínios de transmissão ao vivo é a análise apresentada em [90]. Entretanto, esse trabalho fornece uma caracterização muito limitada do comportamento dos clientes. Os autores não exploram o sistema sob a visão de parcerias e, além disso, concentram seus esforços nas propriedades topológicas do sistema P2P.

Em comparação aos trabalhos anteriores, o estudo apresentado: (1) fornece uma caracterização mais aprofundada do comportamento dos clientes em sistemas de transmissão ao vivo sobre P2P, (2) analisa um conjunto mais diversificado de canais, incluindo canais de entretenimento, notícias e uma cobertura de um grande evento esportivo, e (3) propõe um modelo mais preciso do comportamento dos participantes.

4.1 Modelo de P2P ao Vivo Adotado

Um sistema de envio de mídia contínua ao vivo em P2P é composto por participantes (nodos/peers/usuários) que colaboram entre si para disseminar a mídia criada por um participante especial do sistema, denominado servidor. Os participantes se organizam em uma rede virtual, sobreposta à rede física existente (geralmente uma rede IP). Há duas grandes vertentes de redes sobrepostas para esse fim: uma rede com estrutura definida baseada em árvores (e o servidor sendo a raiz), e uma rede sem estrutura definida, baseada em malhas computacionais.

O servidor do sistema S gera o conteúdo ao vivo que irá ser transmitido. Ele divide a mídia em pedaços ou fatias, conhecidos como *chunks*, que irão ser transmitidos para os demais participantes do sistema P2P. Um participante guarda os pedaços de mídia, em um armazenamento temporário, para poder cooperar com outros participantes. Assim,

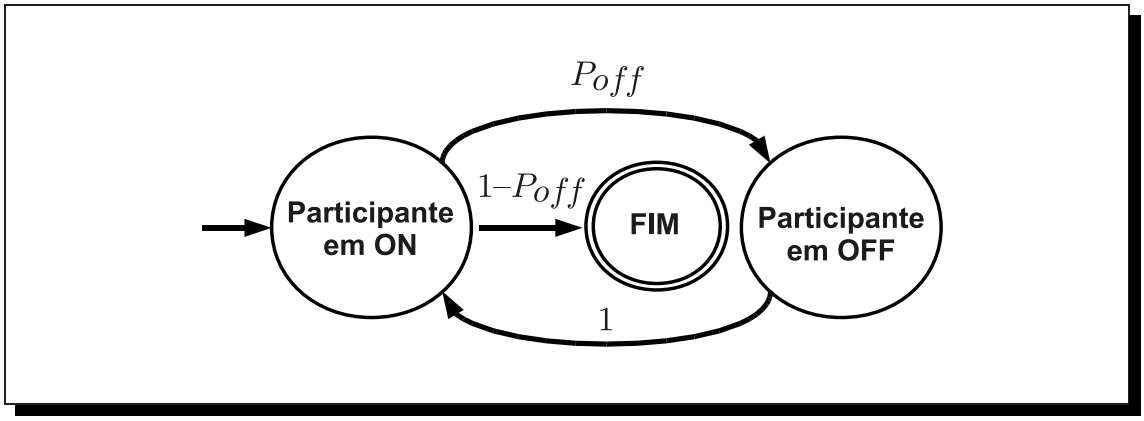


Figura 4.1: Modelo do participante.

um sistema P2P de transmissão ao vivo também aproveita recursos de seu conjunto de participantes P, onde $P = p_1, p_2, ..., p_n$.

Cada participante p_i pode se unir ou abandonar o sistema a qualquer momento. Eles podem ir para um estado inoperante, o que significa que eles não interagem com nenhum outro participante. Quando os participantes saem do estado inoperante, retornam ao sistema e realizam novas interações. Dessa forma, a dinâmica do sistema continua. Se um participante não retorna ao sistema, interpreta-se que o mesmo abandonou definitivamente a aplicação.

A figura 4.1 mostra uma máquina de estado simples que captura a essência da dinâmica dos participantes da rede P2P. Quando um participante se junta ao sistema, este inicia suas interações com outros participantes do sistema. Ele tenta recolher os dados da mídia para realizar sua correta exibição. Durante esse estado de atividade, o participante encontra-se no estado denominado ON. Quando o participante deixa o estado ON, ele pode ficar inoperante por um momento com uma probabilidade P_{off}, ou pode abandonar definitivamente o sistema com uma probabilidade $1 - P_{off}$. Quando o usuário fica inoperante e vai para o estado OFF, ele não realiza interações com seus parceiros e nem recolhe dados do sistema. Porém, em um futuro próximo, irá voltar ao estado de atividade.

Atualmente, as aplicações de envio ao vivo em P2P mais populares, como Sopcast [81], PPlive [66] e GridMedia [26, 108] utilizam uma arquitetura baseada em redes de malha, e são orientadas por busca aos dados (*mesh-based data-driven*). Essa rede sobreposta tende a ser resistente aos efeitos de alta dinâmica dos participantes, promovendo uma visualização da mídia sem interrupções.

A figura 4.2 ilustra o modelo de trabalho proposto, evidenciando os dois tipos de sessões: as sessões de um participante e as sessões de parcerias. Durante uma

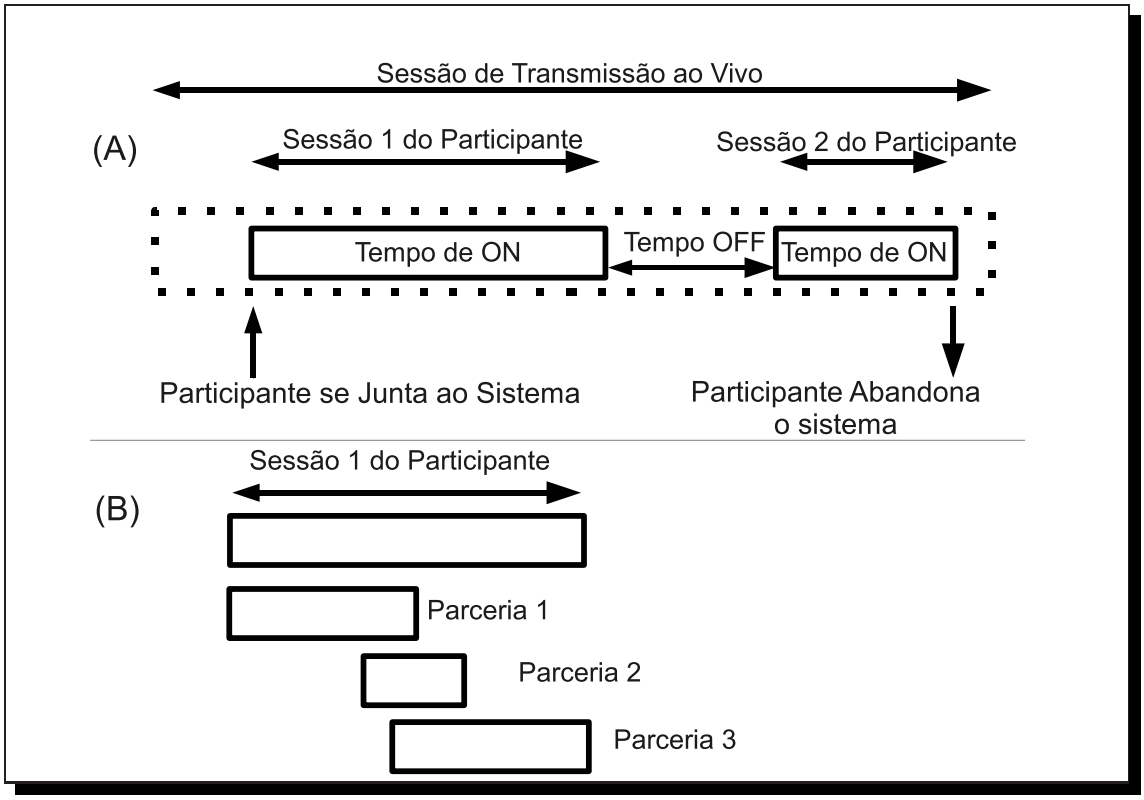


Figura 4.2: Modelo hierárquico das sessões dos participantes.

transmissão ao vivo, os participantes assistem ao vídeo e interagem com seus parceiros durante um período de tempo, denominado *sessão do participante*. Essa sessão inicia-se quando o participante se junta ao sistema P2P e faz sua primeira requisição pela mídia ao vivo. O tempo total da sessão é conhecido como *tempo de residência* ou simplesmente tempo de ON. Se um participante retorna ao sistema após o término de uma sessão, ele inicia uma nova sessão de participante.

O participante pode ficar inoperante durante algum tempo. Esse tempo inoperante pode ser causado por uma quebra temporária em sua conexão com a rede, ou congestionamento em seus recursos. Esse período é denominado tempo OFF entre sessões. O tempo OFF não necessariamente interrompe a visualização da mídia, uma vez que o cliente tem um armazenamento temporário da mídia (*buffer*). Por exemplo, o participante 1, na figura 4.2-a, apresenta duas sessões. Isso é determinado, pois, o tempo que ele permaneceu inoperante por um tempo maior que um limiar definido no modelo. O valor que define o limiar do valor OFF será discutido adiante.

Dentro de uma sessão, um participante estabelece uma ou mais parcerias. Uma parceria ps_{i-j} é definida como o tempo em que p_i interage com seu parceiro p_j. Ou seja, uma parceria tem início quando dois participantes trocam dados, e dura até a

última interação entre eles. A figura 4.2-b mostra as 3 parcerias existentes na primeira sessão do participante. Todas elas apresentam características distintas como tempo de início e duração.

4.2 Metodologia para Coleta dos Dados

Os dados utilizados neste capítulo foram coletados da rede do SopCast com o programa de monitoramento de pacotes de rede tcpdump [1]. Como mencionado anteriormente, o SopCast é uma das aplicações mais populares para transmissão ao vivo em P2P. Ele apresenta uma grande variedade de canais e suporte para vários sistemas operacionais como Linux e Windows.

4.2.1 Canais Analisados

Foram coletadas informações de três canais que transmitem conteúdo ao vivo na rede do SopCast. Foram escolhidos esses três canais, por transmitirem conteúdos distintos para públicos distintos. Os três canais apresentam diferenças marcantes entre eles, como o país original e o público a que é direcionado, a codificação e a qualidade visual da imagem e, finalmente, o tipo de programação. Dessa forma, espera-se capturar uma informação mais abrangente em relação às diferenças regionais do público alvo, tipos de programação e horário da transmissão.

O primeiro canal analisado é o canal de notícias chinês CCTV. Esse é um canal popular em seu país de origem e transmite o vídeo a uma taxa alta pelos padrões encontrados no Sopcast (por volta de 600kbps). Sua transmissão é baseada no horário oficial local da China.

O segundo canal analisado transmitia, nos períodos de coletas, o conteúdo de uma das mais populares redes de tv do Brasil. Esse canal foi estudado durante dois grandes eventos esportivos para o público brasileiro e, durante as transmissões estudadas, o conteúdo ao vivo era transmitido a uma taxa média/baixa pelos padrões do Sopcast (por volta de 250kbps).

Finalmente, o terceiro canal estudado é especializado em transmissão esportiva, durante toda sua programação, variando entre noticiários esportivos, transmissões ao vivo de jogos e matérias relacionadas ao tema. Esse canal transmite o seu conteúdo em linguagem chinesa a uma taxa alta (por volta de 500kbps).

[1] www.tcpdump.org

4.2.2 Coleta dos Dados do Sopcast

Como observado em [34], a coleta do comportamento dos participantes de sistemas P2P, como o Sopcast, é um desafio. As informações armazenadas nos servidores de rastreamento da rede Sopcast não estão disponíveis para uso público, e isso impossibilita a tarefa de reconstruir as características da rede com total precisão.

Para tratar esse desafio, foi utilizada uma metodologia de coleta de informações e reconstrução da rede, semelhante a apresentada em [90]. Por essa metodologia, um conjunto de computadores é estruturado com ferramentas de coleta de pacotes de rede e com a aplicação do Sopcast. Esses computadores se juntam à rede do Sopcast e, enquanto participam como usuários comuns da rede, gravam toda a comunicação realizada entre eles e os demais participantes. Após esse período de coleta, os dados de todos esses computadores são agrupados, e faz-se a reconstrução da rede Sopcast.

A metodologia de coleta realiza as seguintes operações:

- Uma máquina de controle dispara o processo de coleta no conjunto de máquinas do PlaneLab;
- As máquinas sincronizam o relógio local;
- As máquinas começam a captura do seu tráfego de rede nas portas específicas utilizada pelo SopCast;
- As máquinas se juntam ao canal do SopCast;
- Ao final, os dados da coleta são recuperados para análise.

Para coletar os dados do SopCast, utilizou-se o PlanetLab [11]. Foram utilizados 421 computadores desse sistema espalhados pelo mundo afim garantir a maior cobertura possível da rede analisada. Além disso, a banda de subida e descida dessas máquinas não foi limitada.

Outros trabalhos com metodologia semelhantes adotam um número menor de máquinas de coleta. Por exemplo, [90] foram utilizadas 10 máquinas em seus experimentos. A utilização de um número maior de máquinas na caracterização apresentada tenta criar uma visão mais realista do sistema, com resultados mais próximos à realidade.

Finalmente, os períodos de coleta de dados do Sopcast variaram entre dezembro de 2008 a janeiro de 2009. Cada coleta foi realizada durante um período de 100 minutos ininterruptos. Esse tempo foi escolhido por cobrir a transmissão completa de um evento esportivo. Com exceção dos dias de transmissão de evento esportivo, as coletas foram

realizadas a partir das 20:00 horas do horário local do canal de transmissão, e por 7 dias consecutivos para cada um dos canais. O principal motivador para a escolha desse horário é que, por volta das 20:00 horas, há uma grande concentração de usuários nos canais observados.

4.3 Características de Acesso dos Participantes

Esta seção discute o acesso dos canais do SopCast em diferentes períodos, canais e eventos. O objetivo é discutir a atividade do usuário em um canal, durante uma semana e durante um dia. Assim, as demais coletas para realizar a caracterização dos participantes do sistema podem ser melhor direcionadas aos períodos de grande carga e estabilidade.

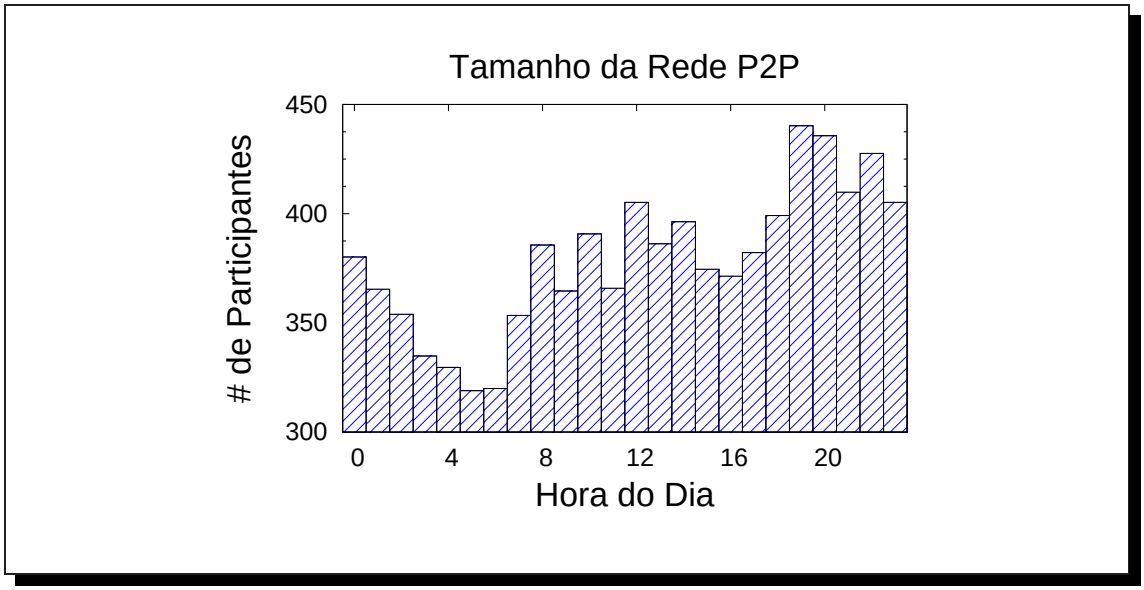


Figura 4.3: Utilização do canal durante um dia.

A figura 4.3 mostra a variação do número de participantes da CCTV, durante um dia. Essa figura representa o período analisado do dia 16/12/2008. Durante esse dia, o número de participantes começa a crescer a partir das 7:00. O período entre 8:00 e 16:00 apresenta um número médio de usuários com pouca variação (+- 25 usuários de variação). Esse canal alcança o seu maior número de participantes por volta das 20:00, com um período de estabilidade de cerca de 3 horas. O período de maior atividade coincide com o período em que as pessoas chegam em casa, após o dia de atividades.

Durante uma semana, o número de participantes em um canal também varia. Como observado em [97] e na atual caracterização, os usuários acessam mais a aplicação

aos finais de semana. A figura 4.4 mostra esse comportamento observado no canal de noticiários da CCTV. Uma das prováveis explicações para concentração de um maior número de usuários, durante os finais de semana, é que durante esses dias as pessoas estão em casa e procuram por entretenimento ou por informação.

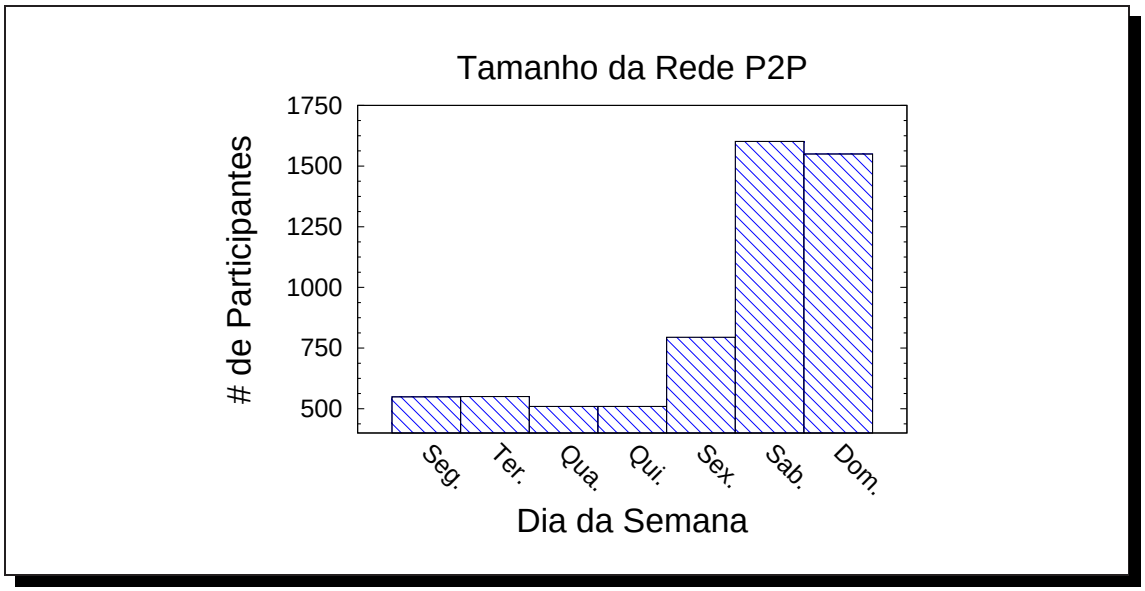


Figura 4.4: Utilização do canal durante os dias da semana.

4.4 Características dos Participantes

Esta seção analisa cada componente do modelo descrito na seção 4.1, sendo eles: intervalo entre chegadas, número de sessões, tempo de ON e OFF, número e tempo de parcerias. Serão apresentados os dados observados para cada um desses componentes, assim como as distribuições que melhor descrevem esses dados.

Na maior parte dos componentes discutidos a seguir, os 3 canais apresentaram dois comportamentos distintos para um dia normal de transmissão ao vivo e um dia com um grande evento esportivo direcionado para um público específico. Assim, tanto o canal de noticiários da CCTV, quanto o canal esportivo chinês, foram agrupados em uma classe denominada de "canal sem evento". Por outro lado, o canal brasileiro, que transmitiu dois eventos esportivos em dois dias diferentes, foi denominado "canal com evento". Quando não há distinção de comportamento entre as duas classes, o texto evidencia a igualdade entre elas.

4.4.1 Delimitação das Sessões dos Participantes

As sessões de participantes não são explicitamente informadas pelo SopCast ou por suas mensagens de protocolo. Assim, definiu-se que uma sessão de participante termina quando este fica inativo por um tempo maior que um valor limite pré-determinado. Para escolher esse valor limite do tempo de inatividade (OFF time), foram analisados os tempos entre requisições, de um mesmo usuário, que os coletores de informação recebiam.

Todos os canais avaliados apresentaram o mesmo comportamento para os tempos entre requisições consecutivas. Portanto, os dados apresentados nesta seção, referem-se a todas as coletas dos 3 canais em questão. O erro médio encontrado entre as amostras das coletas de dados é inferior a 0.6%.

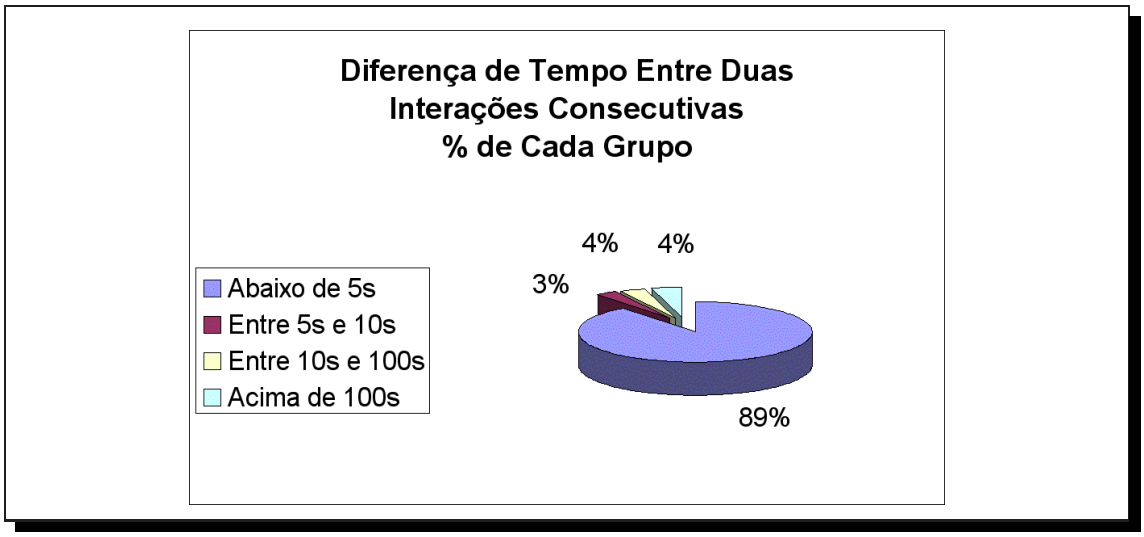


Figura 4.5: Proporção dos tempos entre mensagens encontrados.

Foram encontrados dois perfis distintos de tempos entre chegadas de requisições. O primeiro perfil refere-se às requisições realizadas em um período muito curto de tempo. A figura 4.5 mostra que nas medições realizadas, foi observado que quase 90% das requisições ocorrem em um período menor que 5s. O segundo perfil refere-se às requisições de um usuário com tempos maiores que 5s. A quantidade restante das requisições divide-se de maneira igualitária entre requisições de 5 a 10 segundos, de 10 a 100 segundos e acima de 100 segundos.

Para a análise do ponto de corte de uma sessão, foram escolhidos os tempos entre requisições superiores a 5s. Essa escolha foi feita, pois, o indicativo de um fim de sessão é um tempo de inatividade alto, superior a maior parte dos tempos entre duas mensagens/requisições consecutivas. Como os tempos inferiores a 5s correspondem

à maioria absoluta, ignorar esses tempos pode proporcionar uma melhor precisão na escolha do tempo de inatividade limite entre sessões.

Assim, a figura 4.6 mostra o tempo entre requisições de um mesmo usuário no segundo perfil. Observa-se um crescimento acentuado na probabilidade acumulada de acontecer uma requisição até 120 segundos, e após esse tempo há uma queda no crescimento na probabilidade acumulada. Há um ponto de quebra nessa curva por volta de 150 a 180 segundos. Como várias aplicações adotam um limite de 180 segundos (e.g. Bittorrent), esse valor foi escolhido para o limite de inatividade para as demais análises.

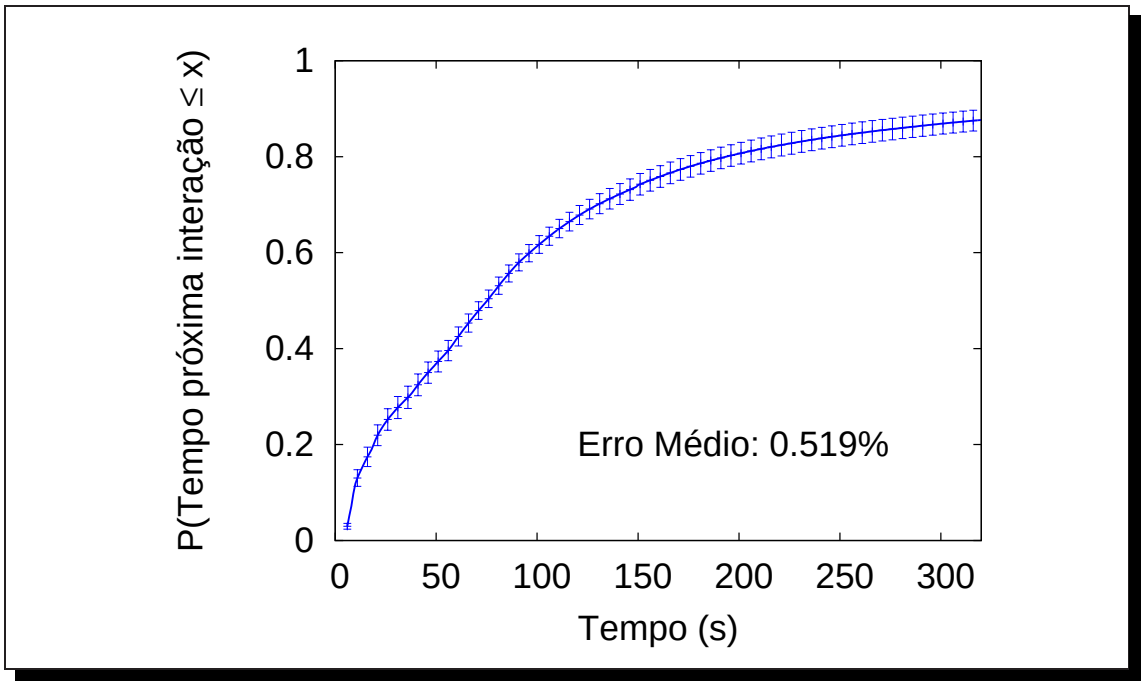


Figura 4.6: Distribuição do tempo entre requisições.

4.4.2 Processo de Chegada de Sessões

Foram analisados os tempos entre chegadas de sessões de participantes durante períodos em que o sistema encontrava-se estável (com um número de usuários aproximadamente constante). A figura 4.7 resume o intervalo entre sessões para as duas classes de trabalho, "dia com evento" e "dia sem evento".

Em o "dia de evento" (os dois jogos de futebol decisivos para o público brasileiro), os tempos entre início de novas sessões são menos espaçados. O evento atrai um grande número de usuários, principalmente, no início do evento. Em um dia "sem evento", representado pelos noticiários e pela programação esportiva padrão, os tempos entre inícios de novas sessões são mais espaçados, mas da mesma forma que acontece em

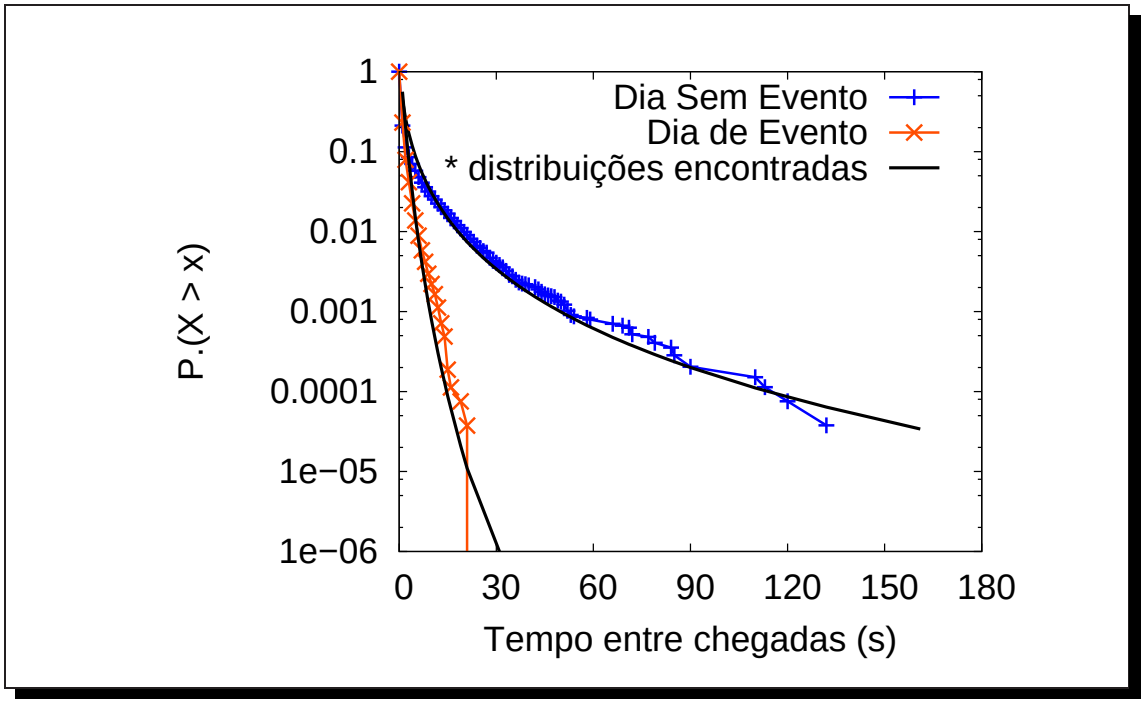


Figura 4.7: Tempo entre as chegadas de sessões.

um "dia de evento", a grande concentração de tempo entre inícios de sessões é pequena (menor que 2 segundos). O erro padrão médio entre as amostras analisadas para caracterizar a classe "dia com evento" está abaixo de 0.5% e para a classe "dia sem evento" é menor que 0.5%.

Para achar a distribuição que melhor modela o tempo de chegada entre novas sessões, foram comparadas as diferenças quadráticas entre um conjunto de distribuições criadas e os dados coletados. Além da comparação das diferenças quadráticas, foram realizadas inspeções visuais para comparar os resultados de modelagem no corpo e na cauda da curva. Foram favorecidas as distribuições que descrevem melhor o corpo da curva, uma vez que intervalos curtos entre sessões têm um impacto maior na aplicação P2P em questão (e.g.: chegadas em rajadas).

A distribuição Lognormal descreve bem o tempo entre chegadas de sessões para as duas classes observadas. Na figura 4.7 são apresentadas duas curvas com os dados feitos a partir da distribuição Lognormal modelada. Para ambas as classes, o erro padrão entre os dados observados no Sopcast e os dados criados pela distribuição está abaixo de 1%. A tabela 4.1 apresenta um resumo das distribuições encontradas para descrever os tempos entre novas sessões, provendo os parâmetros das distribuições encontradas.

Tabela 4.1: Distribuição dos tempos de chegada de sessões.

Carga	Melhor Dist.	Média (segundos)	Desvio Padrão (segundos)	Primeiro Parâmetro	Segundo Parâmetro
Sem Evento	Lognormal	2.019736	4.370575	$m = -0.165739$	$\sigma = 1.318109$
Com Evento	Lognormal	1.417119	1.113226	$m = 0.108310$	$\sigma = 0.693277$

4.4.3 Características das Sessões

O modelo apresentado na seção 4.1 possibilita a representação tanto do comportamento do participante, quanto de suas parcerias. Trabalhos anteriores como [23, 76, 86, 97], apenas caracterizam o comportamento dos usuários, ignorando aspectos importantes, como as parcerias realizadas entre eles.

A sessão de um usuário, apresentada no modelo proposto, é determinada pelos tempos de atividade, inatividade e a probabilidade de ocorrer essa inatividade. Para determinar a possibilidade de um usuário atingir um estado de inatividade ou abandonar o sistema, foram analisadas as distribuições do número de sessões de cada participante do Sopcast nos canais estudados. A ocorrência de um estado de inatividade apresenta as mesmas características em ambas as classes de trabalho ("dia com evento; dia sem evento"), e o erro padrão médio encontrado está abaixo de 0.6%.

A figura 4.8 mostra a função de distribuição de probabilidade de ocorrer um determinado número de sessões de usuários. Para todos os canais observados, a quantidade de sessões de um determinado usuário manteve o mesmo comportamento em todas as coletas realizadas. Por exemplo, mais de 70% dos participantes abandonam o sistema após a primeira sessão.

Nessa mesma figura é apresentada a curva gerada, a partir do modelo proposto anteriormente, com parâmetro *Poff=0.39*. Observa-se que, apesar da simplicidade do modelo proposto, ele representa bem o comportamento do usuário em relação ao número de sessões. O erro padrão médio entre os dados observados e a curva gerada pelo modelo está abaixo de 1%.

4.4.4 Tempos de ON

Os tempos de residência nas sessões dos participantes do Sopcast (ON) e os tempos de inatividade entre sessões (OFF) foram medidos e analisados. Em ambos os casos, o tempo é representado como uma medida percentual do período de coleta de dados.

As classes em que os canais foram agrupados apresentam comportamentos

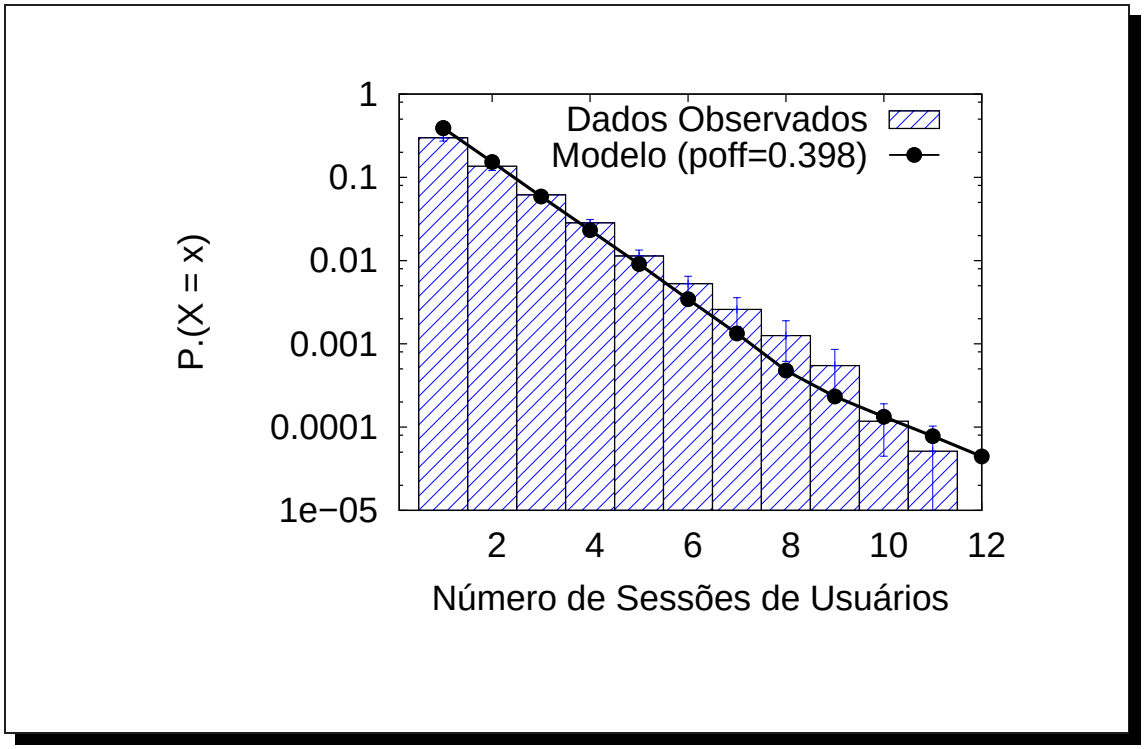


Figura 4.8: Número de sessões de participante.

distintos para o tempo de ON. Em dias "com evento", os tempos de ON tendem a ser maior que o observado em dias "sem evento" de grande interesse. Além disso, ao contrário de trabalhos como [76] que acreditam que o tempo de ON depende somente da entrada e da partida de um participante e do mecanismo de trocas de parcerias da aplicação, nesse capítulo, somente o comportamento do participante é considerado como fator impactante dessa medida. Mesmo que aplicação force a troca de parceiros, os participantes continuam realizando suas atividades na mesma sessão.

A figura 4.9 mostra o tempo de ON para todas as cargas analisadas. Enquanto a curva apresentada para os canais classificados como "sem evento" tem seus dados deslocados para esquerda, nos "dias com evento", essa curva tem seus dados deslocados para a direita. Esse comportamento indica que o tempo de ON é maior em dias de eventos, como o jogo de futebol. Nesse caso, há um número considerável de participantes que assistem à partida sem interrupções por um grande período e, além disso, nota-se que há um número considerável de participantes que permanecem no canal por toda a duração do evento, o que pode ser esperado nesse caso.

O erro médio padrão entre as amostras coletadas para os canais classificados como "dia sem evento" é inferior a 0.72%. No canal classificado como "dia com evento", esse erro é consideravelmente maior, ficando em torno de 5.7%. A maior diferença

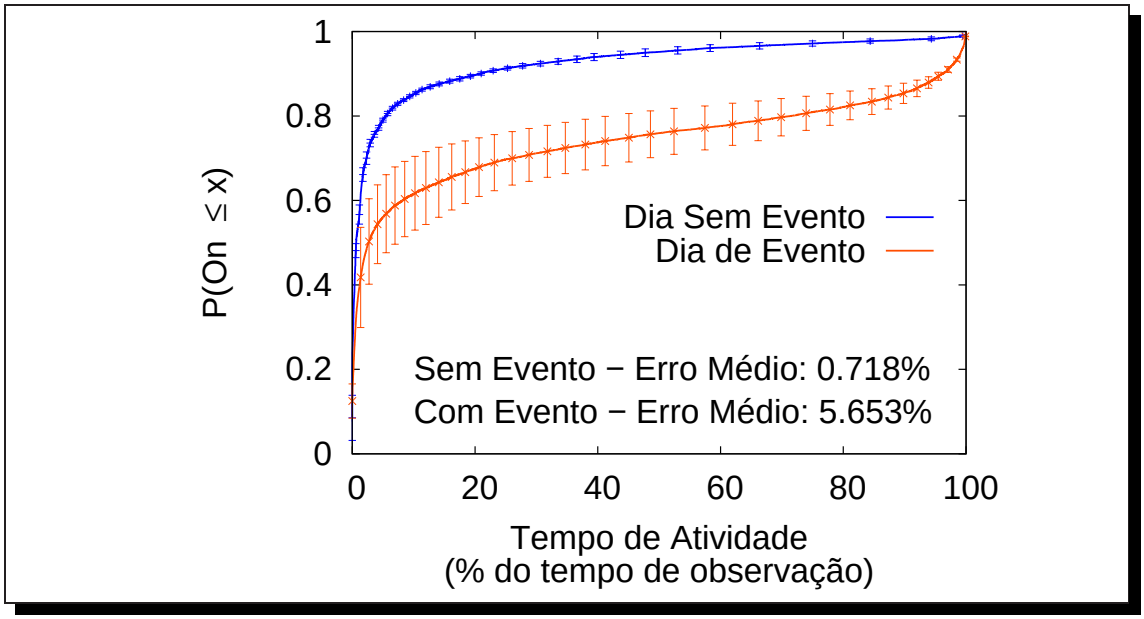


Figura 4.9: Tempo ON do participante.

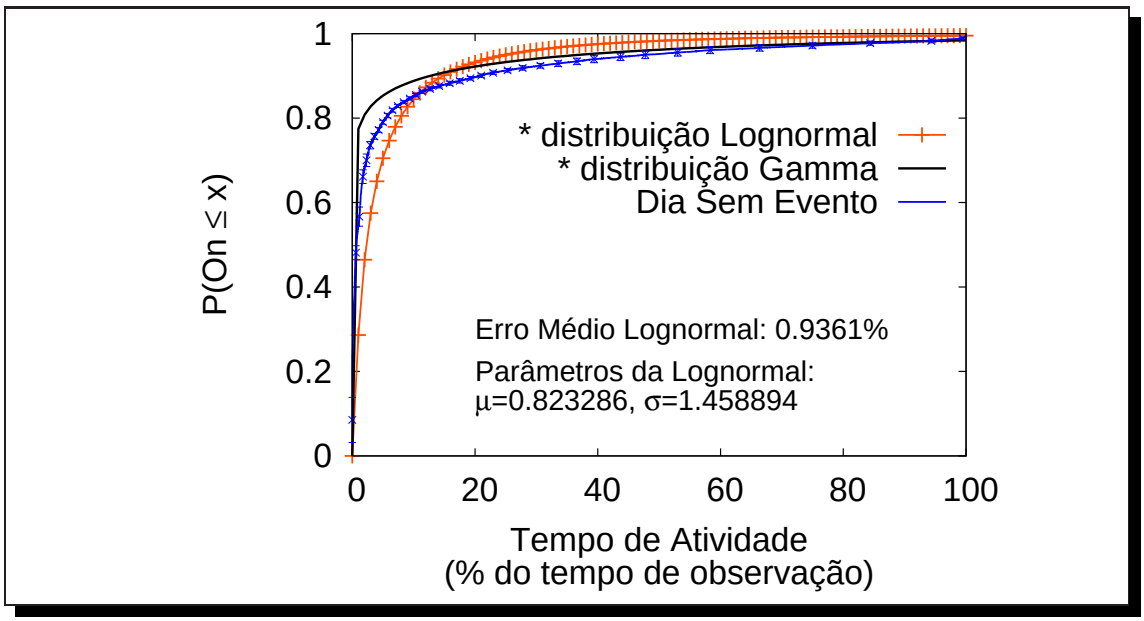


Figura 4.10: Tempo ON do participante - caracterização em dia sem eventos.

encontrada entre os dias de evento analisados pode ser explicada pelo menor número de eventos analisados, em relação aos dias sem evento.

Os canais em dias típicos, que não estejam transmitindo um evento de grande interesse, são bem representados pelas distribuições Gamma e Lognormal. Na figura 4.10 são apresentadas as curvas geradas a partir dessas duas distribuições, além do dado observado em um dia sem evento. A distribuição Lognormal se aproxima melhor do início da curva, onde se concentra o maior número de participantes. O erro médio

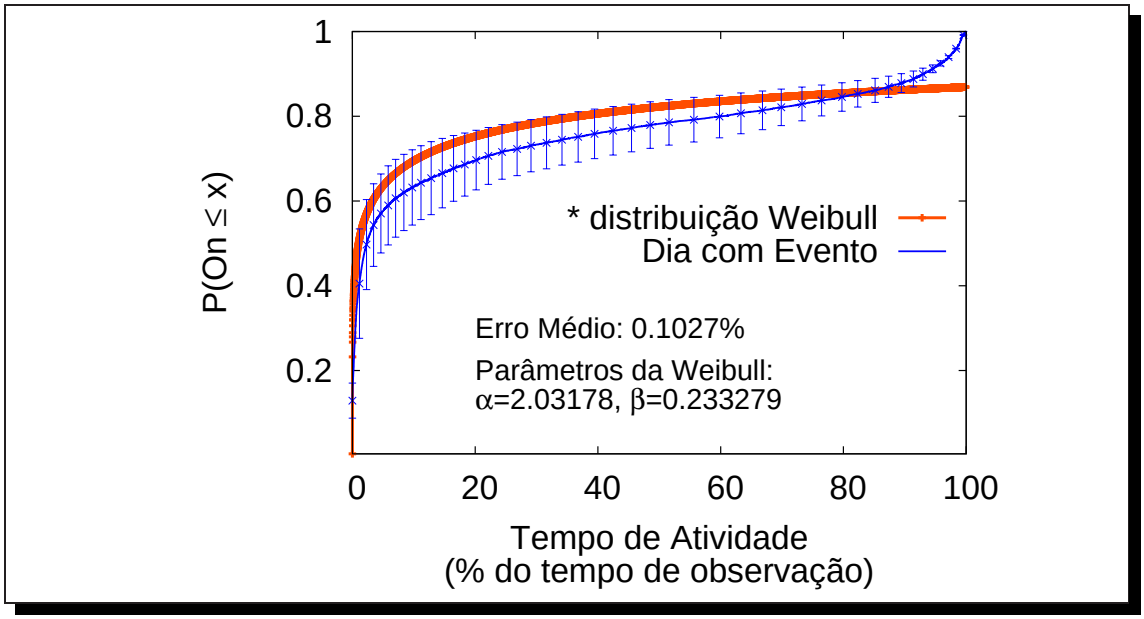


Figura 4.11: Tempo ON do participante - caracterização em dia com eventos

padrão entre os dados criados a partir da distribuição Lognormal e os dados observados nos experimentos é inferior a 1%.

Para o dia de evento, a distribuição que melhor modela o tempo de ON é a Weibull. Essa mesma distribuição já foi relatada em trabalhos anteriores como [76], porém, nesses trabalhos não há a distinção entre cargas diferentes dos sistemas P2P analisados. Na figura 4.11 são apresentadas as curvas geradas a partir da distribuição Weibull e dos dados observados nos dias de evento. O erro médio entre os dados gerados e os dados observados é inferior a 1%.

A tabela 4.2 resume os resultados encontrados para os tempos de ON nas cargas analisadas, provendo os parâmetros das distribuições para cada classe.

Tabela 4.2: Distribuição dos tempos ON.

Carga	Melhor Dist.	Média (segundos)	Desvio Padrão (segundos)	Primeiro Parâmetro	Segundo Parâmetro
Sem Evento	Lognormal Gamma	6.602661	17.962619	$m = 0.823286$ $\alpha = 0.061509$	$\sigma = 1.458894$ $\beta = 107.345437$
Com Evento	Weibull	23.5929	34.9861	$\alpha = 2.031780$	$\beta = 0.233279$

4.4.5 Tempos de OFF

A distribuição que melhor descreve o tempo de ON varia entre os canais observados, porém, os tempos de OFF não variam entre eles. Uma distribuição comum foi

encontrada para descrever o tempo de OFF, entre os canais observados, sem uma diferença significante. A seguir, serão descritas as distribuições e as observações com relação ao tempo de ON e de OFF.

A figura 4.12 mostra as distribuições de tempos de OFF para as cargas observadas. A figura 4.12 mostra o tempo de OFF para todas as cargas analisadas. O erro padrão médio encontrado na sumarização dos dados fica em torno de 2.5%. Os tempos entre sessões são, relativamente, pequenos. Por exemplo, cerca de 80% desses tempos são inferiores a 30 minutos. Esse comportamento pode incentivar políticas de parcerias ou de estruturação no protocolo que levem em conta uma possível volta de um participante em um curto intervalo de tempo.

Os resultados encontrados indicam que uma distribuição Exponencial se enquadra bem aos tempos de OFF observados no Sopcast. Para todos os canais analisados, essa distribuição representa bem tanto a cauda quanto o corpo da curva de dados. O erro padrão médio encontrado entre os dados sumarizados e a curva criada pela distribuição Exponencial é inferior a 1.5%.

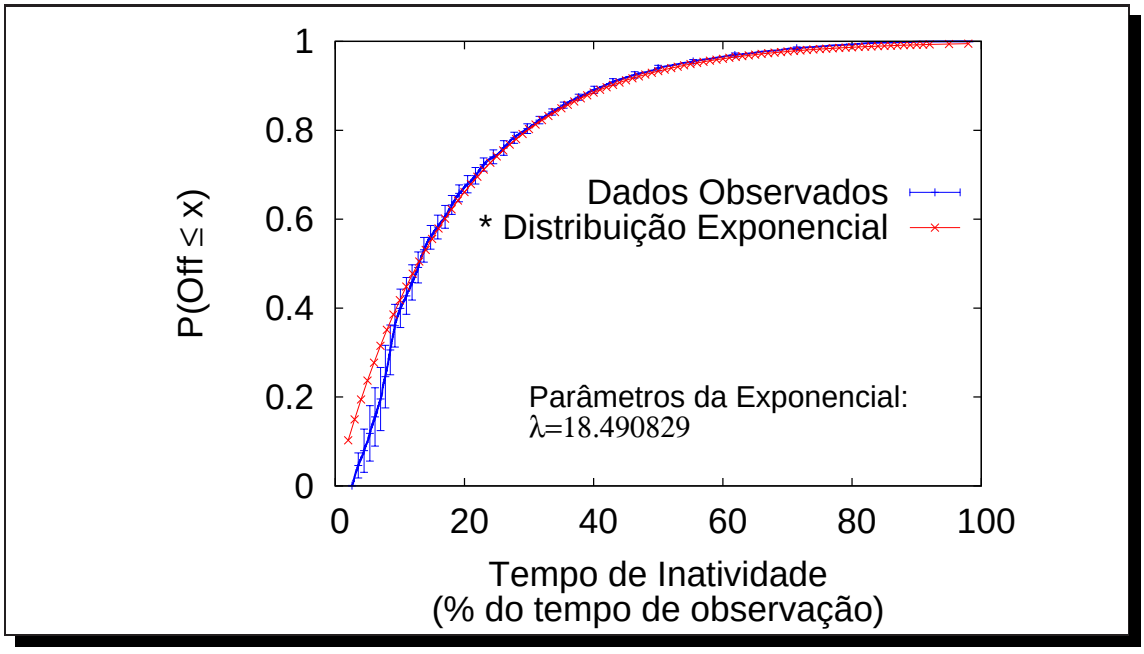


Figura 4.12: Tempo OFF dos participantes.

A tabela 4.3 resume os resultados encontrados para os tempos de OFF nas cargas analisadas, provendo os parâmetros das distribuições para cada classe.

Tabela 4.3: Distribuição dos tempos OFF.

Carga	Melhor Dist.	Média (segundos)	Var. (segundos)
Com e Sem Eventos	Exponencial	18.490829	16.170967

4.4.6 Características das Parcerias

A segunda camada do modelo descrito na seção 4.1 refere-se às parcerias realizadas pelos participantes do sistema. Cada participante realiza parcerias durante sua sessão, e cada parceria dura certo tempo dentro da sessão. O número de parceiros e a duração das parcerias podem variar entre os diversos participantes do sistema, mas todos apresentam em comum um comportamento promiscuo, com parcerias pouco duradouras e em grande número.

(A) Número de Parceiros de um Participante

O número de parceiros apresentados por um participante não varia de forma significativa entre as observações realizadas em todos os canais. A figura 4.13 mostra o número de parceiros dos participantes para *todas as cargas analisadas*. O erro padrão médio para a os resultados sumarizados está abaixo de 1.7%.

Um dos motivos prováveis para o número de parcerias não apresentar comportamento diferente entre as diversas cargas e canais analisados é que essa característica é definida pela aplicação. Ou seja, os usuários não conseguem realizar uma configuração explícita da quantidade desejada de usuários. Percebe-se ainda que, após a realização de parcerias com certo número de participantes, a aplicação do Sopcast sempre tenta manter esse número de parceiros constante.

Para todos os canais observados, cerca de 40% dos participantes apresentaram mais de 80 parceiros, e o número médio de parceiros fica em torno de 100. Esses números elevados, em comparação a outras aplicações P2P, pode indicar que o SopCast implementa uma política gulosa de parcerias.

Foram comparadas as diferenças quadráticas entre os dados e um conjunto de distribuições. Novamente foram realizadas inspeções visuais para verificar a melhor distribuição que descreve o número de parcerias. O número de parcerias dos participantes pode ser bem representado por uma distribuição Gamma ou Normal para canais típicos como a CCTV. A figura 4.13 também ilustra as curvas geradas a partir das duas distribuições citadas. Apesar da distribuição Gamma se aproximar melhor dos dados observados nos canais, espera-se que a distribuição Normal possa ser utilizada,

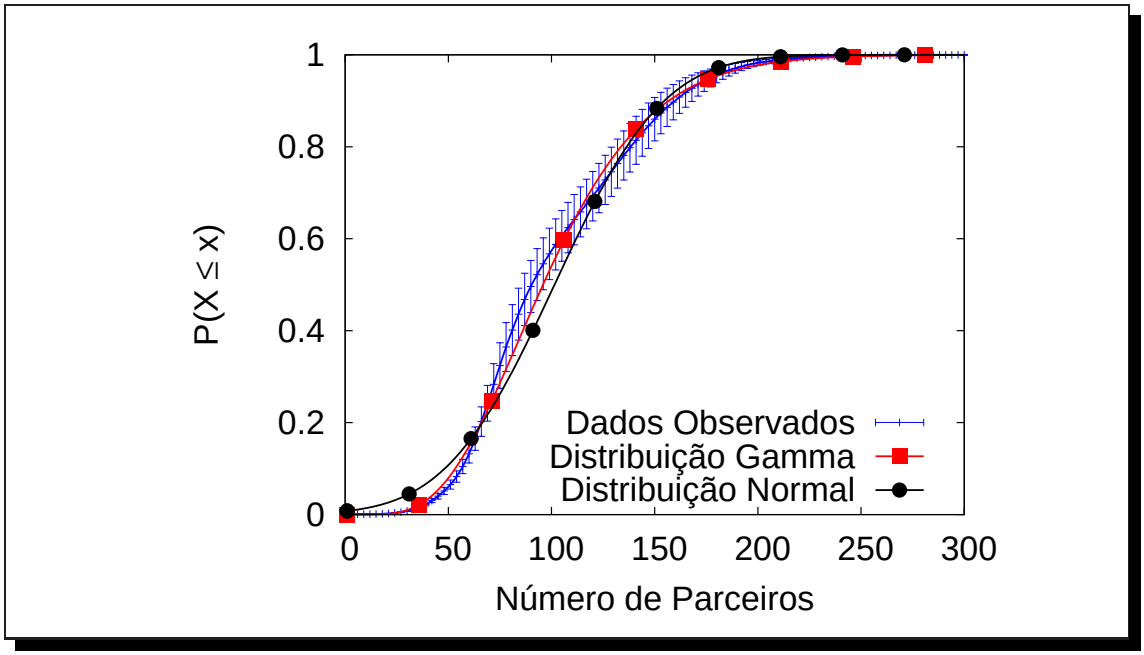


Figura 4.13: Número de parceiros.

sem perdas consideráveis, para representar o número de parceiros dos participantes. Em ambos os casos, o erro padrão entre os dados gerados e os dados observados ficou abaixo de 1.5%. Finalmente, foi observado que número médio de parceiros é próximo ao número encontrado em [78].

A tabela 4.4 resume os resultados encontrados para o número de parceiros nas cargas analisadas, provendo os parâmetros das distribuições.

Tabela 4.4: Distribuição do número de parceiros: Resumo.

Carga	Melhor Dist.	Média (parceiros)	Desvio Padrão (parceiros)	Primeiro Parâmetro	Segundo Parâmetro
Dias Com e Sem Eventos	Gamma Normal	101.452652	41.537279	$\alpha = 6.008078$ $\mu = 101.452652$	$\beta = 16.886042$ $\sigma = 41.537279$

(B) Tempo das Parcerias

Cada parceria dura um tempo menor ou igual ao tempo de sessão do participante. Para caracterizar o tempo de uma parceria, a análise feita tenta responder a seguinte questão: *"há alguma correlação entre o tempo de parceria e o tempo de ON de um participante?"*. Para responder essa questão, foram analisados os tempos de parcerias em relação ao tempo restante da sessão do participante. Ou seja, quanto tempo uma parceria ocupou do tempo restante de vida dos participantes.

A representação do tempo de parceria, como uma proporção do tempo restante de vida de um participante, tem como principal objetivo gerar informações que possam ser diretamente aplicadas para a geração de cargas sintéticas. Por exemplo, se a representação do tempo de parceria fosse absoluta, o gerador de carga sintética determinaria o tempo de parceria, sem levar em consideração a situação dos participantes. Caso algum deles abandone o sistema tão logo a parceria comece, o tempo de parceria, anteriormente estipulado, será falso pois a parceria efetivamente durará menos. Assim, ao adotar a representação como um percentual do tempo restante de vida do participante, o gerador de carga sintética apenas gera o valor correspondente, e não a há a necessidade de se controlar os tempos de início da sessão de um usuário nem o tempo de início da parceria.

A figura 4.14 mostra o percentual da duração de uma parceria em relação ao tempo de vida restante do participante. Da mesma forma como ocorre para o tempo de ON, os tempos de parceria podem ser enquadrados nas duas classes previamente discutidas. Para um "dia com evento", os tempos de parceria duram um tempo relativo menor que o observado para um "dia sem evento".

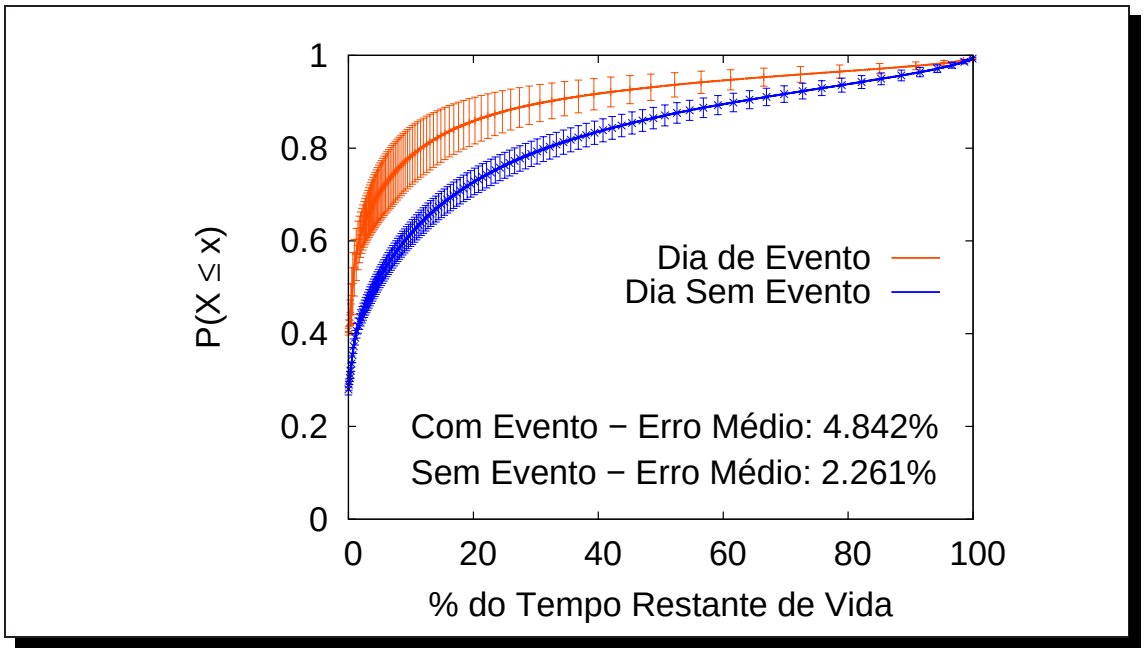


Figura 4.14: Duração das parcerias

Uma provável explicação para tal comportamento é que, em um dia com evento, o tempo de ON dos participantes é maior. Assim, a fração do tempo de parceria com o tempo restante de vida pode ser menor (tempo absoluto da parceria pode não variar entre as classes, enquanto o tempo de ON sim).

Os dados sumarizados para um dia com evento apresentam um erro médio de cerca de 4.8%. Para o dia sem evento, o erro é cerca de 2.6%. No caso para o dia "com evento", esse erro pode ser explicado devido ao baixo número de eventos analisados. Enquanto para o dia "sem evento", a diferença entre as várias amostras pode se dar devido ao fato de que os tempos das parcerias apresentam grandes variações entre os participantes. De fato, algumas parcerias são extremamente duradouras, enquanto outras são de pouca duração.

Os canais em dias "sem evento" são bem representados pela distribuição Gamma. Na figura 4.15 são apresentadas as curvas geradas a partir da distribuição Gamma, com os parâmetros relativos a essa carga, e os dado observado em um dia sem evento. O erro médio padrão entre os dados criados a partir da distribuição Gamma e os dados observados nos experimentos é inferior a 2.5%.

Na figura 4.16 são apresentadas as curvas geradas, a partir da distribuição Gamma, com os parâmetros relativos a essa carga, e os dados observados em um dia com evento. Para facilitar a inspeção visual das curvas, o gráfico apresenta a distribuição complementar à acumulada de probabilidades. O erro médio padrão entre os dados criados a partir da distribuição Gamma, e os dados observados nos experimentos é inferior a 4.5%, superior ao observado na outra classe.

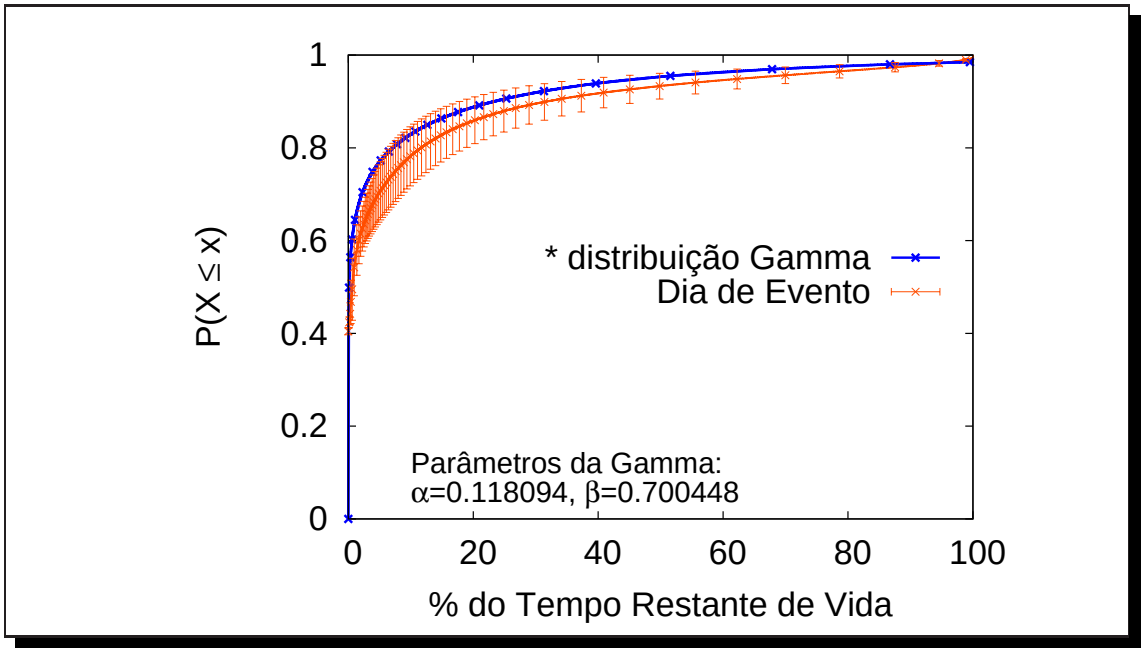


Figura 4.15: Duração das parcerias - caracterização em dia com eventos

A tabela 4.5 resume os resultados encontrados para os tempos de parcerias nas cargas analisadas, provendo os parâmetros das distribuições para cada classe.

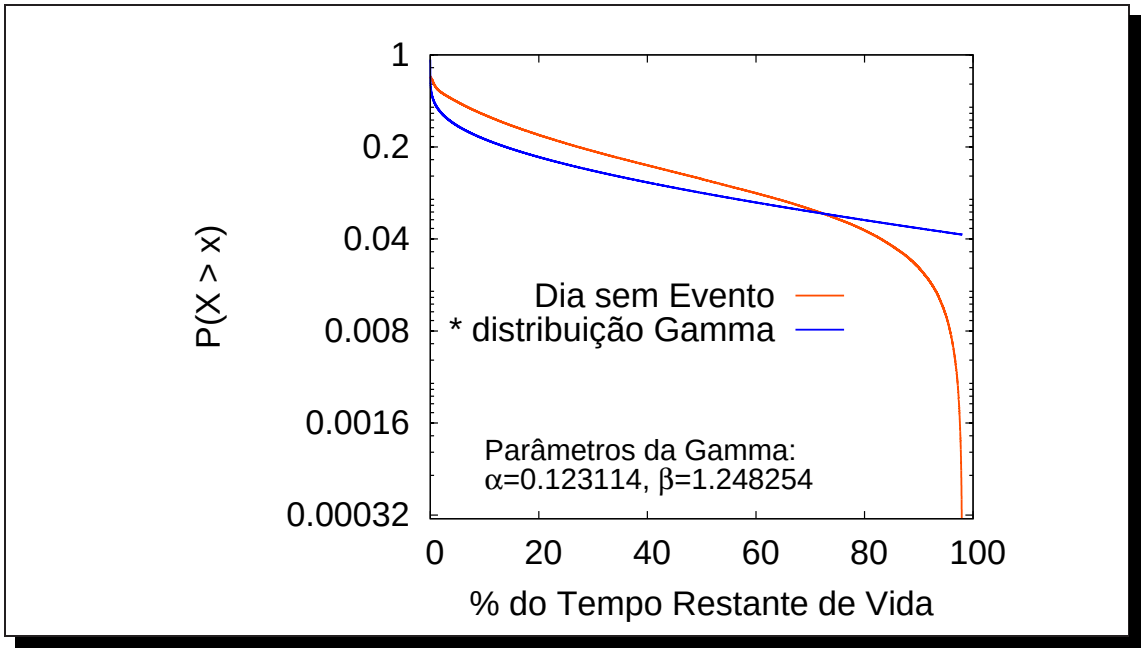


Figura 4.16: Duração das parcerias - caracterização em dia sem eventos

Tabela 4.5: Duração das parcerias.

Carga	Melhor Dist.	Média (segundos)	Var. (segundos)	Parâmetro 1	Parâmetro 2
Dia sem Evento	Gamma	15.3678	24.5569	$\alpha = 0.123114$	$\beta = 0.123114$
Dia com Evento	Gamma	8.2718	19.9503	$\alpha = 0.118094$	$\beta = 0.700448$

4.4.7 Resumo das Características Encontradas

Esta seção resume os perfis de comportamento dos participantes do SopCast encontrados neste capítulo. Esses perfis são caracterizados com foco nos tempos de ON, tempos de OFF e nas características das parcerias, como duração e tamanho. Nas transmissões do Sopcast analisadas, os participantes se encaixaram em um dos dois perfis mostrados na tabela 4.6. Esse enquadramento depende, principalmente, do tipo da transmissão do canal, se é um evento de grande interesse para um público específico, ou se é uma transmissão ordinária.

Tabela 4.6: Resumo dos perfis de participantes

Canal	Tempo entre chegadas	Número de sessões	ON	OFF	Tempo de Parceria
Sem Evento	Lognormal	≤ 2 em	Gamma|Lognormal	Exp.	Gamma
Com Evento	Lognormal	90% dos casos	Weibull	Exp.	Gamma

A principal diferença entre as classes de participantes identificadas é o tempo de ON dos participantes. Porém, mesmo distribuições iguais, para certas características, apresentam valores de parâmetros diferentes (e.g Tempo de Parceria).

Os tempos de ON são menores em canais típicos do SopCast, como a CCTV e o canal especializado em programas esportivos, do que os tempos encontrados em canais transmitindo eventos, como o canal que transmitia o jogo de futebol. No caso específico do canal em dia de jogo de futebol, há um número considerável de participantes que permanecem no sistema por, praticamente, toda a transmissão.

Outro ponto importante observado é que os participantes do SopCast são impacientes. Ao contrário de aplicações de compartilhamento de arquivos, a maior parte dos participantes se junta à transmissão por alguns segundos, e suas parcerias também apresentam esse mesmo perfil. A figura 4.17 mostra um exemplo de como os participantes assistem a uma sessão de vídeo e como fazem parcerias, em um dia sem eventos. Durante o período observado, a linha indica o momento inicial em que um participante ganha um novo parceiro. Os pontos mostram quando o participante perde o parceiro com quem iniciou o contato anteriormente. Note que os pontos estão próximos à linha, o que indica que os participantes criam e abandonam parcerias constantemente. Apesar desse comportamento generalizado, há algumas parcerias que, após criadas, duram um tempo longo.

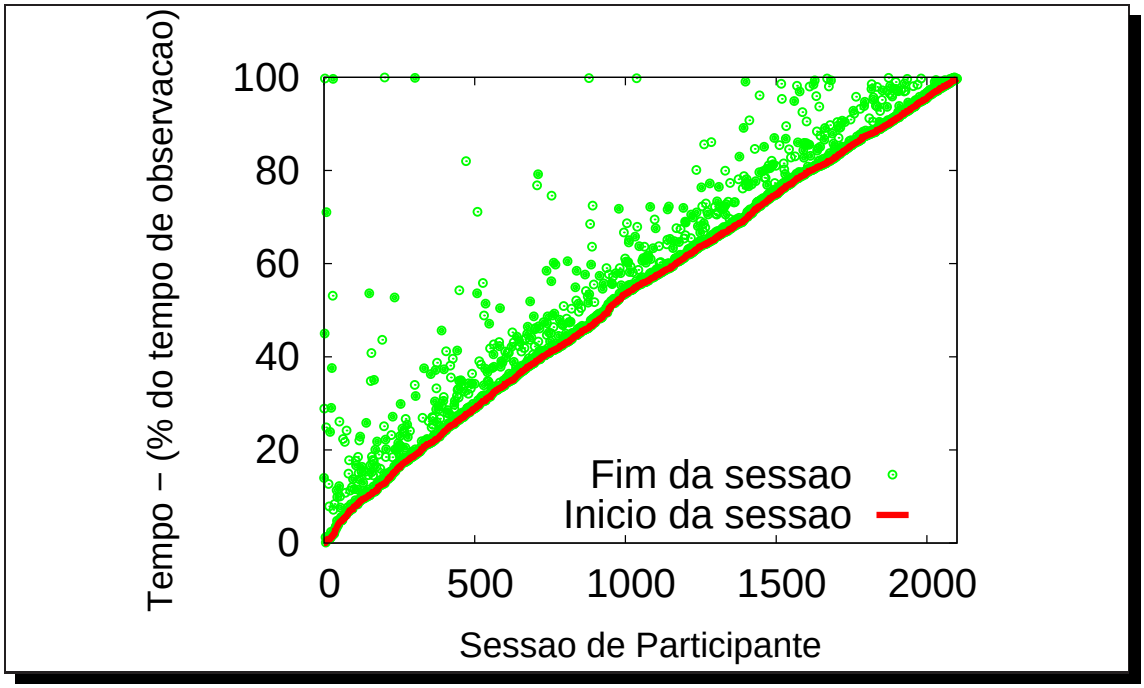


Figura 4.17: Participantes são impacientes.

Esse comportamento pode criar problemas ao sistema de transmissão ao vivo

P2P. Quando um participante se junta ao SopCast, ele se inscreve no servidor de *bootstrap*. Esse servidor guarda as informações do novo participante, para disponibilizar aos demais quando necessário. Porém, o mecanismo de *boostrap* do SopCast mantém informações, mesmo quando os participantes saem. O tempo de reação é alto, e assim, o servidor continua fornecendo o endereço do participante que já abandonou o SopCast.

Se vários participantes entram e abandonam o sistema, constantemente, os novos receberão uma série de endereços inválidos. Isso os fará gastar tempo e recurso com parcerias que não serão concretizadas. A latência inicial da execução do vídeo pode aumentar, pois, há um tempo considerável para encontrar parceiros válidos. A saber, durante os experimentos, o *boostrap* do SopCast continuou enviando informações sobre um participante ausente, por um período superior a 15 minutos após sua saída.

Além dos fatores relacionados ao comportamento de entrada e saída dos usuários do sistema, nota-se, claramente, que o tamanho da rede P2P formada não é influenciado pelo comportamento dos usuários, ou vice-versa. Por exemplo, os canais CCTV e o canal Esportivo, apesar de serem classificados como um mesmo tipo de canal apresentam números de participantes bem distintos entre as observações realizadas. A figura 4.18 apresenta o número de participantes encontrados nos canais, durante um período de observação. Note que, apesar do canal em um dia de jogo de futebol ter um número de participantes superior ao canal chinês CCTV, apresenta um número consideravelmente inferior ao canal esportivo. Essa contra prova ilustra o fato de que a classificação do canal não tem relação com a quantidade de participantes observados.

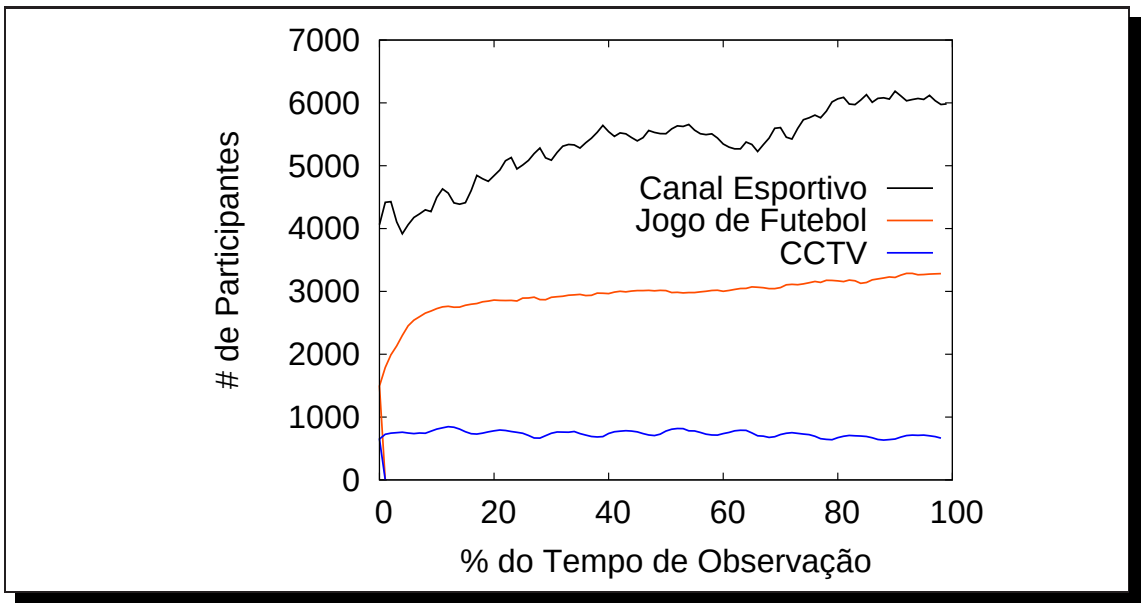


Figura 4.18: Tamanho da rede Sopcast para os canais analisados.

4.5 Resumo do Capítulo

Este capítulo provê uma caracterização do comportamento dos participantes de um sistema de transmissão ao vivo em redes P2P. Foram analisadas várias cargas de trabalho que se encaixam em dois domínios: um canal típico da aplicação Sopcast, denominado “canal sem evento", e um canal atípico, por causa de um grande evento de interesse, denominado “canal com evento”.

Em todas as cargas analisadas, os participantes da rede apresentaram um comportamento com alta dinâmica, com muitas entradas e saídas do sistema, criação e abandono de parcerias. Assim, as parcerias acompanham o comportamento geral dos participantes, sendo elas, normalmente, de curta duração. Observou-se também que o número de sessões de participante, durante o período de análise, é baixo. De fato, mais de 90% dos participantes apresentam menos de 2 sessões observadas.

Nos canais classificados como “canais com evento”, o tempo de residência (ON) observado é maior que nos canais classificados como “canais sem evento”. Em um dia com evento, observa-se uma porção considerável de participantes que acompanham o evento até sua finalização. Enquanto nos canais em dia de evento cerca de 20% dos participantes assistem mais de 90% do referido evento (partida de futebol), em um canal sem evento, cerca de 90% dos participantes assistem não mais que 20% da transmissão, durante o período de observação.

Os tempos de inatividade, quando ocorrem, são observados da mesma maneira em todas as classes. Tanto para um dia com evento, quanto para um dia sem evento, os tempos de OFF são baixos, com uma média de 18 segundos. Nesse caso, não há um número considerável de usuários que permaneçam por longos períodos de inatividade.

Além dos fatos observados, percebe-se que o tempo de parceria pode ser expresso em termos do tempo restante de vida de um participante. Esse resultado facilita a geração de cargas sintéticas realistas, uma vez que para todas as cargas de trabalho, essa correlação entre tempo de parceria e tempo de atividade pode ser expressa pela mesma distribuição de probabilidade.

Finalmente, os resultados apresentados reafirmam que os participantes do Sopcast são impacientes. Tanto para o tempo de permanência no canal, quanto para o tempo de parcerias, eles apresentam um comportamento dinâmico. Isto é, as entradas e saídas são constantes e as parcerias, na maior parte dos casos, são pouco duradouras. Esse comportamento deve ser levado em consideração na criação de novos protocolos e novas aplicações, uma vez que pode interferir diretamente na escalabilidade do sistema e na qualidade do serviço oferecido.

Capítulo 5

Modelo Formal de um Ataque de Poluição

Entre os diversos ataques a sistemas de transmissão ao vivo em P2P possíveis, o ataque de poluição tem seu efeito percebido, de forma clara pelo usuário. Nesse tipo de ataque, os poluidores alteram ou forjam o conteúdo que está sendo transmitido ao vivo pela rede P2P. Esse ataque pode ter consequências indesejáveis aos usuários do sistema. Mesmo que uma aplicação de envio ao vivo tenha mecanismos para checagem da integridade dos dados, os usuários podem ter sua banda de rede sobrecarregada, com pedidos por retransmissão dos dados inválidos e/ou uma alta latência da mídia que está sendo assistida ao vivo [6, 7].

Por esse motivo, neste capítulo será proposto um modelo analítico, que ajuda a entender como um ataque de poluição pode atingir um sistema de envio ao vivo em P2P. Mais ainda, esse modelo captura aspectos do desenvolvimento dos mecanismos por busca de dados na aplicação P2P. Mostra também como estes mecanismos podem influenciar na disseminação do conteúdo poluído e como a sobrecarga de rede é afetada por eles. Além disso, os modelos propostos permitem a comparação entre diferentes tipos de mecanismos de busca de dados, além de determinar limites dos efeitos dos ataques ao sistema P2P. Em particular, serão estudadas 2 estratégias de seleção de dados: *"Mais Raro Primeiro"* e *"Gulosa"*.

5.1 Modelo Básico

Nesta seção, o sistema de transmissão ao vivo em P2P é apresentado como um modelo simples, baseado no modelo proposto por [109] e no sistema definido no capítulo 3. O sistema em questão pode ser definido, como segue:

Um sistema de transmissão de mídia contínua ao vivo em P2P é um sistema que apresenta m participantes que contribuem entre si para transmitir o conteúdo ao vivo. Existe um participante especial que gera o conteúdo *(servidor)* e o repassa aos demais participantes da rede P2P. O conteúdo ao vivo é repassado em pedaços denominados *chunks*. Cada *chunk* apresenta um número de sequência, normalmente iniciado por 0.

Cada participante p_i apresenta n_i parceiros. Cada participante também apresenta uma área de armazenamento temporário B_i (*Buffer*). O armazenamento temporário pode armazenar uma quantidade pré-determinada de *chunks*. Cada posição j de B_i, ou seja $B_i[j]$ é inicializada com conteúdo vazio e elas vão sendo preenchidas à medida que p_i recebe os *chunks* de seus parceiros. O objetivo é manter o *buffer* B_i preenchido de forma que se garanta uma contínua exibição da mídia ao vivo.

O buffer B_i também representa uma janela deslizante onde a posição mais recente é $B_i[s]$ e a menos recente é $B_i[0]$, onde s é a última posição de um buffer com $s+1$ posições. A cada intervalo de tempo, o participante p_i desloca sua janela de *buffer*, descartando o conteúdo mais antigo e acrescentando uma nova posição como "frente" do *buffer*. Por motivos de simplificação, considera-se que o *chunk* a ser consumido para exibição do vídeo ao vivo será o *chunk* na posição $B_i[0]$ (*chunk* mais antigo). A figura 5.1 ilustra o processo descrito acima. Nessa figura estão representados três intervalos de tempo consecutivos em um buffer de 10 posições.

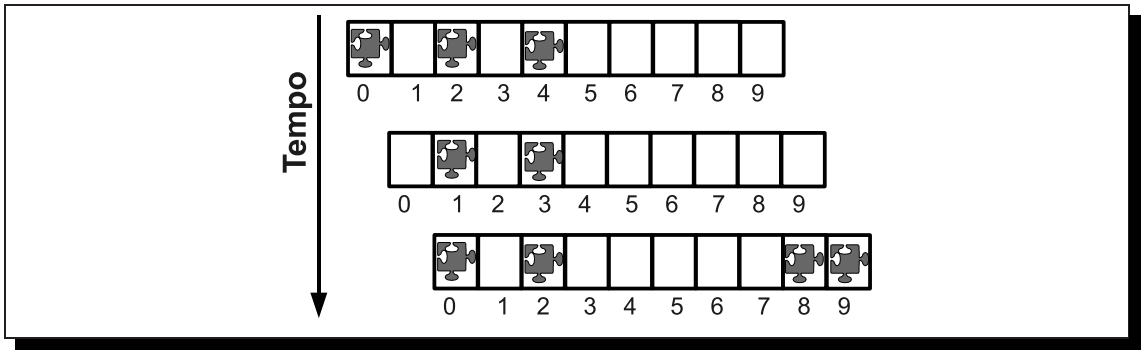


Figura 5.1: Mecanismo de janela deslizante dos participantes da rede P2P.

Para simplificação do modelo apresentado, sem perda da sua capacidade de expressão, consideram-se todos os participantes de forma homogênea e regular. Em outras palavras, todos os participantes possuem a mesma capacidade de

armazenamento, e estão com a mesma janela de interesse em suas respectivas áreas de armazenamento temporário.

Dessa forma, a probabilidade de um determinado participante obter um *chunk* recém criado depende das parcerias que o mesmo possui. Por exemplo, caso esse participante seja parceiro do servidor, a probabilidade de ele conseguir esse *chunk* recém criado é maior que a de outros participantes, que não sejam parceiros do servidor.

5.2 Estratégias de Seleção de *Chunks*

A estratégia de seleção do *chunk* a ser requerido da rede pode influenciar o desempenho e a eficiência da aplicação de mídia contínua ao vivo em P2P. Uma estratégia tenta estabelecer qual dos *chunks*, entre os vários necessários, deve ser requisitado em um determinado momento. As estratégias de seleção de *chunks* geralmente tentam manter a continuidade da exibição da mídia ao vivo para o participante do sistema e também, difundir o mais rápido possível um trecho de mídia que acaba de ser gerado.

As duas estratégias de seleção de *chunks* comumente utilizadas em aplicações P2P são “Mais Raro Primeiro” e “Gulosa”. A estratégia "Mais Raro Primeiro"é adotada em protocolos de aplicações de compartilhamento de arquivos em P2P como o Bittorrent [5] e em envio de mídia ao vivo em P2P como o CoolStreaming [99]. Na estratégia "Gulosa", os participantes privilegiam a escolha de *chunks* que estão próximos ao fim de suas janelas de visualização.

Ao se utilizar a estratégia "Mais Raro Primeiro", os participantes selecionam os *chunks* que estão menos replicados no sistema. Esta estratégia é utilizada para tentar melhorar a distribuição dos dados no sistema [109]. Por sua vez, na estratégia "Gulosa", um participante p_i irá selecionar o *chunk* mais próximo ao ponto de prazo final de visualização. Nessa estratégia é feia uma tentativa de manter a exibição contínua da mídia sem interrupções do vídeo.

5.3 Impacto Gerado pela Disseminação de Conteúdo Poluído

Um sistema de transmissão ao vivo em P2P, sob ataque de disseminação de conteúdo poluído, apresenta m participantes, dos quais b são participantes maliciosos, denominados poluidores ($b \leq m$). Cada participante não poluidor p_i apresenta n_i parceiros. Desses n_i parceiros de p_i, existem b_i poluidores (considera-se: $0 \leq b_i < n_i$).

Quando um participante p_i captura um *chunk* de número h poluído, esse *chunk* é descartado, e a posição $B_i[h]$ continua vazia. Este *chunk* continua sendo necessário ao participante e, enquanto estiver figurando na janela de interesse, p_i irá fazer novas requisições por este dado. A tabela 5.1 apresenta um resumo dos parâmetros utilizados no modelo em questão.

Tabela 5.1: Resumo dos elementos do modelo.

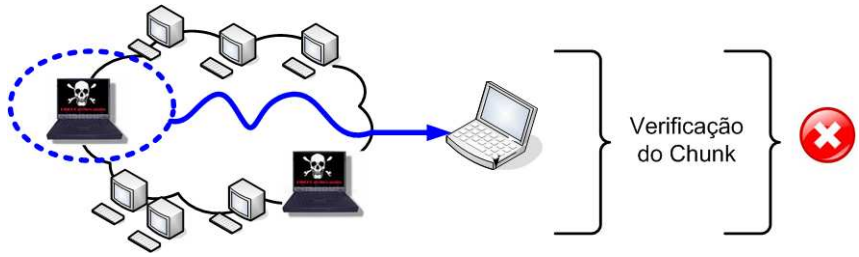
Parâmetro	Descrição
m	número de participantes do sistema
p_i	participante i do sistema
n_i	número médio de parceiros de p_i
B_i	*Buffer* de p_i
$B_i[j]$	posição j de B_i
b	número de participantes maliciosos
b_i	parceiros poluidores de p_i $(0 \leq b_i < n_i)$

Em um modelo simplificado, o processo de se obter um *chunk* válido se aproxima ao problema de se obter boas parcerias. Caso um participante p_i possua somente parceiros bons, todas as tentativas de recuperar um *chunk* retornarão um dado válido. Da forma oposta, se as parcerias de p_i são maliciosas, p_i obterá dados poluídos.

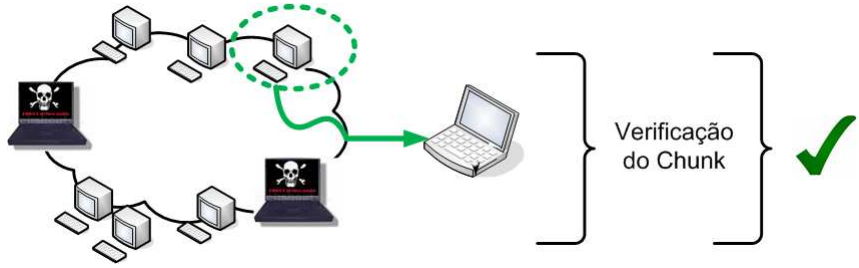
A figura 5.2 ilustra um participante p_i tentando obter um determinado *chunk* h. Nesse cenário, qualquer um de seus parceiros pode atender essa requisição, porém, p_i só terá sucesso se o dado retornado for correto. No primeiro instante, a figura 5.2a mostra p_i requisitando dados para um de seus parceiros. Nesse caso, o parceiro é um poluidor do sistema e retorna uma informação inválida para p_i. Ao receber o dado p_i verifica a integridade do *chunk* e o descarta por ser um dado poluído. Assim, na figura 5.2b p_i faz uma nova requisição pelo *chunk* h aos demais parceiros. Nessa tentativa, p_i escolhe um parceiro honesto e recebe as informações corretas.

De forma mais direta, a figura 5.3 ilustra o processo realizado por um participante até conseguir um determinado *chunk* válido. Seja um sistema homogêneo, onde os participantes possuem os mesmos recursos, e que os parceiros de um participante p_i possuam dados para compartilhar e atender as requisições de p_i. Além disso, o número de participantes do sistema e o número de parceiros de p_i não se alteram. Nesse caso, quando p_i seleciona um *chunk* h para realizar requisição, ele pode escolher qualquer um de seus parceiros.

A partir do estado inicial S, caso p_i escolha um bom parceiro, ele terá um acerto e conseguirá um *chunk* válido. Caso contrário, p_i descarta o poluidor que o atendeu



(a) Primeira tentativa de se obter um dado - insucesso



(b) Segunda tentativa de se obter um dado - sucesso

Figura 5.2: Descarga de um *chunk* da rede - abordagem "marcar e checar".

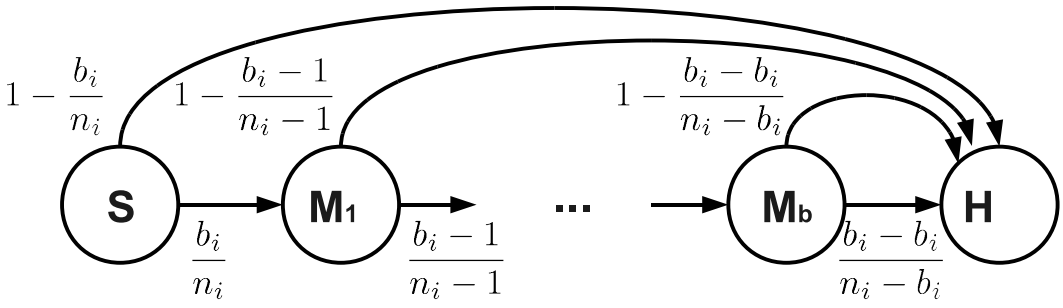


Figura 5.3: Encadeamento de requisições até um sucesso.

e refaz uma escolha aleatória de um parceiro para requisitar novamente o *chunk* em questão. Esse processo é repetido até que p_i obtenha um *chunk* válido, ou até que o *chunk* saia da janela de tempo para visualização da mídia. Nesse trabalho, considera-se que a janela de interesse para visualização é menor que o número máximo de tentativas que p_i irá fazer até conseguir um bom parceiro.

Uma vez que p_i tem n_i parceiros, dos quais b_i são poluidores, a probabilidade de se obter um bom parceiro, logo na primeira escolha, é dada por $1 - (b_i/n_i)$. Caso o

participante não obtenha sucesso em sua primeira tentativa, ele descarta o poluidor anterior, e assim, ele terá maior probabilidade de sucesso em sua segunda tentativa. Essa probabilidade é dada por $1 - \left({}^{b_i - 1}/_{n_i - 1}\right)$. Caso ele tenha um novo insucesso, na terceira tentativa a probabilidade de sucesso será $1 - \left({}^{b_i - 2}/_{n_i - 2}\right)$. No pior caso, esse processo pode ser repetido até se esgotarem todos os poluidores, e assim, a probabilidade de sucesso é dada por $1 - \left({}^{b_i - b_i}/_{n_i - b_i}\right)$; ou seja, 1.

A carga média exigida para as transmissões pode ser analisada como o caminho médio no percurso da figura 5.3 até se obter um acerto (escolha de um parceiro honesto). Cada passo realizado, nessa figura, representa um pedido por transmissão de um *chunk*, sua recepção e verificação de integridade.

No mínimo, é necessário um caminho de tamanho 1, onde 1 representa a taxa de mídia em um determinado tempo. No máximo, serão exigidas $b_i + 1$ unidades de taxa de mídia, pois, o participante deverá requisitar para todos os poluidores de sua parceria, e mais uma vez para um parceiro honesto.

A equação 5.1 apresenta o caminho médio até o sucesso esperado para um sistema homogêneo e com iguais probabilidades de distribuição de dados entre todos os participantes. No modelo apresentado, o tamanho de um caminho médio é influenciado somente pela proporção entre poluidores e nodos honestos.

$$
\begin{aligned}
l &= \left[1 - \left(\frac{b_i}{n_i}\right)\right] * 1 \\
&+ \left[1 - \left(\frac{b_i - 1}{n_i - 1}\right)\right] * \frac{b_i}{n_i} * 2 \\
&+ \left[1 - \left(\frac{b_i - 2}{n_i - 2}\right)\right] * \frac{b_i}{n_i} * \frac{b_i - 1}{n_i - 1} * 3 \\
&+ \left[1 - \left(\frac{b_i - 2}{n_i - 2}\right)\right] * \frac{b_i}{n_i} * \frac{b_i - 1}{n_i - 1} * \frac{b_i - 2}{n_i - 2} * 4 \\
&+ \left[1 - \left(\frac{b_i - b_i}{n_i - n_i}\right)\right] * \frac{b_i}{n_i} * \frac{b_i - 1}{n_i - 1} * ..... * \frac{b_i - (b_i - 1)}{n_i - (b_i - 1)} * (b_i + 1)
\end{aligned}
$$

$$
l = 1 - \left(\frac{b_i}{n_i}\right) + \sum_{s=2}^{b_i+1} 1 - \left(\frac{b_i - (s-1)}{n_i - (s-1)}\right) * s * \prod_{j=0}^{s-2} \frac{b_i - j}{n_i - j} \tag{5.1}
$$

Esse modelo básico do impacto gerado pela distribuição de conteúdo poluído será refinado em dois casos específicos. O primeiro caso apresenta um cenário otimista, onde os impactos dos ataques são os menores possíveis. O segundo cenário apresenta um cenário pessimista, e os impactos de um ataque de poluição são os maiores possíveis.

5.3.1 Estratégia "Gulosa" - Cenário Otimista

A estratégia *"Gulosa"* tenta capturar da rede os *chunks* que estão mais próximos de seu prazo final de visualização. Nessa estratégia de seleção de *chunks*, um participante p_i irá selecionar o mais próximo à posição de visualização em seu *buffer* B_i.

Nesse caso, o *chunk* que deve ser requisitado já foi produzido pelo servidor, e muitos participantes provavelmente já o têm. Isso ocorre porque os participantes apresentam atrasos distintos entre o ponto de visualização da mídia em relação ao trecho que está sendo produzido. Assim, quanto mais tardiamente um *chunk* for requisitado da rede, maiores as chances de outros participantes o ter.

Em um cenário otimista, todos os parceiros de um determinado participante podem responder a uma requisição por *chunk*, na estratégia gulosa. Dessa forma, o participante não fica na dependência de ter que requisitar dados aos poluidores (que sempre anunciam todos os dados). Caso o participante escolha aleatoriamente um de seus parceiros para realizar a requisição, as probabilidades de escolher um parceiro honesto ou poluidor são proporcionais às quantidades destes.

Portanto, o impacto gerado por um ataque, quando se utiliza a estratégia gulosa, é a mesma apresentado, anteriormente, na figura 5.3. Porém, em uma situação mais realista, nem todos os parceiros honestos de um determinado participante podem atender a uma requisição. Entre os parceiros, há uma determinada quantidade que podem servir enquanto os demais esperam por serviço. Mesmo que um *chunk* já tenha sido difundido na rede amplamente, espera-se que nem todos os parceiros o tenham.

Nesse cenário mais realista, o número de parceiros que pode atender a um pedido do participante se reduz. Assim, o modelo anterior terá o número de participantes honestos alterado para um número menor, $y * n_i - b_i$; onde y é a proporção de parceiros que podem servir o *chunk*. Mais precisamente, o impacto para um cenário onde um *chunk* já foi amplamente divulgado e a estratégia de seleção utilizada é a gulosa é dada pela equação 5.2.

$$l = 1 - \left(\frac{b_i}{w}\right) + \sum_{s=2}^{b_i+1} 1 - \left(\frac{b_i - (s-1)}{w - (s-1)}\right) * s * \prod_{j=0}^{s-2} \frac{b_i - j}{w - j} \tag{5.2}$$

onde $w = y * n_i - b_i$; y é a proporção de parceiros com o dado

5.3.2 Estratégia *"Mais Raro Primeiro"* - Cenário Pessimista

A estratégia "Mais Raro Primeiro" seleciona os *chunks* menos difundidos para requisição. Em termos gerais, os *chunks* mais recentemente produzidos são os mais raros. Assim, essa estratégia tenta preencher os dados mais recentes do *buffer*, tentando

aumentar a disponibilidade dos *chunks* de vídeo na rede rapidamente e apresentar um vídeo com baixa latência.

Inicialmente, um *chunk* recém criado só se encontra em poder do servidor. Os poluidores também respondem por pedidos por *chunks* raros, uma vez que eles podem forjar qualquer dado ou informação. Dessa maneira, um poluidor sempre vai oferecer respostas por um pedido por *chunk*. Caso um participante procure pelo *chunk* mais recentemente criado, somente o servidor poderá atendê-lo de forma correta.

Assim, caso um participante seja parceiro do servidor, poderá ter o *chunk* corretamente recebido. Porém, a probabilidade de um participante ser parceiro do servidor é dada por n/m, onde n é o número médio de parcerias que um participante realiza (incluindo o servidor) e m é o número de participantes do sistema. Nessa mesma situação, o participante p_i pode apresentar poluidores entre seus parceiros. A probabilidade de um participante ter poluidores entre seus parceiros é dada por $n*b/m$, onde b é o número total de poluidores do sistema (considera-se o sistema homogêneo e com distribuição dos poluidores entre os participantes com igual probabilidade de escolha em uma parceria). Dessa forma, a probabilidade de um poluidor responder por um pedido de procura por um dado raro é maior que a probabilidade de que o servidor responda por este pedido, quando $B > 1$.

O *chunk* mais raro pode se tornar menos raro com o passar do tempo. Por exemplo, o *chunk* mais raro no tempo $t - 1$, passa a ser o segundo *chunk* mais raro no tempo t. Isso porque o servidor já difundiu esse *chunk* para seus vizinhos. Nesse momento, podem existir p participantes no sistema com o *chunk* que, anteriormente, era o mais raro. Esses p participantes podem repassar o conteúdo para seus parceiros, e assim, o número de réplicas desse *chunk* aumenta.

De maneira simples, um participante p_i não terá oportunidade de obter um *chunk* qualquer, até que esta difusão tenha alcançado algum de seus parceiros. Pode-se fazer uma analogia da organização da rede como um conjunto de círculos concêntricos, onde o servidor encontra-se no centro e, a cada intervalo de tempo, um *chunk* é difundido para as camadas exteriores. Os participantes de uma camada mais externa possuem ligação com algum dos participantes da camada imediatamente mais interna. Note que, para a difusão de um *chunk*, um participante pode estar em uma camada mais interna, e na difusão do *chunk* seguinte, ele pode aparecer em uma camada mais externa.

A figura 5.4 ilustra o processo de difusão de um *chunk* a partir do servidor. No ponto 0, ilustrado na figura 5.4-a, somente o servidor possui esse *chunk*, uma vez que o produziu naquele instante. No momento seguinte, alguns dos participantes, diretamente ligados ao servidor, podem obter esse *chunk*. Assim, o *chunk* encontra-se à distância 1 do servidor, como mostrado na figura 5.4-b. No terceiro momento, o

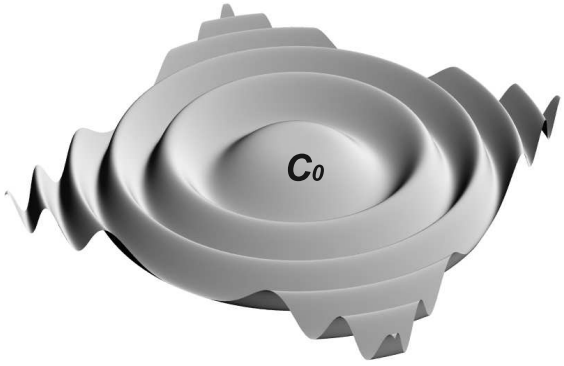
chunk pode ter sido difundido a outros participantes que, anteriormente não tinham nenhuma ligação com o servidor. Na figura 5.4-c, observa-se que o *chunk* foi obtido a partir dos participantes da linha 1 da onda de difusão, e agora, passaram a fazer parte do conjunto de dados de outros participantes. Esse processo continua, até que todos os interessados tenham obtido esse *chunk*, ou até que o tempo de vida do *chunk* (tempo de interesse) tenha ultrapassado um limite tolerável.

Nota-se, claramente, que impacto gerado por um ataque na estratégia "mais raro primeiro" difere do impacto apresentado anteriormente na figura 5.3. A principal diferença ocorre no momento em que os participantes não têm parceiros honestos com o *chunk* procurado. Os participantes ficam um período de h saltos de dados, até que um de seus parceiros honestos possa ter o dado disponível para compartilhamento, e assim, o atender de forma correta.

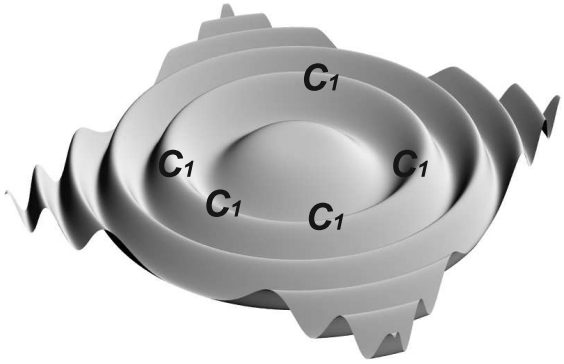
Nesse cenário, o participante se sujeita a capturar dados de poluidores durante um período de h saltos. O modelo anterior então apresentará uma sobrecarga inicial relativa ao período de influência exclusiva dos poluidores. Assim, o impacto causado em um cenário onde um *chunk* ainda não foi amplamente difundido e a estratégia de seleção utilizada é a "mais raro primeiro", é dado pela equação 5.3.

$$l = h + 1 - \left(\frac{b_i}{w}\right) + \sum_{s=2}^{b_i+1} 1 - \left(\frac{b_i - (s-1)}{w - (s-1)}\right) * s * \prod_{j=0}^{s-2} \frac{b_i - j}{w - j} \tag{5.3}$$

onde h é a distância média entre servidor/participante e $w = y * n_i - b_i$



(a) O servidor acaba de produzir o *chunk*



(b) O *chunk* é difundido para a primeira linha

(c) O *chunk* alcança a segunda linha de participantes

Figura 5.4: Exemplo de difusão de um *chunk* usando "mais raro primeiro".

5.4 Análise dos Resultados

Os modelos criados anteriormente foram formalizados e analisados em uma ferramenta específica denominada PRISM [68]. O PRISM é uma ferramenta de checagem de modelos probabilísticos, onde é possível modelar e analisar sistemas que exibem comportamento aleatório ou probabilístico [36, 45, 68]. Essa ferramenta suporta 3 tipos de modelos *Markovianos*, sendo eles: cadeias discretas, contínuas, e processos de decisão. Além desses 3 modelos básicos, também são suportadas extensões deles, com adição de custos e recompensas. Um dos fatores que determinaram o uso da ferramenta foi a sua ampla utilização para analisar sistemas de comunicação, protocolos multimídia e protocolos de segurança.

Os parâmetros dos modelos, como número de parceiros, distância média entre os participantes, foram obtidos a partir da caracterização realizada no capítulo 4. Por exemplo, o número de parceiros utilizado nos resultados abaixo foi obtido a partir da caracterização dos usuários do SopCast. A distância média entre os participantes também foi obtida a partir dos experimentos realizados a caracterização do SopCast.

Por simplificação, serão adotados os valores médios para a resolução dos modelos propostos. Apesar disso, a ferramenta utilizada (o PRISM) aceita distribuições de probabilidades dos parâmetros adotados. A tabela 5.2 resume os parâmetros utilizados para determinar os valores de sobrecarga de poluição a partir dos modelos apresentados.

Tabela 5.2: Parâmetros para avaliação dos modelos propostos.

Parâmetro	Descrição	Valor
m	número de participantes do sistema	1000
n_i	número médio de parceiros	100
b_i	número médio de parceiros poluidores	1; 10
y	Proporção de parceiros que podem servir	0.5 ; 1
h	Distância média entre os participantes	1.677

No caso de um participante ter 100 parceiros em média e 1 poluidor entre eles (1% de poluidores na rede), o modelo proposto para um cenário otimista apresenta uma sobrecarga de aproximadamente 2%. A proporção entre poluidores e não poluidores é bastante baixa (1 : 100), mesmo que somente 50% dos parceiros possam servir os dados, em um determinado momento essa proporção continua baixa (1 : 50). Essa proporção torna pouco provável a escolha de um poluidor como candidato a servir o dado requisitado, levando a uma baixa sobrecarga.

O resultado encontrado para o cenário pessimista apresenta uma sobrecarga de

aproximadamente 169%. Em outras palavras, espera-se que os participantes do sistema sejam obrigados a pedir retransmissão de 1.69 *chunk* para cada um requisitado. Assim, os participantes do sistema podem precisar de praticamente o triplo de banda necessária para a recepção de 1 mídia. Mesmo que o número de parceiros com capacidade de compartilhar a mídia aumente, o cenário pessimista continua apresentando um mau desempenho com relação a sobrecarga. Se todos os participantes puderem colaborar com o seu parceiro ($y = 1$), a sobrecarga terá um valor de aproximadamente 167%. Ou seja, continua sendo um valor alto.

Quando o número de poluidores aumenta, os resultados encontrados mostram um cenário ainda pior. Mesmo no modelo com cenário otimista, a sobrecarga é aproximadamente 32,2%, quando se tem 10 parceiros poluidores. Para o modelo em cenário pessimista, esse número sobe para aproximadamente de 180%. Novamente, mesmo que o número de parceiros úteis seja de 100%, a sobrecarga continua alta, superando 177%.

A tabela 5.3 exibe os resultados encontrados para os modelos em cenário otimista e pessimista. Na tabela abaixo, além dos resultados com a carga média da rede necessária para a recepção do conteúdo ao vivo, são exibidos o número de estados e de transições do modelo em questão.

Tabela 5.3: Resultado da avaliação dos mdelos - 100 parceiros em média.

Cenário	# de Poluidores	Estados	Transições	Carga Média da Rede
Otimista	1	5	6	1.0204081632653061
Pessimista ($y = 0.5$)	1	5	6	2.677 | 2,69040816
Pessimista ($y = 1$)	1	5	6	2.677 | 2,677
Otimista	10	23	33	1.3225806074103288
Pessimista ($y = 0.5$)	10	23	33	2,992580607
Pessimista ($y = 1$)	10	23	33	2.776890102

Quando o número de parceiros de um participante diminui, os resultados para os cenários otimista ou pessimista também se alteram. A tabela 5.4 resume os resultados encontrados para os modelos propostos com um número menor de parceiros; mantida a proporção de poluidores da rede.

O modelo em cenário otimista apresenta uma sobrecarga maior que a situação anterior com 100 parceiros. Porém, a sobrecarga ainda pode ser considerada pouco impactante, com cerca de 4% de banda de rede adicional necessária para a transmissão ao vivo. Para um cenário pessimista, caso 50% dos parceiros possam colaborar com um

participante, o resultado encontrado mostra uma situação de alta sobrecarga. Mesmo com 1 único poluidor, um participante está sujeito a cerca de 171% de sobrecarga. Mesmo que a quantidade de parceiros úteis seja de 100%, no cenário pessimista um participante estará sujeito a mais de 168% de sobrecarga.

Quando o número de poluidores aumenta, o cenário otimista terá a sua proporção de poluidores/participantes bastante alterada em relação às situações anteriores. Por esse motivo, a sobrecarga no cenário otimista se eleva, chegando a cerca de 166%. No cenário pessimista, sob as mesmas condições, essa sobrecarga ultrapassa 333%. Porém, se a proporção de parceiros úteis for alterada no cenário pessimista, o resultado encontrado para ele não se distancia tanto do resultado encontrado no cenário otimista. Nesse caso, o valor encontrado para o cenário pessimista é cerca de 191%.

Esse resultado mostra que, além da influência notória do algoritmo de seleção de *chunks*, uma melhor relação de parcerias pode influenciar na sobrecarga imposta durante um ataque. Parceiros que são altruístas podem alterar a relação entre atacantes e participantes úteis do sistema, e assim, diminuir a influência dos poluidores.

Tabela 5.4: Avaliação do modelo - 50 parceiros em média.

Cenário	# de Poluidores	Estados	Transições	Carga Média da Rede
Otimista	1	5	6	1.0416666666666667
Pessimista ($y = 0.5$)	1	5	6	2,711666667
Pessimista ($y = 1$)	1	5	6	2.687
Otimista	10	23	33	2.6666666666666665
Pessimista ($y = 0.5$)	10	23	33	4,336666667
Pessimista ($y = 1$)	10	23	33	2.910902351

5.5 Resumo do Capítulo

Neste capítulo foi descrito um modelo estocástico simples, que pode ser utilizado para comparar o impacto de ataques de poluição em diferentes estratégias de busca de dados em redes P2P de mídia ao vivo. Essa comparação foi baseada, principalmente, em uma métrica: sobrecarga de rede (quanto a mais de dados é necessário para a visualização da mídia devido aos dados poluídos).

Foram analisadas as duas estratégias mais comuns de seleção de dados para busca na rede P2P. Uma das estratégias tenta distribuir mais rapidamente os *chunks* recém criados, e a outra, tenta evitar perdas na exibição do conteúdo. Essas estratégias

influenciam no resultado de um ataque de poluição e, dependendo da estratégia escolhida, a sobrecarga imposta à rede pode ser maior ou menor.

No caso de um cenário otimista, todos os parceiros podem atender aos pedidos realizados por um determinado participante. Esse cenário é apropriado ao uso da estratégia de seleção de *chunks* gulosa, onde se escolhe o *chunk* mais próximo de seu deadline, por consequência, mais antigo na rede (e mais difundido). Porém, essa estratégia de seleção só é viável em situações em que a latência da rede não é importante. Para os cenários mais comuns, onde a relação entre poluidores e parceiros é baixa, a sobrecarga para esse cenário ficou abaixo de 5%. Porém, quando essa relação se altera, essa sobrecarga pode chegar a mais de 170%.

No cenário pessimista (estratégia de seleção de *chunks* raros) a sobrecarga imposta por ataques é alta. Na maior parte dos casos, os participantes devem ter pelo menos 3 vezes a quantidade de banda de rede necessária para a transmissão.

Finalmente, além da influência dos algoritmos de seleção de *chunks*, uma melhor relação de parcerias pode influenciar na sobrecarga imposta durante um ataque. Quanto melhor a relação entre parceiros úteis/poluidores, menores são os danos causados por um ataque. Esse resultado evidencia a importância de algoritmos de seleção de parceiros e de isolamento de participantes poluidores.

Capítulo 6

Sistema de Transmissão ao Vivo em P2P Sob Ataque de Poluição

Neste capítulo serão apresentas simulações que demonstram os efeitos de ataques de poluição de conteúdo em sistemas de transmissão ao vivo em P2P. São abordados dois cenários distintos nessas simulações. O primeiro deles apresenta um sistema de transmissão ao vivo sem nenhum mecanismo de verificação do conteúdo transmitido. Neste cenário, um ataque com apenas 1 poluidor em uma rede com 1000 participantes pode ser devastador. Os resultados apresentados mostram que, em alguns casos, todos os dados recebidos por parte da rede de transmissão ao vivo estão alterados pelo único poluidor. No segundo cenário apresentado, os participantes verificam a integridade dos *chunks* recebidos. Caso um desses esteja danificado, há uma nova tentativa para se obter esse dado do sistema. Neste caso, apesar dos participantes do sistema não consumirem trechos de vídeo inválidos, eles necessitam de uma banda de rede adicional para realizar as retransmissões necessárias. Em alguns casos, a sobrecarga é superior a 100% da banda de rede original.

Na literatura, existe uma série de trabalhos que tratam, principalmente, os comportamentos oportunistas dos usuários de um sistema de transmissão ao vivo em P2P. Trabalhos como [13,40,85] apresentam propostas para reputar os participantes do sistema, banindo os participantes que contribuem pouco. Nesse contexto, um sistema de reputação consiste em um sistema capaz de avaliar a atuação de um participante na rede P2P e, através de seu comportamento passado, atribuir-lhe uma nota. Por exemplo, os participantes com baixa relação *upload/download* em suas interações passadas serão avaliados de forma punitiva. Por consequência, terão notas ruins de reputação, e assim, terão menor prioridade no sistema em relação aos participantes que possuem maior relação *upload/download*. Dessa forma, esses trabalhos apresentam

propostas que podem incentivar um comportamento altruísta dos participantes da rede P2P de transmissão ao vivo.

Com relação ao conteúdo poluído, há uma série de trabalhos desenvolvidos no contexto de compartilhamento de arquivos em P2P. Sistemas como Credence [91] apresentam uma abordagem distribuída, na qual os participantes da rede assinalam reputação aos *objetos* descarregados do sistema (arquivos baixados). Além de uma nota que descreve se um usuário considerou um objeto como poluído ou não, há uma reputação relativa aos usuários participantes do sistema. Por exemplo, a proposta de reputação em sistema de compartilhamento de arquivos denominada Scruber [15], consegue identificar por meio de reputação os participantes maliciosos que, ativamente, disseminam conteúdo poluído. Um efeito colateral da adoção de tais medidas é uma diminuição da propagação de conteúdo poluído de forma ingênua. Caso algum participante, mesmo que não seja poluidor, dissemine conteúdo poluído, terá sua reputação penalizada. Assim, os participantes honestos do sistema são incentivados a verificar o conteúdo que eles têm e descartar o quanto antes os arquivos poluídos.

Entre os primeiros trabalhos que tratam poluição de conteúdo em sistemas de transmissão ao vivo em P2P estão [28–31]. Nesses trabalhos é apresentada uma proposta de um sistema de transmissão ao vivo em P2P resistente à alteração de conteúdo e intrusão de participantes maliciosos. Também são apresentadas comparações entre quatro alternativas para verificar a integridade do vídeo distribuído no sistema de mídia contínua ao vivo em P2P. Nesse sentido, os autores mostram que é possível realizar a verificação do conteúdo recebido da rede, e assim, evitar a execução de trechos de mídia poluída.

Em [20] é apresentado um experimento, no qual um poluidor ativo é colocado em um sistema real. Os resultados obtidos indicam que ataques de poluição podem danificar por completo um sistema ao vivo de vídeo em P2P. Por exemplo, em um dos experimentos, o sistema de transmissão ao vivo teve uma forte evasão dos participantes que estavam insatisfeitos com o serviço oferecido. O número de participantes caiu para cerca de 10% da quantidade de participantes existentes antes ao ataque (de 1000 para cerca de 100 participantes). Nesse mesmo trabalho, são sugeridas algumas abordagens para evitar o consumo de dados corrompidos. Os autores mostram que é possível verificar a integridade dos dados da mídia transmitida com um baixo custo adicional. Utilizando os esquemas propostos por eles, há um custo de processamento em $O(n)$ para assinalar os blocos de dados a serem transmitidos, onde n é a quantidade de *chunks* contida em cada bloco. A verificação a ser realizada pelo participante é de $O(1)$. O custo adicional de banda de rede é cerca de 5% da banda de rede necessária para o envio do fluxo original.

Apesar das propostas existentes para evitar conteúdo poluído em um sistema de transmissão ao vivo em P2P, não são apresentados evidências de que o assinalamento dos dados transmitidos seja uma medida eficiente para o combate aos poluidores. Além disso, o experimento realizado para verificar a abrangência de um ataque aos sistemas de transmissão ao vivo em P2P não apresenta uma visão generalizada da aplicação. São mostrados apenas alguns dados pontuais, de alguns poucos participantes.

6.1 Modelo de Simulação

Nesta seção serão abordados os principais componentes do simulador e o modelo adotado para simulação. Serão descritos a topologia de rede utilizada, a estrutura da aplicação P2P, o tamanho da rede simulada e o comportamento dos participantes.

6.1.1 Topologia da Rede Simulada

A topologia da rede selecionada, geralmente, influencia no resultado da simulação. Assim, topologias realistas são necessárias para produzir resultados que refletem a realidade. As simulações realizadas baseiam-se em duas topologias: uma topologia física, e uma topologia lógica. A topologia física deve representar uma topologia real com as características da Internet. Por sua vez, a topologia lógica representa a rede P2P sobreposta à topologia física.

Dessa forma, os participantes do sistema P2P são um subconjunto dos nodos da topologia física existente. O custo da comunicação entre dois parceiros na rede sobreposta pode ser calculado, baseado no caminho mínimo entre estes participantes na topologia física de rede. Apesar desse custo poder influenciar no atraso percebido pelos participantes do sistema de transmissão ao vivo em P2P, a maior parte das aplicações existentes não a utilizam como critério de formação de parcerias.

Trabalhos anteriores mostram que ambas, topologia física [88] e a topologia da rede P2P sobreposta [75], seguem as propriedades de *"small world"* e leis de potência [3] (*power law*). As leis de potência descrevem o grau de conexão dos nodos, enquanto as propriedades de *"small world"* descrevem as características de caminho (*path lenght*) e agrupamento dos nodos (*clustering coefficient*) [93].

Mais detalhadamente, uma rede *"small world"* é um tipo de grafo, no qual a maior parte dos nodos não estão conectados entre si, porém, a maioria pode ser acessada de qualquer outro por um pequeno número de saltos [3, 93]. Muitos grafos e redes são bem modelados por redes do tipo *"small-world networks"*. Por exemplo, várias redes sociais, a conectividade da Internet e até na conexão dos neurônios de vermes [93].

Em redes que seguem às leis de potência, uma fração dos nodos tem muitas conexões, enquanto outros têm poucas [3]. Redes que seguem às leis de potência também são chamadas de livres de escala (*scale-free*). Uma das principais características desse tipo de rede é a tolerância a falhas aleatórias. Por exemplo, remoção aleatória de nodos desse tipo de rede não destrói por completo a rede.

A construção da topologia física a ser utilizada nas simulações pode refletir a Internet, sob o ponto de vista de sistemas autônomos (AS), ou a estrutura de roteadores da rede. Sob o ponto de vista de AS, representam-se vários computadores, roteadores, e redes que estão sob a mesma administração, como uma única entidade (um único nodo). Enquanto isso, ao se representar a estrutura de roteadores da rede, cada roteador é um nodo, mesmo que eles pertençam à mesma rede ou domínio administrativo.

Apesar desses dois tipos de topologias serem relacionadas, afetam a conectividade dos nodos da Internet em diferentes escalas. Por exemplo, a topologia de AS abstrai muitos detalhes da conectividade física entre os ASs, e cada AS representa um grupo de vários roteadores topologicamente contínuos. Porém, como mostrado em [87], os dois tipos de topologia geradas apresentam similaridades com respeito às métricas empregadas como grau médio e *ranking* dos nodos de rede.

Neste trabalho, a topologia da rede física foi gerada utilizando o programa BRITE [59]. O BRITE foi concebido para gerar topologias Internet constituídas por sistemas autônomos (AS) e roteadores. Ele gera a topologia da rede em 4 passos:

1. Disposição dos nodos na rede;
2. Interconexão dos nodos;
3. Geração dos atributos dos componentes da topologia. São gerados atraso e banda de rede nas ligações entre nodos; identificação do AS dos nodos; etc;
4. Geração da topologia para o formato final específico da ferramenta de simulação.

Foi utilizado o modelo de probabilidade de Waxman [94, 95] para interconectar os nodos na topologia física de rede. O modelo de Waxwan é dado por:

$$P(u, v) = \alpha * e^{-d/(\beta * L)} \tag{6.1}$$

Onde $0 < \alpha, \beta \leq 1$; d é a distância Euclidiana do nodo u para o nodo v; L é a distância máxima entre dois nodos.

Os valores de parâmetros para o BRITE, sugeridos em [32], podem ser utilizados para a geração de uma topologia baseada em um ISP (provedores de serviço Internet). Nesse trabalho, os autores avaliam ferramentas de geração de topologia e verificam a similaridade com 2 provedores de serviço de Internet reais. A espinha dorsal da rede AT&T dos Estados Unidos e a estrutura da rede DFN G-Win (rede de pesquisa alemã) foram utilizadas como base para as comparações neste trabalho. Os valores de parâmetros sugeridos são resumidos na tabela 6.1.

Tabela 6.1: Waxman - parâmetros de AT&T e DFN G-Win.

Topologia real	Tipo de Construção	Método	α	β	Grau
DFN	Bottom up	Aleatório[1], GLP[2]	0.42-0.46	0.62-0.68	3
AT&T	Bottom up	Aleatório, GLP ou BA[3]	Irrelevante	Irrelevante	2

Os parâmetros sugeridos em [87] para o modelo de Waxman, incluindo o número de nodos, valores de α e β, podem ser vistos na tabela abaixo:

Tabela 6.2: Waxman - parâmetros para topologia Internet.

Grau Médio	Tamanho da Topologia	*alpha*	β
5.06	1000	0.050	0.20
2.03	5000	0.005	0.05
2.82	5000	0.005	0.10
7.22	5000	0.005	0.30
10.82	5000	0.005	0.50
2.79	5000	0.010	0.05
5.03	5000	0.010	0.10
14.42	5000	0.010	0.30

Com base nos parâmetros de partida supracitados, o BRITE foi utilizado para gerar topologias de rede aleatória com o modelo de Waxman.

As topologias geradas contêm 10000 nodos com grau médio normalmente distribuído entre 2 e 3. Os valores utilizados de α e β foram variados entre os valores indicados para a rede DFN G-Win na tabela 6.1. A cada simulação, uma das topologias criadas é escolhida, aleatoriamente, para ser utilizada. A tabela 6.3 resume os parâmetros da topolgia criada para as simulações.

Tabela 6.3: Topologias de rede geradas por Waxman.

Número de nodos	α	β	Grau
10000	$0.42 - 0.46$	$0.62 - 0.68$	$2 - 3$

6.1.2 Topologia Lógica

O simulador de rede NS-2 [58] foi utilizado para gerar a rede sobreposta constituída de 1000 participantes. Essa rede sobreposta é criada sobre a topologia física, criada pelo BRITE, descrito anteriormente. No NS-2 foi construído um conjunto de novos agentes, que simulam todas as entidades participantes de um sistema de transmissão ao vivo em P2P. Esses agentes seguem um protocolo orientado por pedidos a dados, e estruturam os participantes baseando-se em malha (*mesh-pull overlay network*), da mesma forma que sistemas populares como PPLive, PPStream e GridMedia [33, 66, 67, 103].

As principais características dos nodos da topologia lógica seguem o modelo proposto no capítulo 4. Entre essas características, destacam-se o grau dos nodos (número de parceiros de um participante do sistema), o tempo de atividade do nodo e o tempo de uma dada parceria. As características específicas de cada cenário de simulação serão decritas, quando necessário, junto à simulação em questão.

6.1.3 Modelo de Transmissão ao Vivo em P2P

O modelo adotado do sistema de distribuição de mídia contínua ao vivo é baseado em uma malha computacional, orientado por pedido de dados. Esse modelo é descrito em detalhes no capítulo 3.

Em um breve resumo, em um sistema desse tipo existe um participante especial denominado *Servidor*. O servidor origina a mídia a ser transmitida por toda a estrutura P2P. Um novo participante, ao se unir ao sistema, faz contato com um subconjunto de outros participantes. Caso seja de interesse, o participante contactado pelo novato o adiciona à sua lista de parceiros e começa a interagir com ele. Cada participante reconhece e troca dados, apenas com os parceiros aos quais estão conectados. O subconjunto de participantes inicial é obtido, aleatoriamente, entre todos os participantes do sistema, através de um mecanismo independente de inicialização (*bootstrap*).

Independente da codificação do fluxo contínuo, cada participante deve descarregar dados da rede sem erros. Essa captura de dados deve ser realizada a uma taxa igual ou próxima a taxa de geração de dados pelo servidor. Nas simulações realizadas, assume-se

que os participantes possuem capacidade de armazenamento e transmissão suficiente para visualização da mídia. Além disso, o espaço de armazenamento temporário é suficiente para o compartilhamento do conteúdo (com pelo menos 2 minutos de mídia).

6.1.4 Modelo do participante:

Os participantes do sistema seguem o modelo e as características apresentadas no capítulo 4. Além disso, cada participante é classificado como *bom participante* ou como *poluidor*. No cenário geral de simulação, os poluidores repassam apenas conteúdo poluído e nunca abandonam o sistema. Eles também não se redimem, se tornando participantes *bons*. Além disso, os poluidores anunciam um mapa completo de dados, ou seja, sempre têm disponível algum dado desejado por outro participante.

Nesse mesmo cenário base, os participantes *bons* trocam mapas de dados consistentes com seus dados disponíveis/desejados. Eles podem entregar dados corrompidos com probabilidade de p_{error} e, quando não fazem verificação do conteúdo recebido, repassam dados poluídos ingenuamente.

Os participantes são limitados por recursos como largura de banda e número máximo de conexões ou parceiros. Consequentemente, só realizam parcerias entre si caso tenham recursos disponíveis para atender e requisitar dados dessa nova parceria. Quando um deles perde um parceiro ou deseja uma melhor condição do fluxo de mídia contínua, ele pode requisitar parceiros adicionais. Essa requisição é feita ao *"servidor de bootstrap"* e os prováveis novos parceiros são escolhidos aleatoriamente.

Finalmente, todos os participantes do sistema coletam informações sobre suas parcerias e troca de dados a cada 30 segundos.

6.1.5 Cenário Adotado

A Figura 6.1 mostra o cenário das simulações realizadas. Nesse cenário, é criada uma rede sobreposta com 1000 participantes, incluindo o servidor e os poluidores. Nas simulações realizadas, o servidor produz vídeo a uma taxa de 6 *chunks* por segundo, o que é um valor comum nesses tipos de aplicações [20].

Cada participante se conecta a uma média de 90 parceiros, de acordo com caracterização do capítulo 4. Inicialmente, um participante tenta se conectar à 60% do número máximo de parceiros que lhe é permitido. Ao longo de sua participação no sistema P2P, se conecta a outros, ora por pedido de parceria por outros participantes do sistema, ora por necessidade de incrementar a sua conectividade. Assim, eles sempre tentam manter o número de parceiros no valor máximo estipulado para cada um deles.

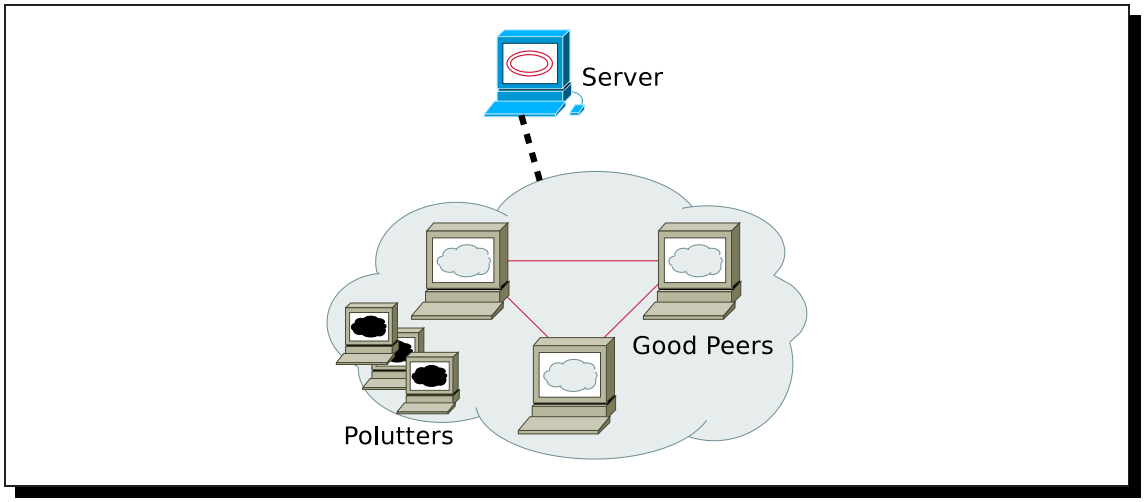


Figura 6.1: Cenário do sistema simulado.

Nesse mesmo cenário base, o servidor é criado no início da simulação. Os participantes *"bons"* vão se conectando entre os tempos 0 e 5 minutos. Os poluidores se conectam entre os minutos 2 e 5. Os resultados apresentados são de um período de 1 hora de simulação, com 35 repetições. A tabela 6.4 resume os valores utilizados.

Tabela 6.4: Parâmetros da simulação.

Parâmetro	Valor
Número de participantes	1000
Taxa da mídia	300kbps = 6 *chunks*/s
Tempo de duração da seção	1 hora
Número de vizinhos	90
Intervalo de medições	30s

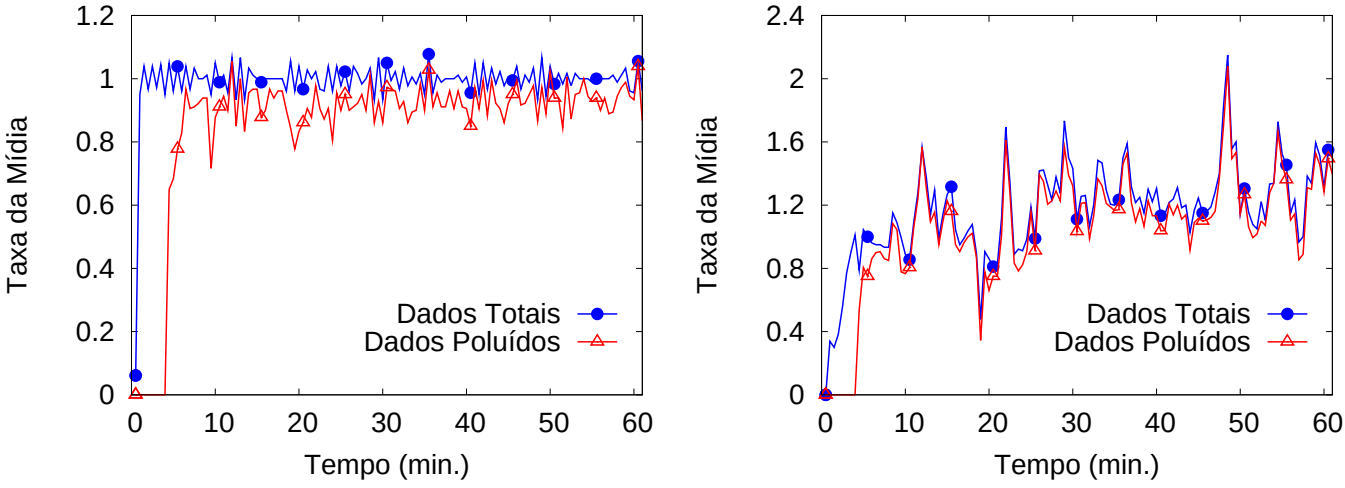
6.2 Resultados de um Ataque

6.2.1 Sistema Sem Verificação da Integridade dos Dados

Inicialmente, serão discutidos os resultados relativos à presença de poluidores em uma rede de transmissão ao vivo em P2P sem nenhum tipo de verificação de conteúdo.

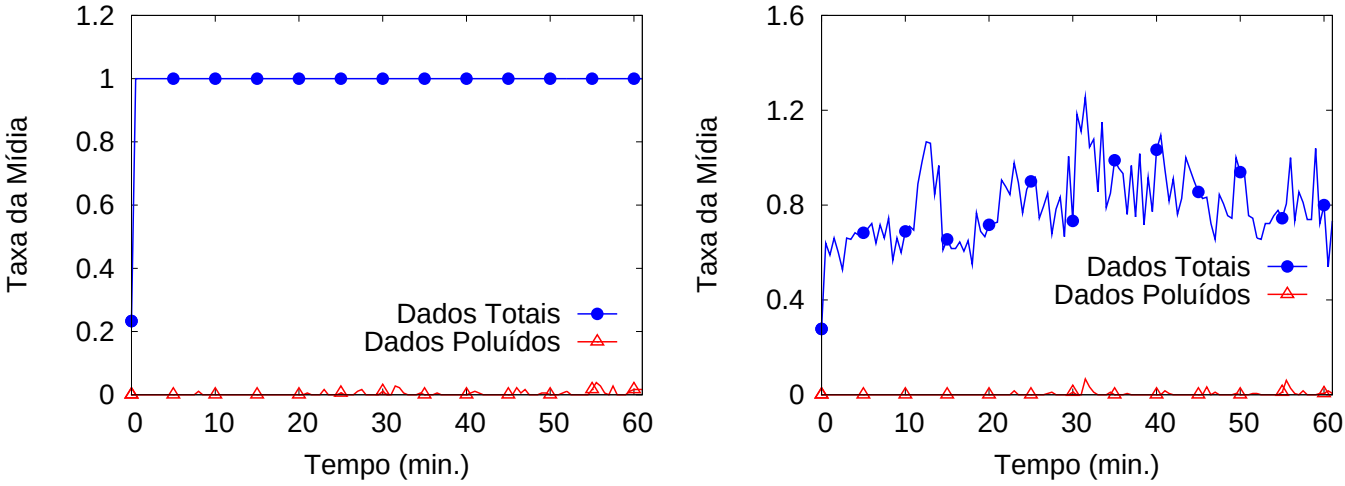
A figura 6.2 apresenta os resultados das simulações, sob a presença de apenas 1 poluidor em todo o sistema. Nessas figuras são apresentados a quantidade de dados capturados da rede P2P de um determinado participante, e também a quantidade de dados enviados por esse mesmo participante. Os dados apresentados são relativos

a 1 mídia. Assim, os participantes devem capturar 1 mídia/s para poder ter uma visualização plena da transmissão ao vivo.



(a) Recepção de determinado participante

(b) Repasse de determinado participante



(c) Recepção de determinado participante

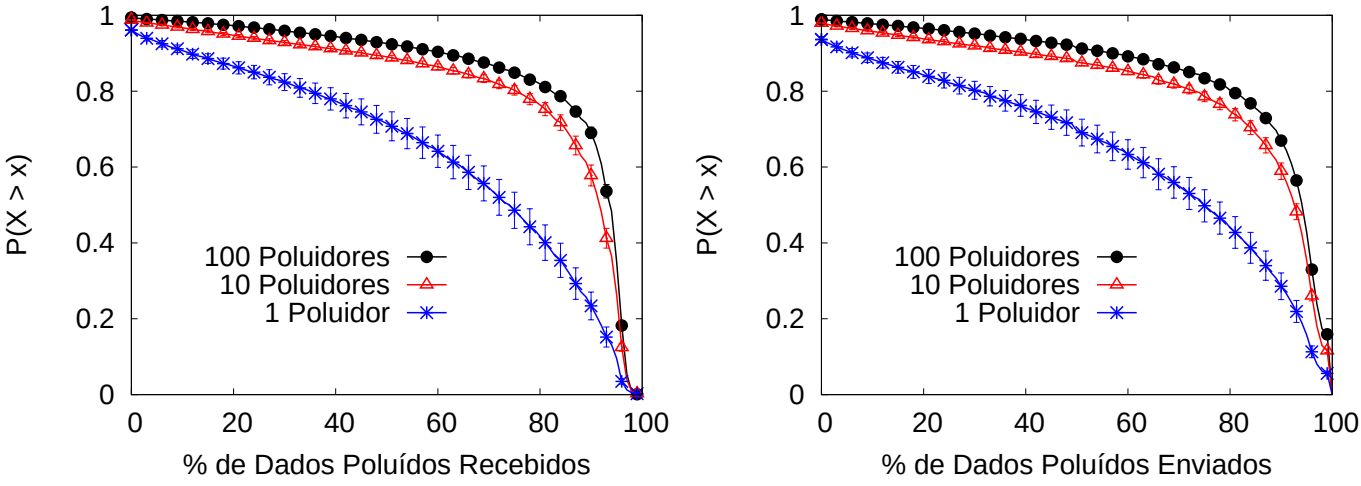
(d) Repasse de determinado participante

Figura 6.2: Visão de um participante sob ataque de poluição.

As figuras 6.2a e 6.2b apresentam o pior cenário possível para um participante no sistema. Nos primeiros minutos, enquanto o poluidor ainda não está presente no sistema, o participante recupera dados corretos da rede a uma taxa próxima a 1 mídia/s. Entretanto, após a entrada do poluidor na rede, praticamente a totalidade dos dados recebidos pelo participante está poluída. O mesmo comportamento se observa para os dados enviados nesse participante, pois, os dados que ele tem para encaminhar aos seus

parceiros são os mesmo dados poluídos recebidos da rede. Nesse caso, o participante sofreu um eclipse total do poluidor.

Porém, a visualização de um ataque a partir de um determinado participante pode levar a erros de interpretação. No mesmo contexto apresentado nas figuras 6.2a e 6.2b, as figuras 6.2c e 6.2d apresentam um comportamento totalmente diferente. Nessas figuras, o participante monitorado praticamente não sofre interferência de um ataque de poluição. Nesse caso, o poluidor não alterou os dados que esse participante recebeu. Uma provável explicação para isso, é que o participante se encontra próximo ao servidor, podendo obter a mídia ao vivo diretamente da fonte.



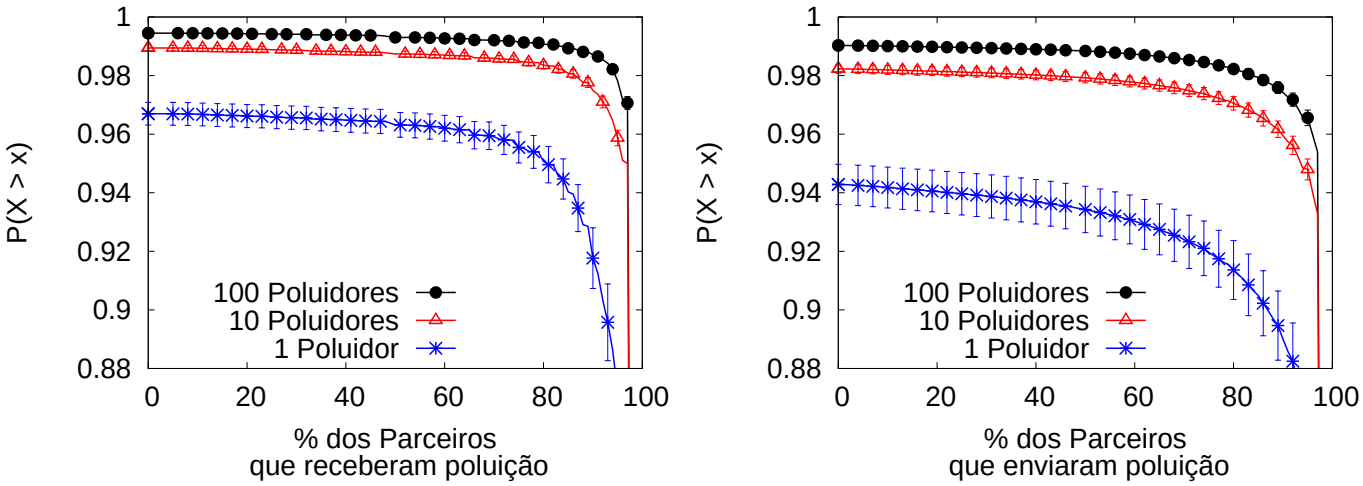
(a) Dados recebidos **(b)** Dados repassados

Figura 6.3: Visão geral do sistema sob ataque de poluição - Dados.

As figuras 6.3a e 6.3b apresentam uma visão agrupada do sistema de transmissão ao vivo em P2P. Nessas figuras são apresentadas distribuições de probabilidade que mostram o comportamento geral do sistema. Na primeira figura, é apresenta a probabilidade de se obter pelo menos um percentual de dados poluídos. Por exemplo, na presença de um único poluidor, a probabilidade de se obter pelo menos 60% dos dados poluídos é superior a 0.6. Na mesma situação, na presença de 10 e 100 poluidores, essas probabilidades sobem para valores acima de 0.9.

De forma semelhante, o repasse de dados de um participante para seus vizinhos também sofre forte influência de um ataque. Os números sofrem pequena variação com relação aos dados recebidos e, novamente, com apenas 1 poluidor no sistema a probabilidade de se repassar um fluxo com pelo menos 60% dos dados poluídos é superior a 0.65.

Os erros médios para os dados apresentados na figura 6.3a são 2.888% para 1 poluidor na rede; 0.861% para 10 poluidores e 0.601% para 100 poluidores. Na figura 6.3b, os erros para 1, 10 e 100 poluidores são respectivamente 2.819%, 0.860% e 0.601%.



(a) Dados recebidos **(b)** Dados repassados

Figura 6.4: Visão geral do sistema sob ataque de poluição - Parcerias.

As figuras 6.4a e 6.4b mostram o resultado do ataque, em relação à quantidade de parceiros contaminados ou que contaminaram. Considera-se um parceiro contaminado, o que recebeu pelo menos uma porção de dados poluídas pelo participante analisado, e considera-se um parceiro que contaminou, o que enviou pelo menos uma porção de dados poluído para o participante analisado. Assim, a medida apresentada é independente da proporção de dados poluídos enviados ou recebidos. Basta que haja a presença poluição para se realizar essa medição. No caso de um único poluidor, a probabilidade de se ter pelo menos 80% dos parceiros contaminados é superior a 0.9. Nesse mesmo cenário, a probabilidade de ser contaminado por pelo menos 80% dos parceiros é superior a 0.92.

Os erros médios para os dados apresentados na figura 6.4a são 0.531% para 1 poluidor na rede; 0.120% para 10 poluidores e 0.089% para 100 poluidores. Na figura 6.4b, os erros para 1, 10 e 100 poluidores são 0.860%, 0.192% e 0.134% respectivamente.

6.2.2 Sistema Com Verificação da Integridade dos Dados

A primeira proposta para combater ataques de poluição refere-se à checagem do conteúdo recebido da rede. Caso o conteúdo esteja correto, o participante que o

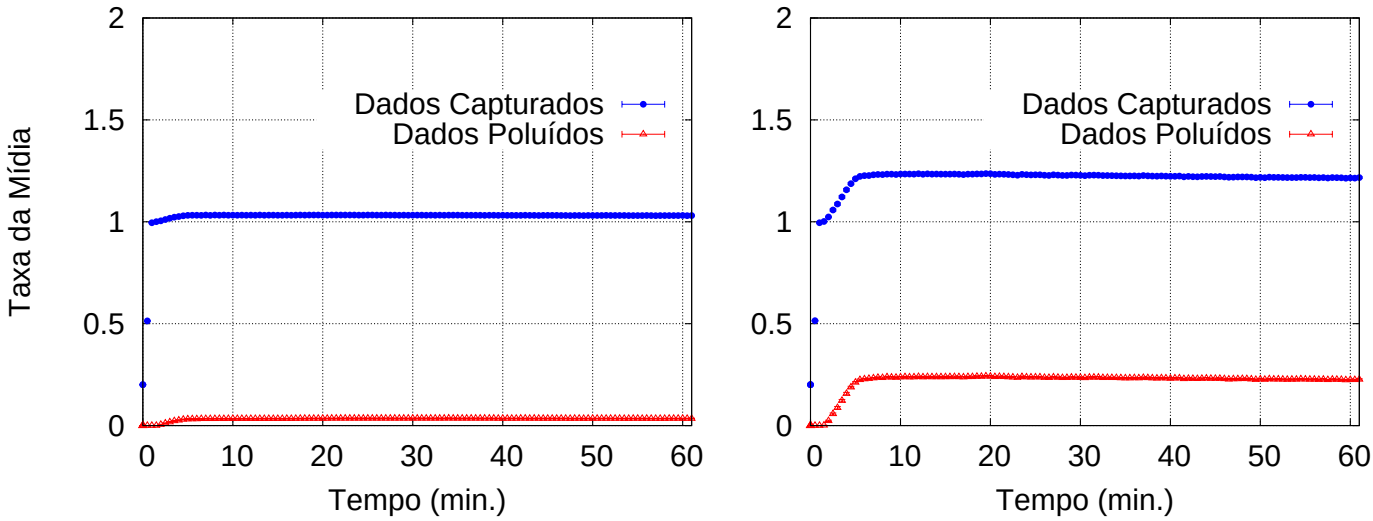
recebeu o coloca em seu armazenamento temporário para sua exibição. Caso contrário, o participante o descarta e pede novamente esse conteúdo para um outro parceiro.

Apesar da checagem do conteúdo ser uma atividade com baixo custo computacional, o pedido de retransmissão acarreta uma sobrecarga da rede. A quantidade de dados a mais que devem ser recuperados, por conta de conteúdo poluído, gasta uma porção considerável da banda de rede (como será detalhado adiante). Além disso, o tempo para exibição da mídia e/ou a quantidade de dados não recuperados da rede podem aumentar significativamente.

A figura 6.5 mostra a taxa de recepção de dados na rede P2P de transmissão de vídeo ao vivo quando, se usa o mecanismo de checagem de dados. Na presença de 1 poluidor apenas, a sobrecarga imposta à rede não é consideravelmente grande. Nesse caso, os participantes são penalizados abaixo de 5% da banda de rede, necessária para a transmissão do conteúdo ao vivo. Porém, quando o número de poluidores aumenta para 10 e 100, a sobrecarga imposta à rede de transmissão ao vivo P2P é cerca de 25% e 120%, respectivamente.

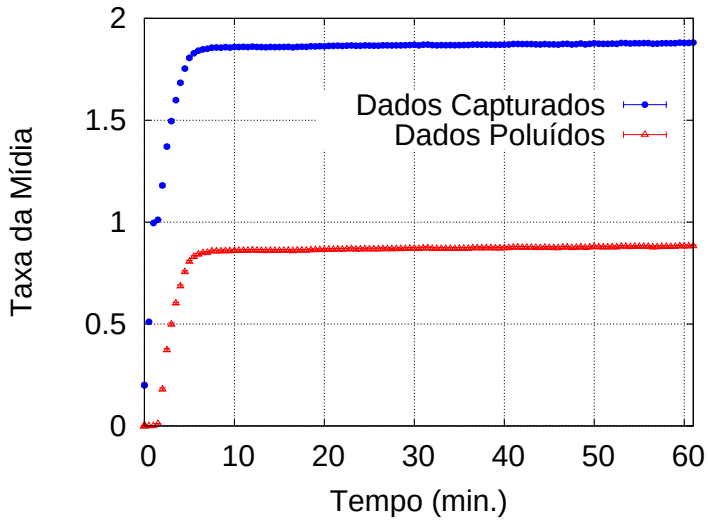
Tal diferença na banda necessária para retransmissão de dados em um sistema com 10 e 100 poluidores, pode ser explicado pelo comportamento dos participantes não maliciosos. Ao realizar um pedido de retransmissão de dados, um participante não o faz para o mesmo parceiro que enviou o dado corrompido anteriormente. Porém, com o aumento do número de poluidores, aumenta-se a possibilidade de um participante fazer o pedido de retransmissão a outro poluidor. Isso justifica uma taxa de transmissão extremamente alta para a rede com 10% de poluidores.

Esse tipo de situação, além de mostrar a ineficiência de um mecanismo simples de marcação, mostra também a necessidade de banir rapidamente um poluidor do sistema de transmissão ao vivo P2P.



(a) Rede com 1 poluidor

(b) Rede com 10 poluidores



(c) Rede com 100 poluidores

Figura 6.5: Taxa de transmissão de dados na rede abordagem "marcar e checar".

Os erros médios para os dados apresentados na figura 6.5 são 0.531% para 1 poluidor na rede; 0.120% para 10 poluidores e 0.089% para 100 poluidores. Na figura 6.4b, os erros para 1, 10 e 100 poluidores são 0.860%, 0.192% e 0.134% respectivamente.

6.3 Resumo do Capítulo

Neste capítulo, foram apresentadas simulações que mostram a extensão dos danos de um ataque de poluição, em sistemas de transmissão de mídia contínua ao vivo em arquiteturas P2P. Foram descritos os ambientes de simulação, os cenários e os modelos adotados. As simulações realizadas capturam os aspectos negativos impostos por um ataque à banda de rede e à contaminação dos participantes.

Os resultados encontrados mostram que um sistema sem nenhum tipo de proteção ou medida de combate sofre de maneira severa com um ataque de poluição. Mesmo sob a presença de um único poluidor, a probabilidade de se ter pelo menos 80% dos parceiros contaminados é superior a 0.9. Nesse mesmo cenário, a probabilidade de ser contaminado por pelo menos 80% dos parceiros é superior a 0.92. Um dos principais motivadores desse dano é o comportamento ingênuo dos participantes. Como não há qualquer tipo de verificação de dados, eles repassam conteúdo poluído para os demais participantes do sistema. Assim, tornam-se poluidores passivos.

Mesmo quando medidas de verificação de dados são adotadas, o sistema sofre penalizações em sua banda de rede quando há um ataque. Por exemplo, quando há 1% de poluidores na rede, a sobrecarga imposta à banda necessária é cerca de 25% da banda original. Esse número aumenta para 120% quando existe 10% de poluidores.

Os resultados encontrados evidenciam que a tarefa de checar dados não é suficiente para combater um ataque de poluição. Mesmo que os poluidores passivos sejam eliminados, os poluidores ativos continuarão causando um grande dano à banda de rede necessária para a transmissão ao vivo.

Capítulo 7

Ataques de Poluição no SopCast

Em geral, os sistemas P2P estão sujeitos à poluição de conteúdo, pois, todos os seus participantes podem atuar como servidores para um outro grupo de participantes [48]. Nesse momento, eles podem modificar o conteúdo servido, sem que o sistema P2P como um todo tome conhecimento ou impeça sua manipulação. A poluição de conteúdo pode ocorrer em quaisquer sistemas nos quais os participantes são utilizados para repassar conteúdo. Basta que seja conhecida a forma com a qual os dados são manipulados na comunicação e o protocolo do sistema. Muitas vezes, a comunicação é realizada sem criptografia, o que facilita a exploração do sistema P2P [20].

A caracterização apresentada nesta seção foi realizada em uma das aplicações mais populares para transmissão ao vivo em P2P, o SopCast [81]. A caracterização e análise apresentadas estão focadas na ocorrência de ataques de poluição, e nesse sentido são exploradas relações como o tempo de permanência de um poluidor na rede e o dano que este pode causar. As principais métricas apresentadas para medir o impacto são medições das taxas de *download*, taxa de *upload* e quantidade de parcerias dos participantes da rede.

Os resultados apresentados mostram que apenas um poluidor pode atingir um grande número de participantes do sistema. De fato, mesmo que o poluidor não seja agressivo e só repasse o conteúdo que conseguiu da rede previamente, ele pode contaminar cerca de 97% de seus parceiros. Além disso, como os participantes do SopCast retransmitem dados poluídos ingenuamente, em média 65% da rede recebe conteúdo indesejado durante um ataque.

Esta seção está organizada como segue: a seção 7.1 descreve de forma resumida o ataque de poluição de dados e a maneira como é realizado nos experimentos desse capítulo. A subseção 7.2 mostra uma descrição dos experimentos e da metodologia adotada. Finalmente, a subseção 7.3 apresenta os principais resultados observados.

7.1 Poluição de Dados no SopCast

A forma de ataque de poluição neste trabalho envolve a alteração de um conjunto de *bytes* no pacote correspondente aos dados da mídia, que é transmitida ao vivo pela rede P2P. Nos experimentos realizados, quando um poluidor recebe um desses pacotes e o repassa, ele o modifica, inserindo uma assinatura em uma região predeterminada do pacote. Essa assinatura contém uma sequência numérica que permite identificar a trajetória de um dado poluído pela rede P2P.

No caso do SopCast, há uma distinção clara entre pacotes de dados e pacotes com mensagens de controle do protocolo. Os dados são transmitidos por uma porta específica, com utilização do protocolo UDP. Além disso, os tamanhos das mensagens são significativamente diferentes [33, 79, 90]. Um pacote de dados tende a ser grande, ocupando todo o quadro de transmissão. Dessa maneira, a identificação de um pacote a ser poluído (marcado pelo poluidor) é realizada a partir da porta de origem no poluidor e o tamanho do pacote de dados.

Para interceptar um pacote de rede que está saindo da máquina do poluidor, utiliza-se uma característica do *framework* de filtragem de pacote presente no kernel do *Linux 2.6.x*, o Netfilter[1]. São criadas regras que interceptam os pacotes relativos ao SopCast na saída da máquina, ou seja, no momento em que um pacote de dados é servido. Nessa hora, uma aplicação poluidora pode concretizar as alterações de dados no pacote e, por fim, continuar a transmissão para a rede.

7.2 Experimentos Realizados

Para realizar o experimento, um canal privado de transmissão ao vivo foi criado na rede SopCast. A mídia que é transmitida foi codificada através do *"Windows Media Encoder 9"*[2] a uma taxa de aproximadamente 120kbps.

Os participantes da transmissão ao vivo do experimento são criados por programas coletores de informação, espalhadas pelos computadores do PlanetLab. Como o canal é fechado e não possui seu endereço divulgado na página de canais do SopCast, supõe-se que todos os participantes são os criados pelo PlanetLab, e nenhum outro participante não conhecido faz parte da rede.

Há dois motivadores para realizar os experimentos em uma rede fechada. O primeiro, não causar nenhum prejuízo real ao sistema SopCast. O segundo, isolar os

[1] http://www.netfilter.org/

[2] www.microsoft.com/windows/windowsmedia/

efeitos do comportamento dos participantes. De fato, a criação de uma rede própria faz com que *churns* não influenciem os resultados apresentados.

O experimento apresentado foi repetido 22 vezes. A cada rodada de experimento, foram conectados 600 participantes ao canal aproximadamente. Cada um desses participantes captura toda a atividade de rede através da porta específica do SopCast. Essa captura é feita utilizando o mesmo processo apresentado na seção 4.2. O poluidor adicionado ao sistema tem comportamento agressivo e altera todos os pacotes que envia em um determinado intervalo de tempo.

7.3 Análise dos Resultados Experimentais

Inicialmente, a figura 7.1 apresenta a taxa de transmissão de dados dos participantes da rede SopCast. Nos experimentos conduzidos, a grande maioria dos participantes tem uma taxa de transmissão menor que a taxa de captura do vídeo. De fato, cerca de 80% dos participantes enviam dados a uma taxa menor que 120kbps.

Porém, existe um número pequeno de participantes que contribuem de forma significativamente maior. Transmitem praticamente 10 vezes a taxa da mídia. Esses participantes com taxa de transmissão significativamente maior, provavelmente, apresentam um grande número de parceiros e/ou são muito próximos ao servidor (inclusive o servidor da mídia original).

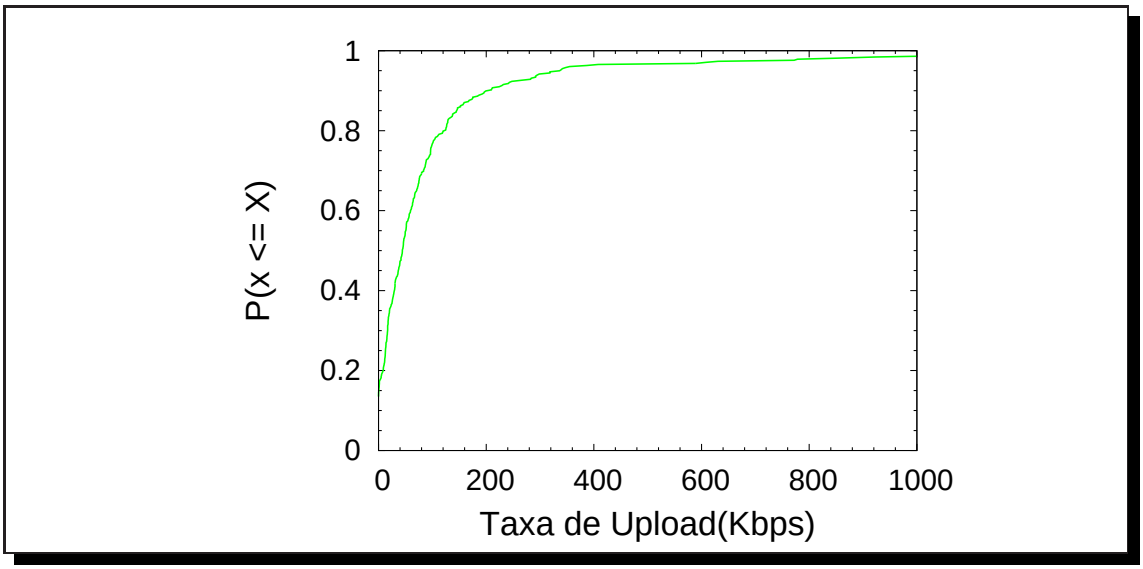


Figura 7.1: Taxa de retransmissão dos participantes do SopCast.

O comportamento da figura 7.1 indica que na rede criada, a maior parte dos participantes captura os dados de poucos participantes. Como não há limitações

na rede e a conectividade é alta, praticamente, todos os participantes conseguem algum atendimento do servidor original da mídia. Assim, pouco é exigido do compartilhamento de banda da grande maioria dos participantes da rede.

Apesar do SopCast ser uma aplicação popular e com grande número de usuários, há indícios de que apresenta características indesejáveis em seu protocolo. Por exemplo, não existe a implementação de um mecanismo de checagem dos dados. Isso facilita os ataques de poluição nessa rede de forma significativa. Quando um dos participantes inicia um ataque e forja dados, os demais participantes os repassam na rede. Eles se tornam poluidores passivos.

A figura 7.2 evidencia esse problema, mostrando a quantidade de vezes que um pacote poluído é retransmitido ingenuamente pela rede. Em aproximadamente 97% dos casos, cada pacote poluído é repassado por até 15 participantes. Ou seja, mesmo em uma rede pequena, com pequeno diâmetro, o repasse de dados poluídos pode ter um grande alcance. Em casos extremos, alguns pacotes poluídos podem ter um número de retransmissões bem maior, chegando a 82 retransmissões.

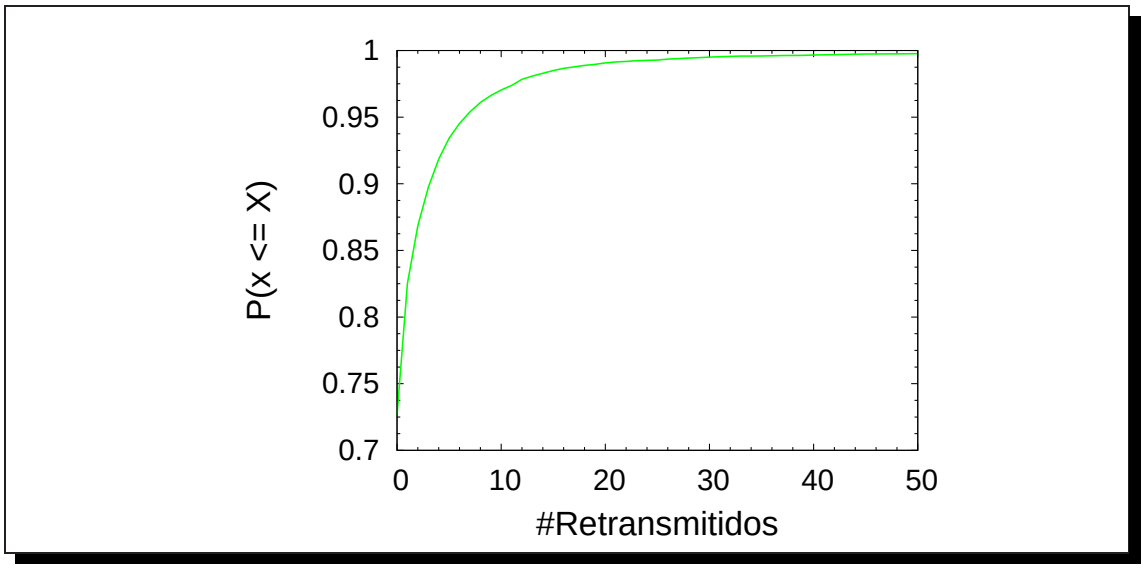


Figura 7.2: Número de retransmissões de um dado poluído.

O fato encontrado na figura 7.2 mostra-se ainda mais preocupante quando confrontado com os dados apresentados pela figura 7.1. Aquela figura indicava que a maioria dos participantes conseguia atendimento de outros poucos participantes. O servidor consegue atender quase toda a rede e mesmo assim, os pacotes que foram requisitados de um participante menos importante tiveram um tempo de vida grande na rede. Além disso, deve-se ressaltar que em cada retransmissão feita por um pacote, há a criação de um novo poluidor passivo, que pode contaminar parceiros.

A quantidade de dados poluídos descarregada da rede é apresentada na figura 7.3. Nesse cenário, um ataque é iniciado no instante t=100 segundos e encerrado em t=240 segundos. O poluidor repassa somente conteúdo que ele realmente já recebeu, porém, ao retransmitir, altera a informação.

No ataque apresentado na figura 7.3, cerca de 25% dos dados descarregados da rede, durante um ataque, são poluídos. No pior caso dos danos causados a um experimento realizado, a taxa de dados poluídos chegou a 70% do total de dados.

Esses números podem ser interpretados de maneira mais pessimista, pois, grande parte da rede consegue parceria com o servidor original da mídia. Além disso, o poluidor não forja dados. Ou seja, o poluidor só envia dados que recebeu anteriormente.

Mais ainda, observa-se claramente que outros participantes tornaram-se poluidores passivos. A taxa total de descarga de dados poluídos é superior a 10.000Kb/s e o poluidor supre cerca da metade desse total. Esse fato pode ser agravado se a rede aumentar, e a distância dos participantes em relação à fonte original aumentar. Assim, mais participantes da rede terão que recorrer aos seus parceiros para realizar a descarga de dados, o que aumenta a possibilidade de se obter um dado poluído.

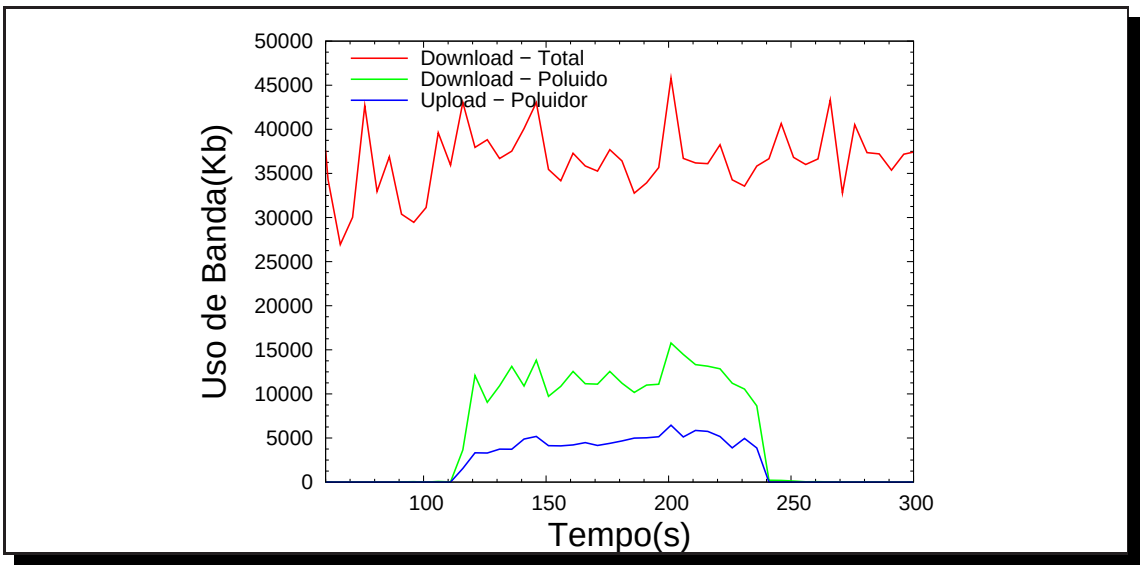


Figura 7.3: Utilização da banda da rede no SopCast.

A figura 7.4 apresenta o número médio de parceiros que o poluidor apresenta durante seu período de existência na rede SopCast. O número total de parceiros médio é cerca de 50 durante esse período. No momento de ataque, um poluidor consegue contaminar praticamente todos os seus parceiros. Observa-se ainda nessa figura que, no momento do ataque, o poluidor transmite pelo menos 1 dado poluído para cerca de 97% de seus parceiros.

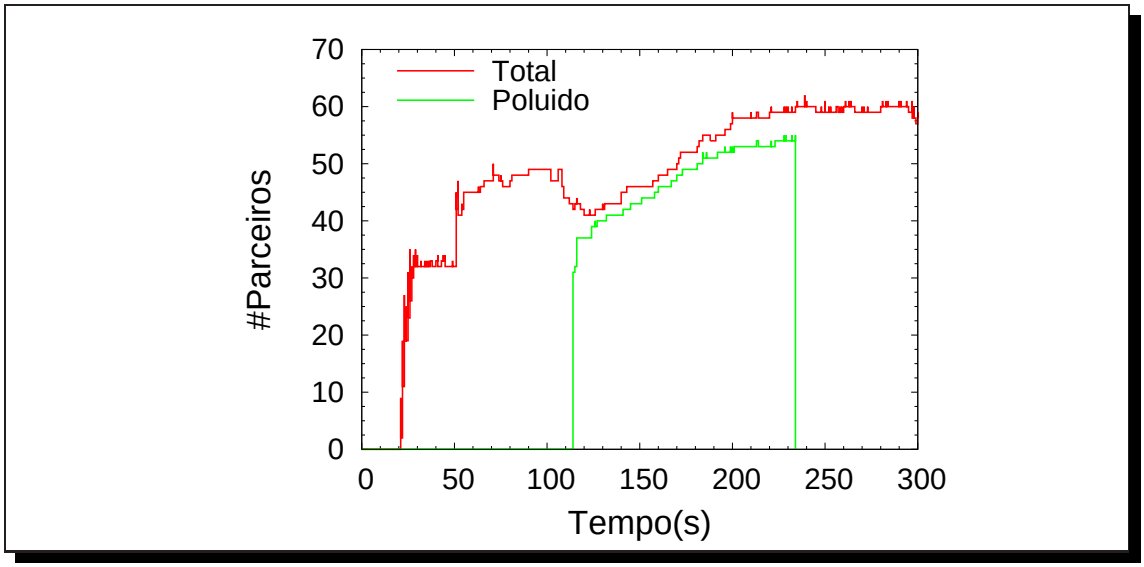


Figura 7.4: Número de parceiros do poluidor.

A figura 7.5 apresenta o número total de participantes da rede, nesse experimento, e a quantidade deles que foi contaminada. Observe que há um pico de 300 participantes contaminados em um determinado instante, o que representa 75% da rede SopCast. Na média, mais de 65% da rede foi contaminada durante um ataque. Como os participantes da rede retransmitem dados de forma ingênua, mais participantes da rede são contaminados indiretamente. O número de participantes da rede contaminados é cerca de 3 a 5 vezes maior que o número de participantes contaminados pelo poluidor.

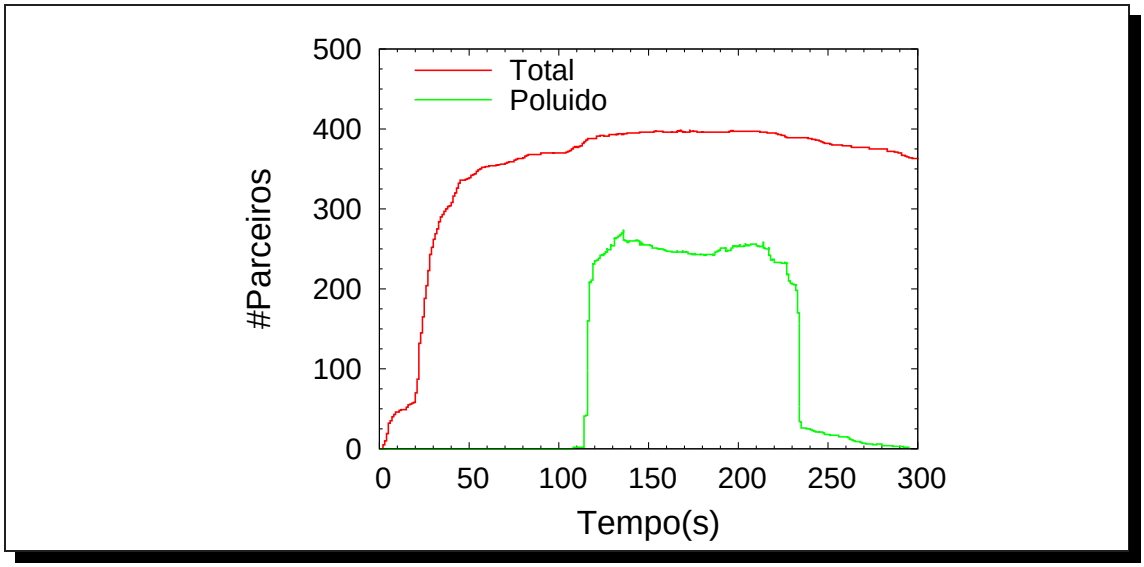


Figura 7.5: Número de participantes da rede SopCast.

Os resultados apresentados até o momento mostram que apenas um poluidor é

capaz de atingir um grande número de participantes do sistema. Mesmo que apenas alguns pacotes sejam danificados, a qualidade da mídia apresentada ao usuário pode ser severamente comprometida.

Outro ponto importante que deve ser observado é que o poluidor não forja dados, anunciando-se com quaisquer dados desejados por outros participantes. O poluidor dos experimentos apresentados é um participante típico da rede, e só retransmite dados os quais tenha recebido anteriormente.

Além disso, suas parcerias são formadas da mesma maneira que é feita por participante comum, ou seja, o poluidor não utiliza nenhum atrativo para aumentar os dados de seu ataque. Em suma, os danos causados em um ataque simples poderiam ser muito piores que os danos apresentados nesta seção.

7.4 Limite Inferior para Simulação

Os resultados encontrados nos experimentos conduzidos na rede SopCast podem ajudar a determinar um limite inferior para as simulações realizadas no capítulo 6. A determinação de um limite inferior para as simulações é importante para verificar se a simulação obedece ao que é previsto em um experimento real e assim, poder validar os resultados simulados.

Os motivos que apontam os experimentos do SopCast como limite inferior são listados abaixo:

- Há um único poluidor, e ele estabelece parcerias após a rede estar formada. Assim, grande parte dos participantes já estabeleceu parcerias entre si e, como a rede é pequena e o SopCast é guloso, muitos participantes estabeleceram parcerias com o servidor da mídia transmitida ao vivo.

- O poluidor não forja dados. Em outras palavras, ele só polui os dados que possui e não atrai muitos pedidos com um mapa falso de dados disponíveis.

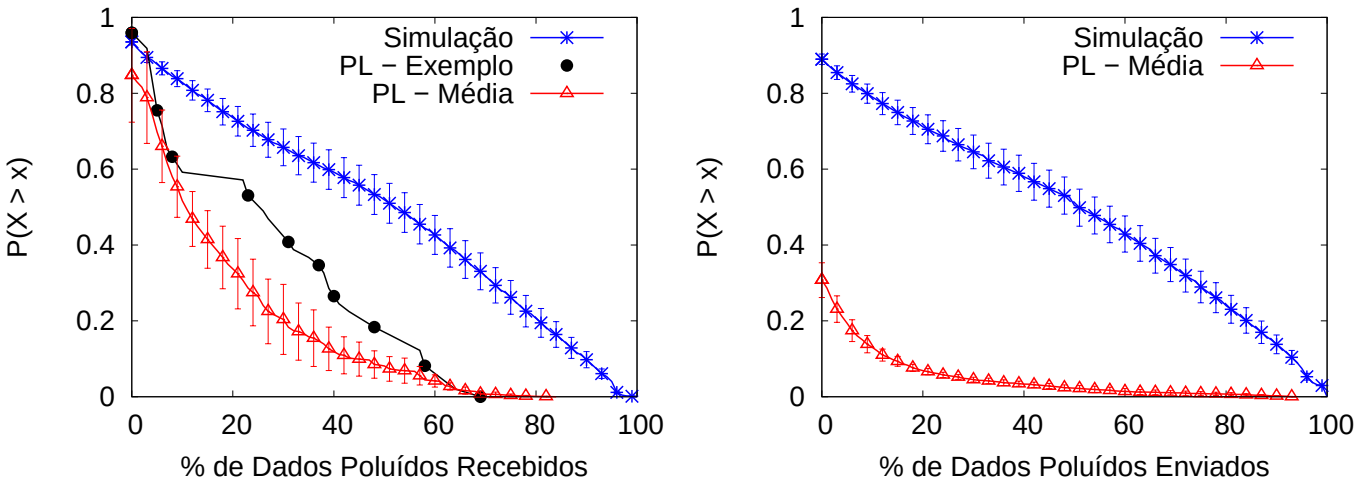
- Três fatores em conjunto levam a um grande número de parcerias entre participantes-servidor. O primeiro, a rede é relativamente pequena; o segundo, o servidor não possui limitações de recursos e, como a taxa do vídeo é baixa, ele pode atender muitos participantes simultaneamente; finalmente, os participantes do SopCast apresentam um comportamento guloso, tentando captar um grande número de parceiros.

Observa-se também um comportamento viciado dos participantes do SopCast, onde poucos participantes, que possuem muitos recursos, servem a grande maioria

dos dados. Como discutido anteriormente, nos experimentos apresentados, o servidor atende a maior parte dos dados necessitados pelos participantes do SopCast.

A figura 7.6 ilustra o dano causado pelo ataque de poluição ao SopCast, em termos de carga e descarga de dados da rede, em um ataque de poluição. Nessa mesma figura, também é apresentado o resultado encontrado nas simulações.

A figura 7.6a mostra a descarga de dados da rede SopCast durante um ataque. A curva "PL - Média" apresenta as médias dos experimentos realizados e a curva "PL - Exemplo" mostra uma instância de experimento em que a poluição teve um comportamento mais danoso que a média dos ataques. Os dados apresentados mostram que, em um comportamento médio, em cerca de 30% dos casos, os participantes receberam pelo menos 20% de dados poluídos. Note que, pela figura 7.6b, o SopCast apresenta um repasse de mensagens poluídas, e em 20% dos casos os participantes repassaram pelo menos 5% de dados poluídos.



(a) Dados Recebidos **(b)** Dados Repassados

Figura 7.6: Visão geral do SopCast sob ataque de poluição.

A tabela 7.1 apresenta o resumo dos dados para a contaminação dos participantes da rede. Como dito anteriormente, um participante foi contaminado se recebeu pelo menos 1 dado poluído em um intervalo de tempo. Da mesma forma, um parceiro contaminou o outro se ele enviou pelo menos 1 dado poluído para seu parceiro em um intervalo de tempo. O número de parceiros contaminados e que contaminaram são próximos. Cerca de 99.988% dos participantes foram contaminados por seus parceiros e, de maneira similar, cerca de 99.72% contaminaram. Esse resultado pode indicar que o ataque apresentado nos experimentos pode ter um dano maior que o estimado.

Tabela 7.1: Parceiros contaminados ou que contaminaram.

	Média	Coeficiente de Variação (CV em %)
Parceiros Contaminados	99.98	$< 1\%$
Parceiros que Contaminaram	99.72	$< 1\%$

7.5 Resumo do Capítulo

Neste capítulo, foram apresentados os resultados de experimentos, conduzidos em uma das redes mais populares para transmissão ao vivo em P2P. Os experimentos foram realizados utilizando a estrutura do PlanetLab, contando com cerca de 650 computadores. Não houve limitações impostas aos computadores utilizados nos experimentos, tanto para descarga de dados, quanto para o envio.

Durante um ataque de poluição nos experimentos realizados, um poluidor repassa conteúdo alterado para os parceiros que requisitam dados a ele. O poluidor não forja dados e nem usa mecanismos para atrair vítimas. Além disso, o poluidor se porta como um participante qualquer da rede, sem ter privilégios de localização na estrutura ou melhores recursos.

Os resultados encontrados indicam que o SopCast não utiliza mecanismos de checagem da integridade dos dados. Os dados poluídos são aceitos pelos participantes do sistema, sem nenhuma restrição. Além disso, os participantes que receberam poluição tornam-se poluidores passivos, repassando poluição para seus parceiros. Em aproximadamente 97% dos casos, cada dado poluído é repassado por até 15 participantes. Há casos extremos em que esses dados chegaram a ser repassados por mais de 80 participantes de distância da origem.

Durante um ataque, cerca de 25% dos dados descarregados da rede estão poluídos. No pior caso de danos causados a um experimento realizado, a taxa de dados poluídos chegou a 70% do total de dados da rede. Mais ainda, o poluidor consegue causar danos a cerca de 97% de seus próprios parceiros. Como eles se tornam poluidores passivos, mais participantes são contaminados. Assim, em média, mais de 65% da rede foi contaminada durante um ataque.

Finalmente, os experimentos conduzidos servem como um parâmetro de limite inferior para as simulações já realizadas. Isso permite verificar se a simulação obedece ao que é previsto em um experimento real, e assim, permite validar as simulações.

Capítulo 8

Combate à Poluição em Transmissões ao Vivo em P2P

Este capítulo mostra as abordagens utilizadas para combater um ataque de poluição em um sistema de transmissão ao vivo em P2P. Até o presente capítulo foram apresentados motivação e comprovação prática dos danos causados por ataques de poluição em sistemas de transmissão ao vivo em arquiteturas P2P. Também foi proposto um modelo de tais sistemas de transmissão e do comportamento de seus participantes. Dessa maneira, podem-se simular as soluções criadas e compará-las com os limites previamente estabelecidos.

Neste capítulo também são apresentados os mecanismos de combate à poluição implementados. Inicialmente, será utilizada uma abordagem centralizada com lista negra, como recomendado pela literatura. Em seguida, são propostas duas novas abordagens para combater o ataque no cenário de transmissão ao vivo em P2P. A primeira delas é baseada em sistemas de reputação utilizados em compartilhamento de arquivos P2P. A segunda é uma nova abordagem distribuída de reputação dos participantes do sistema. Essa abordagem apresenta um protocolo mais simples, com menor sobrecarga e com tempos de detecção e combate à poluição mais rápidos.

Como principal resultado e contribuição destaca-se o desenvolvimento da abordagem simplificada. Essa abordagem mostra-se capaz de lidar com ataques de poluição e conluio dos poluidores. Mais ainda, essa abordagem apresenta ganhos significativos sobre as demais abordagens, principalmente sob a presença de um grande número de atacantes. Os resultados mostram que, em um sistema com 100 poluidores, a nova abordagem precisa de cerca de 3% a mais de banda de rede, enquanto uma abordagem por lista negra pode precisar de mais de 100% de banda adicional.

Um aspecto importante a ser considerado é a maneira como o conteúdo pode ser

classificado em poluído ou limpo. No contexto deste trabalho, qualquer informação corrompida é considerada poluída, mesmo que não tenha sido intencionalmente modificada. Assim, qualquer uma das técnicas de marcação e verificação de *chunks* [20, 29, 30] pode ser utilizada.

8.1 Abordagem Distribuída

Este modelo é inspirado nos trabalhos de combate a poluição em compartilhamento de arquivos em sistemas P2P [14, 15]. Porém, o modelo atual foi estendido para capturar características de um sistema ao vivo P2P como alta taxa de troca de dados e limitações severas de tempo.

O modelo descentralizado de reputação tem como objetivo identificar e isolar os participantes maliciosos, sem a necessidade de um participante ou servidor especializado. O modelo apresentado neste trabalho, além de punir os atacantes, também pode promover a reabilitação de um participante previamente caracterizado como poluidor. Por exemplo, caso um participante tenha sido penalizado por repassar conteúdo poluído ingenuamente, ou por uma má condição de transmissão, ele terá oportunidade de realizar novas parcerias, tão logo mude seu comportamento.

No modelo descentralizado, um participante p_i constrói a reputação de um parceiro p_j ($p_j \in LP_i$) a partir de dois componentes. A reputação $R_i[p_j]$ apresenta um componente expressando a *experiência individual* do participante p_i em relação ao parceiro p_j, e um componente expressando o *testemunho da rede* sobre o participante p_j. A *experiência individual* de p_i em relação à p_j captura o comportamento do parceiro p_j somente com relação aos envios de dados de p_j para p_i. O *testemunho da rede* expressa a opinião de todos os outros participantes p_k sobre p_j, onde $p_k \in LP_i \wedge p_k \in LP_j$.

A experiência individual do parceiro p_j, denominada como $RI_i[p_j]$, é computada do mesmo modo que no sistema centralizado de lista negra apresentado no capítulo 2.3.6. Assim, é descrita pela equação 8.1. A cada intervalo Tr_i, p_i verifica a sua interação com p_j. Caso a interação de p_i com p_j tenha um grau de poluição aceitável ($n/r \leq limiteNR_i$), p_i incrementa o componente individual $RI_i[p_j]$. Caso contrário, p_i decrementa o valor de $RI_i[p_j]$. Se p_i ainda não realizou interações com p_j, faz-se $RI_i[p_j] = R_{inicial_i}$.

$$R_i[p_j] = \begin{cases} max(0, R_i[p_j] - \alpha_{p_i} * (1 + n/r)^{y_i}) & \text{se } n/r > \text{valor } limiteNR_i \\ min(1, R_i[p_j] + \alpha_{g_i} * (1 - n/r) & \text{caso contrário} \end{cases} \quad (8.1)$$

Para capturar o testemunho da rede sobre p_j, denominado $RT_i[p_j]$ (para todo $p_j \in LP_i$), o participante p_i requisita aos parceiros p_k (para todo $p_k \in LP_i$) a reputação $R_k[p_j]$. Estas requisições são realizadas em intervalos de tempo Tr_i e, para evitar difamação, o testemunho $RT_i[p_j]$ é a média ponderada de todos os relatórios recebidos conforme a equação 8.2. Novamente, participantes bem reputados na aplicação P2P apresentam maior impacto no testemunho da rede de p_j.

$$RT_i[p_j] = \frac{\sum_{p_k \in LP_i} R_k[p_j] * R_i[p_k]}{\sum_{p_k \in LP_i} R_i[p_k]} \tag{8.2}$$

Caso p_i não apresente nenhum testemunho de rede sobre o parceiro p_j, faz-se $RT_i[p_j] = T_{inicial_i}$, onde $T_{inicial_i}$ é um valor inicial válido para o testemunho.

Finalmente, p_i deve pesar a importância dada a cada um dos componentes do modelo de reputação ($RI_i[p_j]$ e $RT_i[p_j]$) para calcular a reputação do seu parceiro p_j. Para isto, a constante β ($0 \leq \beta \leq 1$), controla os pesos dados à *experiência individual* e ao *testemunho da rede*. Baixos valores de β enfatizam a *experiência individual*, enquanto valores altos enfatizam o *testemunho dos parceiros*. O valor final da reputação $R_i[p_j]$ é calculado como mostra a equação 8.3.

$$R_i[p_j] = \beta * RT_i[j] + (1 - \beta) * RI_i[j] \tag{8.3}$$

Da mesma forma que ocorre na lista negra centralizada, o participante p_i decide continuar a parceria com o nodo p_j de acordo com o valor $R_i[p_j]$. Caso $R_i[p_j] \geq R\text{mínimo}_i$ ($0 \leq R\text{mínimo}_i \leq 1$), o parceiro p_j é considerado confiável e p_i continua suas interações com p_j. Caso $R_i[p_j] < R\text{mínimo}_i$ o parceiro p_j é considerado um poluidor e p_i interrompe suas interações com p_j, retirando p_j de LP_i.

A tabela 8.1 apresenta um resumo dos parâmetros utilizados para o modelo descentralizado adotado neste trabalho.

Tabela 8.1: Resumo dos elementos do modelo descentralizado.

Parâmetro	Descrição
p_i	participante i do sistema
LP_i	conjunto de parceiros de i
LPC_i	conjunto de parceiros candidatos de i
Tr_i	intervalo de tempo que p_i verifica sua interação com p_j
R_i	conjunto de reputações que p_i tem de seus parceiros
$R_i[p_j]$	reputação de p_j visto em p_i
$RI_i[p_j]$	experiência individual de p_i sobre o parceiro p_j
$R_{inicial_i}$	valor inicial para a experiência individual
$T_{inicial_i}$	um valor inicial válido para o testemunho
$RT_i[p_j]$	testemunho da rede sobre p_j, visto por p_i
r	total de chunks requisitados em um intervalo
n	total de chunks poluídos em um intervalo
$limiteNR_i$	limite aceitável de dados poluídos (relação n/r)
α_{p_i}	Fator de penalização da reputação
α_{g_i}	Fator de premiação de reputação
y_i	Fator para acelerar exponencialmente a penalização
β	controla os pesos dados à *experiência individual* e ao *testemunho da rede*

8.2 Abordagem Distribuída Simplificada

No modelo descentralizado apresentado anteriormente, um nodo p_i obtém a reputação de um parceiro p_j, através de dois componentes, a *experiência individual* e o *testemunho da rede*. Naquela abordagem há uma latência para que o testemunho da rede convirja para um valor que realmente expresse o comportamento de um parceiro p_j. Por este motivo, aquela abordagem pode sofrer com ataques em rajadas, onde os atacantes se organizam para atacar de forma concentrada, em um curto intervalo de tempo. Mais ainda, um poluidor pode se comportar bem, durante um longo intervalo de tempo, para dissimular o seu curto comportamento malicioso. Como o testemunho da rede demora a mapear esse comportamento, o atacante pode causar danos ao sistema sem ser, devidamente, penalizado ou identificado.

Os problemas de ataques em rajada e dissimulação de comportamento são motivadores para o desenvolvimento de um novo mecanismo de reputação no sistema de

transmissão ao vivo em P2P. Na nova abordagem proposta, um participante p_i somente considera a *experiência individual* como seus parceiros, para tentar detectar e isolar os participantes maliciosos. Essa abordagem pode ser contra-intuitiva, pois, se espera que o *testemunho da rede* sobre a reputação de um participante forneça informações relevantes para que p_i decida se ele tem um mau parceiro. Porém, o principal objetivo ao abandonar o componente de rede é tornar a identificação e a punição dos poluidores mais rápida e flexível.

Além disso, nota-se que a convergência de opiniões da rede sobre o comportamento de um determinado participante demora mais que uma decisão baseada na experiência individual. Mais ainda, os interesses das parcerias de p_i podem mudar rapidamente. Um determinado parceiro p_j pode atuar de maneira honesta e altruísta durante um longo período de tempo e, repentinamente, p_j pode se tornar um poluidor ativo, atacando p_i em um rápido ataque de rajada.

Para o cálculo das reputações de seus parceiros, o participante p_i computa a *experiência individual* do parceiro p_j do mesmo modo que é realizado no sistema descentralizado. Ambos os sistemas descentralizados seguem a equação 8.1 apresentada para a abordagem de lista negra centralizada. A reputação final que p_i atribui ao parceiro p_j é a própria experiência individual ($R_i[p_j] = RI_i[p_j]$).

A reputação calculada por um participante p_i, considerando apenas a *experiência individual*, não permite reabilitação de um parceiro p_j. Caso o parceiro p_j passe por más condições temporárias, em sua conexão de rede e, por consequência, seja marcado como um poluidor, p_j nunca mais terá outra oportunidade de interação com p_i. Mesmo que p_j se torne um excelente parceiro para outros participantes do sistema, p_i não dará chances de uma nova interação à p_j. Como provável consequência, a taxa de transmissão total da rede P2P pode cair, levando vários participantes a experimentar uma má qualidade no serviço (e.g. p_j pode não conseguir outros bons parceiros).

Assim, para evitar bloqueios injustos e inflexíveis, o novo sistema de reputação reage às condições da rede, ora facilitando parcerias, ora punindo mais rapidamente os participantes que enviam conteúdo indesejado. Esse novo mecanismo tenta capturar as reais condições do sistema P2P de transmissão ao vivo e, dependendo do que ocorre no sistema como um todo, ele relaxa a condições para avaliar as reputações. Dessa forma, um participante pode realizar parcerias que antes estavam bloqueadas. Por outro lado, o mecanismo também reage às condições adversas do sistema P2P, e enrijece as condições do mecanismo de reputação. Assim, prováveis parceiros maliciosos são punidos mais rapidamente.

De maneira mais precisa, definem-se dois estados distintos para o sistema de transmissão ao vivo em P2P, denominados *calmaria* e *tempestade*. O sistema P2P no

estado de *calmaria* não apresenta ataques de poluição e de maneira oposta, durante estado de *tempestade*, o sistema P2P é atacado pelos poluidores.

Caso um participante perceba um estado de *calmaria*, ele altera para um valor mais baixo o valor da sua nota mínima de reputação de corte. Assim, os parceiros ou candidatos a parceiros podem ter uma nota mais baixa e não serem considerados atacantes. Caso contrário, o participante perceba um estado de *tempestade*, ele altera o valor da nota mínima para um valor mais alto, bloqueando mais rapidamente os parceiros indesejados.

O valor mínimo para que um parceiro seja considerado confiável, $R\text{mínimo}_i$, é modificado por um participante p_i a cada intervalo de Tm_i. Esse valor pode variar entre um valor mínimo Rt_{min_i} e um valor máximo Rt_{max_i}, onde $0 \leq Rt_{min_i} < Rt_{max_i} \leq 1$. O participante p_i realiza a penalização de $R\text{mínimo}_i$ por incrementos de γ_{p_i}; ele relaxa o limite decrementando o valor atual de γ_{g_i}. Para bloquear os atacantes de maneira mais rápida, faz-se $\gamma_{p_i} > \gamma_{g_i}$. Assim, a modificação do valor $R\text{mínimo}_i$ é realizada dependendo se a rede está em *calmaria* ou *tempestade*. Esse comportamento é descrito pela equação 8.4.

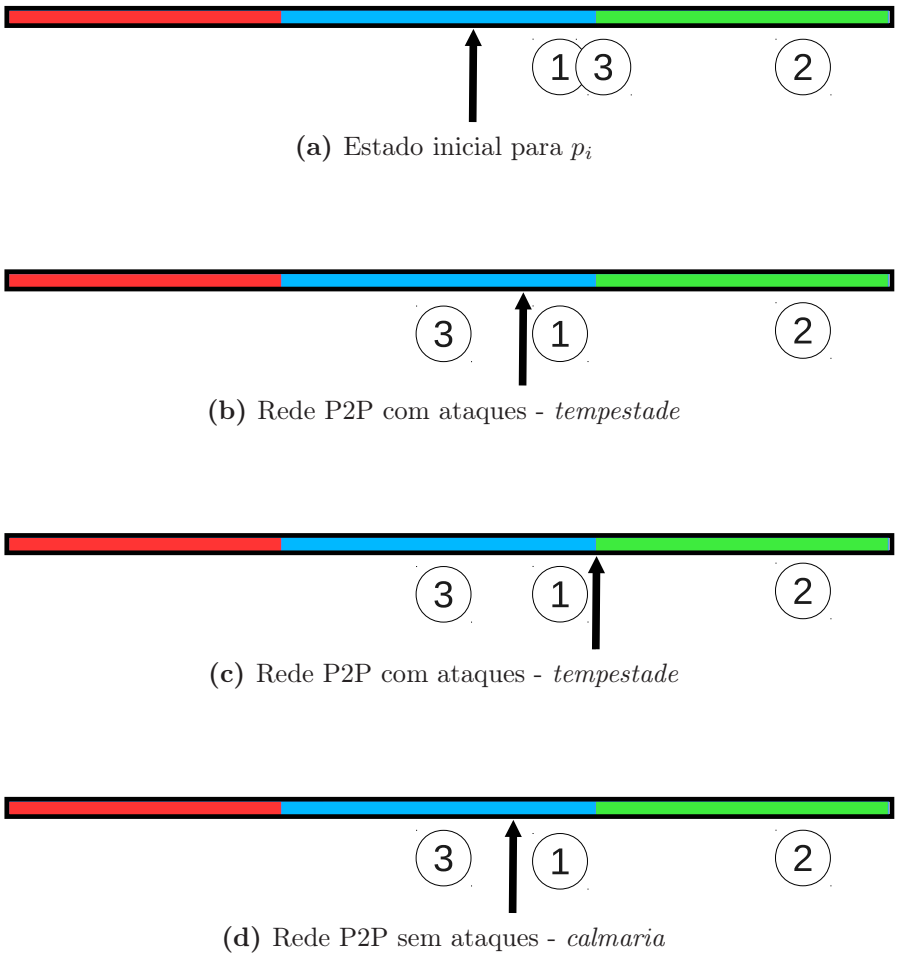
$$R\text{mínimo}_i = \begin{cases} max(Rt_{min_i}, R\text{mínimo}_i + \gamma_{p_i}) & \text{Rede em } \textit{tempestade} \\ min(Rt_{max_i}, R\text{mínimo}_i - \gamma_{g_i}) & \text{Rede em } \textit{calmaria} \end{cases} \tag{8.4}$$

O estado de um sistema distribuído é um atributo que não há como ser determinado sem observar todos os participantes da rede. Assim, não é possível determinar com exatidão se o sistema P2P de transmissão ao vivo está em *calmaria* ou *tempestade*. Porém, um participante p_i pode tentar inferir o estado global do sistema P2P a partir de sua visão local, ou seja, a partir das interações com seus parceiros.

Caso o participante p_i identifique conteúdo poluído vindo de seus parceiros, ele poderá suspeitar que a rede esteja sob ataque. Caso nenhum de seus parceiros envie conteúdo poluído, ele poderá suspeitar que a rede está em uma *calmaria*. Essa interpretação pode refletir o comportamento da rede, uma vez que um ataque, geralmente, é realizado por um conjunto de poluidores que combinam entre si estratégias para causar maiores danos ao sistema. Por exemplo, um participante p_i pode receber poucos dados poluídos de um parceiro p_j ou do parceiro p_w. Tanto p_j quanto p_w não seriam descobertos, pois, enviaram poucos dados corrompidos, porém, p_i pode receber uma quantidade total de poluição que tornaria sua exibição comprometida.

Assim, se p_i receber conteúdo poluído acima de uma quantidade aceitável, deverá

interpretar o estado do sistema como (*tempestade*). Em consequência, p_i eleva o valor da reputação mínima aceitável para seus parceiros ($R\text{mínimo}_i = R\text{mínimo}_i + \gamma_{p_i}$). Agindo assim, os poluidores da rede, mesmo não perdendo reputação, devido ao comportamento dissimulado, podem ser identificados rapidamente. Caso ocorra o contrário, ou seja p_i não identificar conteúdo poluído na rede acima de um limite aceitável, pode-se interpretar que a rede está passando por uma *calmaria*. Com isto, p_i pode relaxar o limite mínimo ($R\text{mínimo}_i = R\text{mínimo}_i - \gamma_{p_i}$). Assim, se qualquer parceiro p_j passou por um problema temporário de rede, p_j poderá ter uma nova oportunidade de interagir com p_i.



(a) Estado inicial para p_i

(b) Rede P2P com ataques - *tempestade*

(c) Rede P2P com ataques - *tempestade*

(d) Rede P2P sem ataques - *calmaria*

Figura 8.1: Mecanismo de alteração do limite de reputação.

A figura 8.1 ilustra o processo de alteração do limite de reputação mínima. A figura 8.1-a apresenta um participante em seu estado inicial. Esse participante possui 3 parceiros. No momento inicial, nenhum deles é classificado como poluidor.

Assim que o participante detecta poluição, recebida por qualquer um de seus parceiros, acima de um valor, ele altera o limiar de reputação para classificação de um poluidor. A figura 8.1-b mostra o parceiro p_3 perdendo reputação por ter enviado conteúdo poluído e, ao mesmo tempo, o limite de reputação é alterado. Se o limite se mantivesse como anteriormente, provavelmente a identificação do poluidor demoraria algum tempo a mais.

Caso o ataque continue, o participante também continua alterando o seu limite. Imagine a situação em que um participante não seja um poluidor ativo, mas repasse conteúdo passivamente. Essa situação pode indicar que a rede está em um período de *tempestade* e, por esse motivo, o participante deve se precaver. A figura 8.1-c mostra uma situação que o participante recebeu algum conteúdo poluído. Nessa mesma figura, o parceiro p_1 foi caracterizado como poluidor, mesmo sem ter sua reputação alterada. Nesse caso, o sistema ainda estava em *tempestade* e o limite continuava se elevando até um ponto máximo.

Tão logo quanto o sistema passe por uma *calmaria*, o participante pode alterar o limite novamente, para tentar estabelecer novas parcerias. A figura 8.1d mostra a situação em que o limite é alterado e em que o parceiro p_1, anteriormente classificado como poluidor, passa a ter uma nova chance de interação.

Finalmente, a tabela 8.2 resume os parâmetros utilizados nessa abordagem.

8.3 Análise do Combate à Poluição

Os sistemas de combate aos ataques de poluição, propostos neste capítulo, foram implementados em um simulador para análise e comparação entre eles. As simulações apresentadas foram executadas da mesma forma que no capítulo 6.1. Da mesma forma que naquele capítulo, as simulações foram realizadas utilizando o NS-2 [58].

A rede sobreposta para as simulações comuns é constituída de 1000 participantes. Essa rede é criada sobre a topologia física gerada pelo BRITE, utilizando os mesmos modelos e parâmetros previamente apresentados. Por exemplo, o modelo de Waxman [94, 95] para interconectar os nodos na topologia física de rede.

As topologias geradas contêm 10000 nodos com grau médio normalmente distribuído entre 2 e 3. Os valores utilizados de α e β foram variados entre os valores indicados para a rede DFN G-Win na tabela 6.1. A cada simulação, uma das topologias

Tabela 8.2: Resumo dos elementos do modelo simplificado.

Parâmetro	Descrição
p_i	participante i do sistema
Tr_i	intervalo de tempo entre interações de p_i e LN
R_i	conjunto de reputações que p_i tem de seus parceiros
$R_i[p_j]$	reputação de p_j visto em p_i
$R_{inicial_i}$	valor inicial para a experiência individual
r	total de chunks requisitados em um intervalo
n	total de chunks poluídos em um intervalo
$limiteNR_i$	limite aceitável de dados poluídos (relação n/r)
α_{p_i}	Fator de penalização da reputação
α_{g_i}	Fator de premiação de reputação
y_i	Fator para acelerar exponencialmente a penalização
$R\text{mínimo}_i$	valor mínimo para que um parceiro seja considerado confiável onde $0 \leq Rt_{min_i} < Rt_{max_i} \leq 1$.
γ_{p_i}	fator para diminuir o limite $R\text{mínimo}_i$
γ_{g_i}	fator para aumentar o limite $R\text{mínimo}_i$
Tm_i	intervalo de tempo entre modificações de $R\text{mínimo}_i$

criadas é escolhida aleatoriamente para ser utilizada. A tabela 8.3 resume os parâmetros da topologia criada para as simulações.

Tabela 8.3: Topologias de rede geradas por Waxman.

Número de nodos	α	β	Grau
10000	$0.42 - 0.46$	$0.62 - 0.68$	$2 - 3$

O sistema de transmissão ao vivo em P2P simulado segue um protocolo de reenvio orientado por dados, e estruturam seus participantes baseando-se em malha (*mesh-pull overlay network*). Esse sistema de transmissão ao vivo foi descrito no capítulo 3.1 desta tese, e funciona da mesma forma que sistemas populares como PPLive, PPStream e GridMedia [33, 66, 67, 103]

As principais características dos participantes do sistema seguem o modelo proposto no capítulo 4. Entre essas características destacam-se o grau dos nodos (número de parceiros de um participante do sistema), o tempo de atividade do nodo e o tempo de uma dada parceria. As características específicas de cada cenário de simulação serão descritas, quando necessário, junto à simulação em questão.

Independente da codificação do fluxo contínuo transmitido pela rede P2P, cada

participante deve descarregar dados sem erros da rede. Essa captura de dados deve ser realizada a uma taxa igual ou próxima a taxa de geração de dados pelo servidor. Nas simulações realizadas, assume-se que os participantes possuem capacidade de armazenamento e transmissão suficiente para visualização da mídia. Além disso, o espaço de armazenamento temporário é suficiente para o compartilhamento do conteúdo recém capturado da rede por pelo menos 2 minutos.

8.3.1 Modelo do Participante

Os participantes do sistema seguem o modelo e as características apresentadas no capítulo 4. Ou seja, a maneira como se comportam, a quantidade de parceiros e o seu tempo de permanência no sistema seguem o modelo previamente discutido. Além disso, mais características são atribuídas aos participantes para gerar um cenário realista.

Cada participante é classificado como *bom participante* ou como *poluidor*. Os poluidores, em um cenário geral, repassam apenas conteúdo poluído e nunca abandonam o sistema. Além disso, os poluidores forjam conteúdo e sempre têm disponível algum dado desejado por outro participante. Os participantes *bons* podem entregar dados corrompidos com probabilidade de p_{error} e, quando não fazem verificação do conteúdo recebido, repassam dados poluídos ingenuamente. Quando ocorrer variações do comportamento do poluidor, serão descritas no texto desta tese.

Todos participantes são limitados por recursos, como largura de banda e número máximo de conexões ou parceiros. Conseqüentemente, eles só realizam parcerias entre si, caso tenham recursos disponíveis para atender e requisitar dados dessa nova parceria. Quando um deles perde um parceiro ou deseja uma melhor condição do fluxo de mídia contínua, este poderá tentar obter parceiros adicionais até atingir o seu limite. Finalmente, todos os participantes do sistema coletam informações sobre suas parcerias e troca de dados a cada 30 segundos.

8.3.2 Cenário Adotado

A Figura 8.2 mostra o cenário das simulações. Nelas, o servidor produz vídeo a uma taxa de 6 *chunks*/s, valor comum nesse tipo de aplicação [20]. Cada participante se conecta a um número médio de 90 parceiros. Inicialmente, um participante tenta se conectar diretamente a 60% do seu número máximo de parceiros. Ao longo de sua participação no sistema, ele se conecta a novos parceiros, ora por pedido de parceria por outros participantes do sistema, ora por necessidade de incrementar a sua conectividade. Eles sempre tentam manter o número máximo de parceiros.

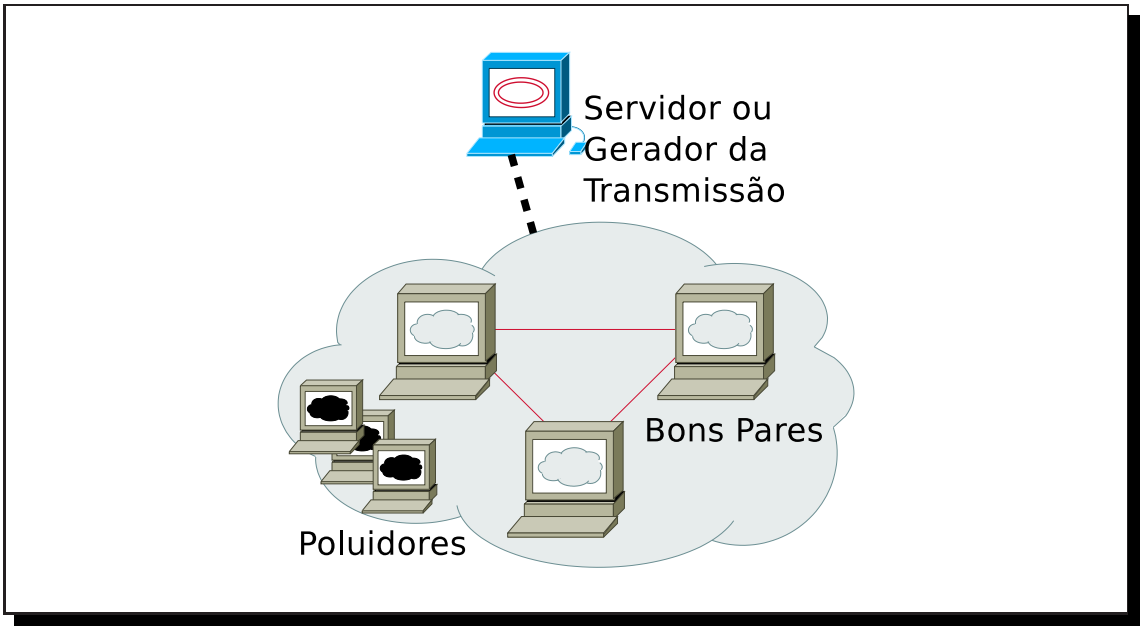


Figura 8.2: Cenário do sistema simulado.

Nesse mesmo cenário base, o participante gerador da mídia é instanciado no início da simulação. Os participantes *"bons"* vão se conectando entre os tempos 0 e 5 minutos. Os poluidores se conectam entre os minutos 2 e 5.

Os resultados apresentados são de um período de 1 hora de simulação, com 35 repetições. A tabela 8.4 resume os valores utilizados nas simulações.

Tabela 8.4: Parâmetros da simulação.

Parâmetro	Valor
Número de participantes	1000
Taxa da mídia	300kbps = 6 chunks/s
Tempo de duração da seção	1 hora
Número de vizinhos	90
Intervalo de medições	30s

8.3.3 Comparação das Abordagens.

Inicialmente, a abordagem de defesa utilizando lista negra centralizada foi testada durante um ataque de poluição onde os atacantes não realizam conluio. A figura 8.3-a apresenta os resultados para essa situação. Nessa figura, o sistema de lista negra

centralizada mostra-se capaz de detectar os poluidores do sistema e conter rapidamente os danos causados pelo ataque realizado.

Na presença de 10 poluidores, o sistema precisou de cerca de 6.6% de banda adicional para lidar com o ataque em seu período inicial. O erro padrão médio (SEM) foi inferior a 1% nesse caso. Após esse período, a banda de rede adicional necessária foi inferior a 0.03%. Na presença de 100 poluidores, a média de sobrecarga imposta ao sistema durante o período inicial do ataque foi de 33.3%, e erro padrão médio de 1.3%. O sistema de lista negra, novamente, consegue identificar os poluidores e a sobrecarga imposta ao sistema, após o período inicial de ataque, foi inferior a 0.06%.

A rápida identificação dos poluidores e o seu banimento do sistema são consequências de uma alta taxa de mensagens recebidas pelo servidor de lista negra. Todos os participantes reportam o comportamento dos poluidores para o servidor, e assim, ele pode identificar as fontes geradoras de poluição. O tempo necessário para conter o ataque foi inferior a 5 minutos.

A figura 8.3-b apresenta o resultado para o mesmo sistema de combate a ataques. Porém, no caso atual, os poluidores realizam conluio para tentar aumentar a eficiência do ataque conduzido ao sistema. Na presença de 10 poluidores, os resultados apresentam um ligeiro aumento na sobrecarga imposta ao sistema (comparado ao ataque sem conluio). Porém, a utilização de uma lista negra centralizada foi capaz de identificar e isolar os poluidores. Nesse caso, os poucos poluidores reputam-se de maneira dissimulada, mas isso é insuficiente para burlar o sistema de lista negra.

Entretanto, a existência de poucos poluidores no sistema é uma exceção. Em ataques reais realizados a sistemas distribuídos, o número de atacantes é alto e em muitas vezes, superior ao número de usuários lícitos do sistema.

Assim, quando o número de poluidores aumenta, o sistema de lista negra centralizada não se mostra capaz de identificar e isolar os poluidores. Quando os atacantes se combinam para obter uma boa reputação, a abordagem centralizada não atinge o seu objetivo, tornando-se ineficiente. Os atacantes continuam a influenciar fortemente a rede P2P, conforme é verificado pela alta sobrecarga imposta a essa rede. Durante o período inicial de ataque (cerca de 5 minutos), o sistema necessita de mais de 56% de banda adicional e, após esse período, é necessário mais de 82% de banda. Em ambos os casos, o erro médio padrão é cerca de 1.5%.

Nesse caso, há um número suficiente de participantes maliciosos para se reputarem bem no sistema centralizado de lista negra. Esse sistema não consegue definir o estado de todos os poluidores, e mesmo que seja identificado como tal, eventualmente ele terá relatos a seu favor. Assim, sempre poderá existir um número suficiente de poluidores para atacar a aplicação de mídia contínua ao vivo em P2P.

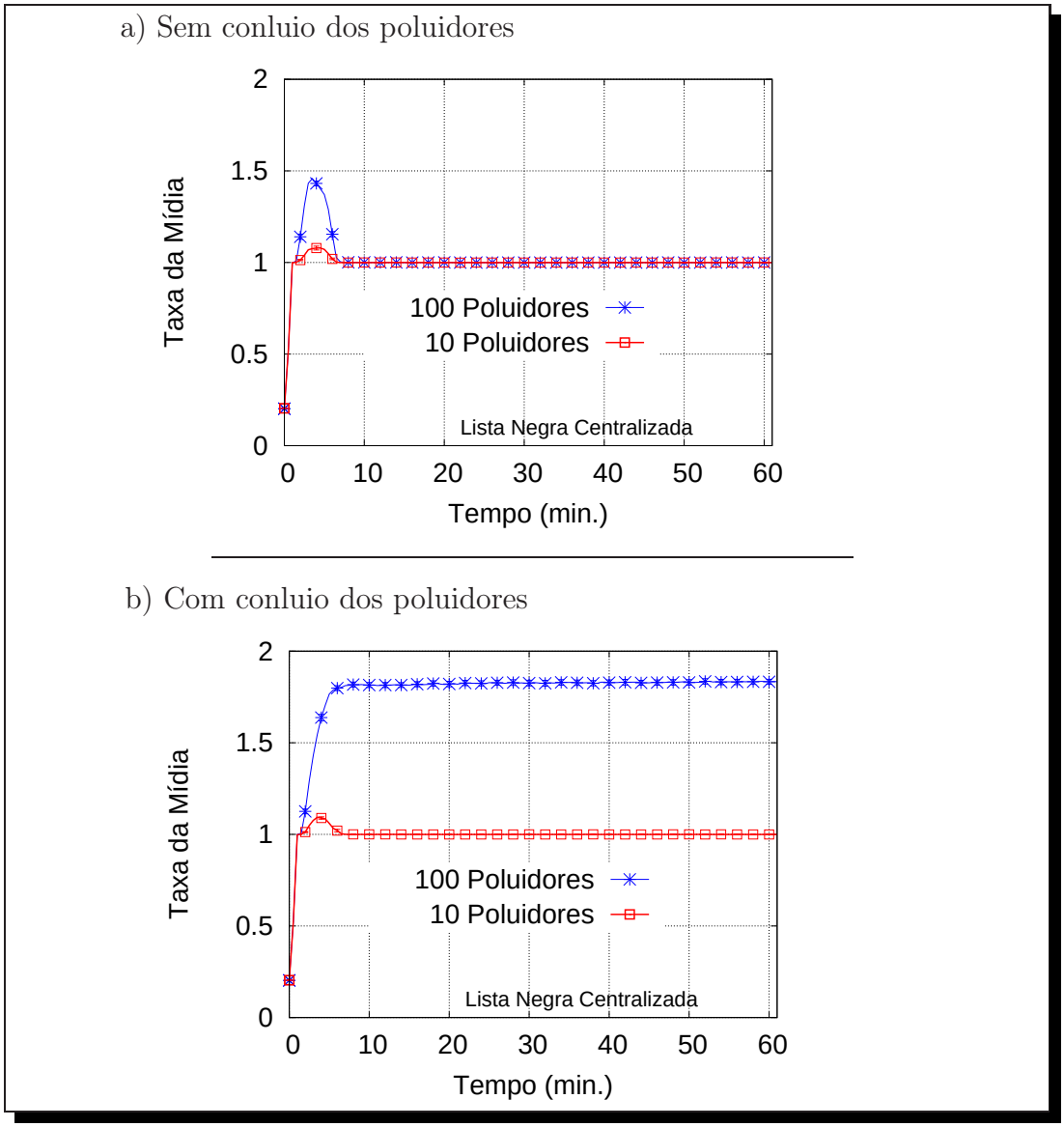


Figura 8.3: Lista negra centralizada.

A tabela 8.5 resume os resultados encontrados para o sistema de transmissão de mídia contínua ao vivo em P2P sob ataque, utilizando lista negra centralizada como mecanismo de defesa.

A figura 8.4 apresenta os resultados encontrados no combate aos ataques utilizando um sistema de reputação distribuído tradicional. Por essa abordagem, tanto os ataques de poluição com conluio dos participantes maliciosos, quanto ataques sem conluio, são identificados e têm seus efeitos minimizados.

Na presença de 10 poluidores e um ataque sem conluio, como observado na figura 8.4-a, a sobrecarga imposta ao sistema é cerca de 10.3% de banda adicional. Quando os 10 poluidores realizam conluio (figura 8.4-b), esse número tem um ligeiro

Tabela 8.5: Sobrecarga causada pelo ataque em um sistema com a lista negra centralizada (% da banda necessária).

	Início do Ataque	Erro	Após o Início	Erro
10 Poluidores	6.623 %	0.948577 %	0.0232 %	0.007 %
100 Poluidores	33.258 %	1.33306 %	0.0528 %	0.019 %
10 Poluidores & Conluio	6.944 %	0.760 %	0.0238876	0.009 %
100 Poluidores & Conluio	56.802 %	1.518 %	82.6189 %	1.451 %

aumento, e fica próximo a 11.5%. Em ambos os casos, o erro padrão é inferior a 0.7%. Após o período inicial de ataque, a sobrecarga cai para um valor cerca de 4% de banda adicional. Novamente, o erro padrão médio é semelhante, cerca de 0.3%.

Quando o número de poluidores aumenta para 100, a banda adicional necessária durante o período inicial de ataque é cerca de 56.8% superior à banda necessária quando não há ataques. Para ataques com ou sem conluio (figuras figura 8.4-a e figura 8.4-b), o erro padrão médio é inferior a 1.3%. Após o período inicial de ataque, a sobrecarga imposta cai para um valor inferior a 30%.

Para o caso de ataques sem conluio, os valores apresentados para um sistema distribuído são superiores aos apresentados para uma lista negra centralizada. No caso do sistema distribuído, não há a presença de uma entidade centralizadora, o que leva a tempos maiores para a decisão sobre a honestidade de um participante. Além disso, como há dois componentes de reputação, um deles pode demorar mais a convergir que o outro, sobre a opinião de um participante. Esse fato pode tornar as decisões mais demoradas, ou fazer com que um determinado participante não reconheça um poluidor.

Porém, o sistema distribuído não se mostra afetado pelo conluio dos poluidores. De maneira oposta ao que ocorre a um sistema de lista negra centralizada, o sistema distribuído continua sendo eficiente, da mesma maneira que ocorre quando os poluidores não realizam conluio. Até mesmo os valores encontrados para um ataque, com ou sem conluio, são semelhantes.

A tabela 8.6 resume os resultados encontrados para o sistema sob ataque utilizando o sistema distribuído. Nessa tabela são apresentados os valores para ataques de 10 e 100 poluidores, com ou sem conluio.

Finalmente, a figura 8.5 apresenta os resultados encontrados quando o sistema está sob ataque, e utiliza o esquema distribuído de reputação simplificado. O resultados apresentados têm valores semelhantes, tanto os ataques sem conluio (apresentados na figura 8.5-a), quanto os ataques com conluio (apresentados na figura 8.5-b).

O pico de sobrecarga encontrada foi inferior a 50% da banda de rede original.

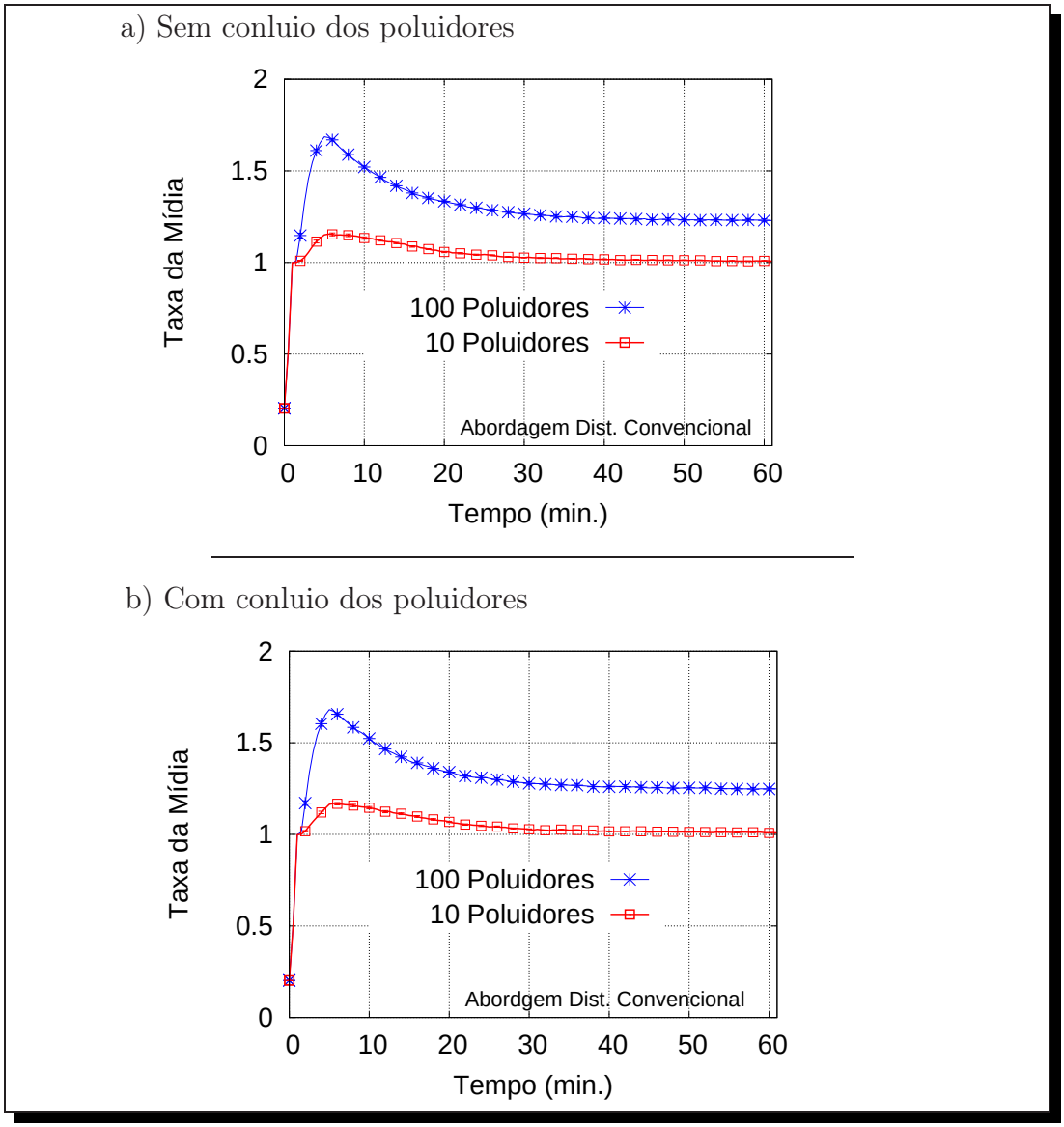


Figura 8.4: Sistema de reputação distribuído convencional.

Durante o período inicial de ataque, as sobrecargas médias para 10 e 100 poluidores no sistema são respectivamente 8.2% e 32.2% aproximadamente. Após o período inicial de ataque, esses valores caem para valores abaixo de 3%.

Na presença de 10 poluidores no sistema, o tempo para se atingir o melhor resultado, em relação à identificação dos poluidores, foi de 10 minutos. Para 100 poluidores, esse tempo foi inferior a 5 minutos. Com poucos poluidores no sistema, vários participantes não experimentam a sensação de ataque e ficam em estado de calmaria. Assim, quando um poluidor é barrado por um participante, ele encontra um outro em calmaria e faz um ataque pequeno. No caso de muitos poluidores, vários participantes ficam em um estado de tempestade quando se inicia o ataque. Dessa

Tabela 8.6: Sobrecarga causada pelo ataque em um sistema com a abordagem convencional (% da banda necessária).

	Início do Ataque	Erro	Após o Início	Erro
10 Poluidores	10.337 %	0.658 %	4.007 %	0.244 %
100 Poluidores	56.798 %	1.034 %	29.249 %	0.785 %
10 Poluidores & Conluio	11.571 %	0.645 %	4.327 %	0.292 %
100 Poluidores & conluio	56.554 %	1.287 %	30.642 %	0.949 %

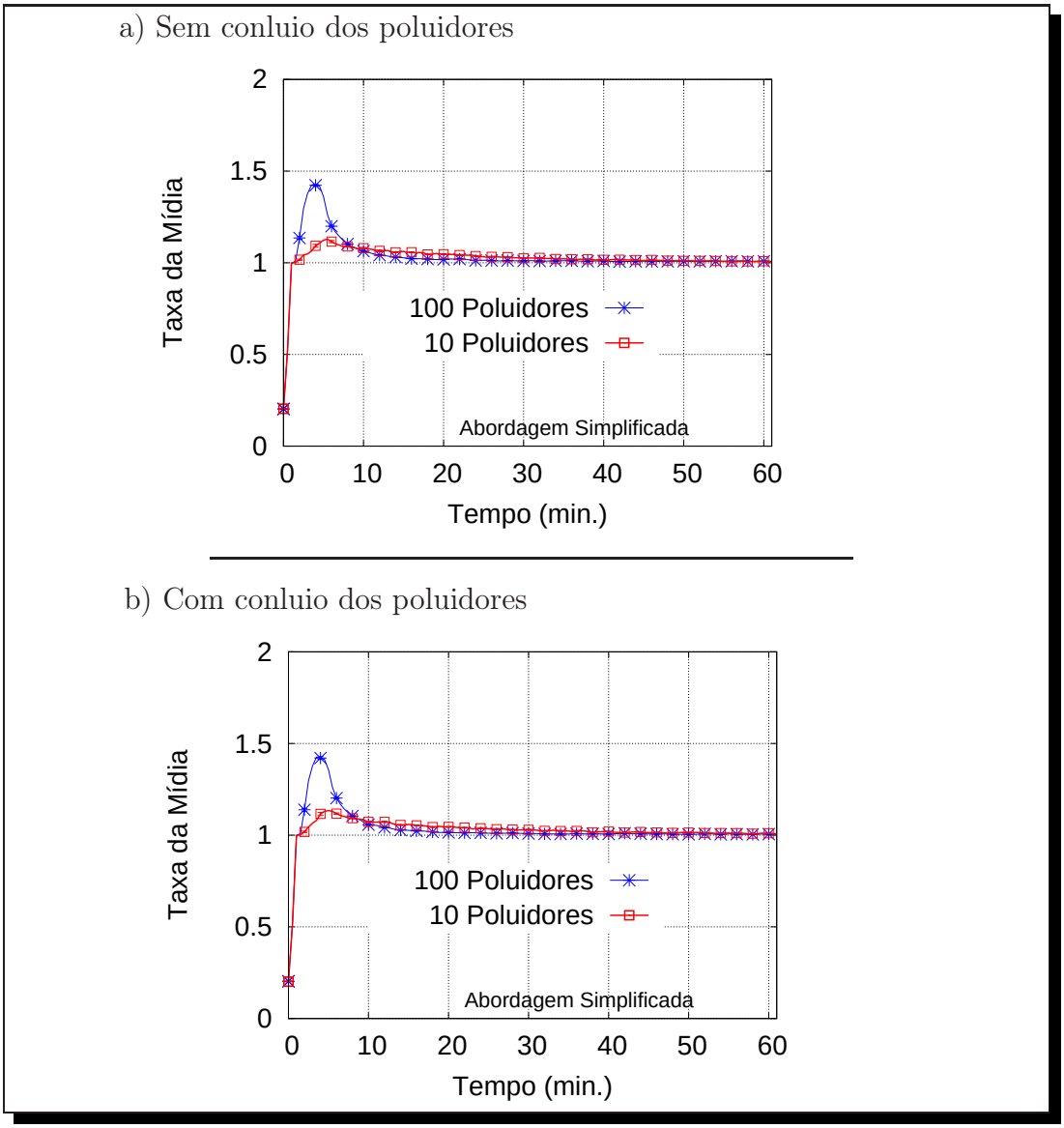


Figura 8.5: Sistema de reputação simplificado.

forma, mesmo quando um poluidor troca de parceiro, os participantes do sistema já estão preparado para um ataque.

A tabela 8.7 resume os resultados encontrados para o sistema sob ataque, utilizando o sistema simplificado de reputação. Como o sistema simplificado não apresenta o componente de rede, os resultados encontrados para um ataque, com ou sem conluio, apresentam valores próximos como resultado.

Tabela 8.7: Sobrecarga causada pelo ataque em um sistema com a abordagem simplificada (% da banda necessária).

	Início do Ataque	Erro	Após o Início	Erro
10 Poluidores	8.239 %	0.644 %	3.018 %	0.200 %
100 Poluidores	32.230 %	0.616 %	1.750 %	0.201 %
10 Poluidores & Conluio	9.308 %	0.720988 %	3.08642 %	0.201 %
100 Poluidores & conluio	32.187 %	0.632 %	1.508 %	0.162 %

O efeito da utilização das três abordagens no sistema de transmissão ao vivo em P2P, durante um ataque sem conluio dos poluidores, pode ser observado na figura 8.6. Quando o número de atacantes é pequeno, como apresentado na figura 8.6-a, as três abordagens não apresentam grandes distinções entre seus resultados. Porém, a lista negra centralizada consegue isolar mais rapidamente os atacantes, e assim, os efeitos do ataque são os menores observados.

Quando o número de poluidores é maior, como apresentado na figura 8.6-b, a abordagem distribuída convencional apresenta um resultado pior, se comparado às outras duas. Nesse caso, tanto o efeito do ataque em seu período inicial, quanto o efeito após a detecção dos poluidores, apresentam valores próximo ou acima a 30% de sobrecarga na banda de rede.

Porém, as conclusões para um ataque sem conluio não podem ser estendidas a um ataque com conluio. A figura 8.7 apresenta os resultados das 3 abordagens quando o ataque realizado apresenta conluio dos poluidores. Novamente, quando o número de poluidores é pequeno, os resultados são próximos, e a lista negra centralizada identifica e bane os poluidores mais rapidamente. Como é observado na figura 8.7-a, o tempo que uma lista negra leva para chegar ao seu melhor resultado é praticamente 2 vezes mais rápido que as demais abordagens.

Quando o número de atacantes é maior, como mostrado na figura 8.7-b, a lista negra centralizada torna-se ineficiente, por causa do conluio dos participantes. Nesse caso, a sobrecarga imposta ao sistema é praticamente idêntica à sobrecarga observada em um sistema sem nenhum tipo de mecanismo de combate. A abordagem distribuída convencional apresenta resultados semelhantes aos casos anteriores, e assim, mostra-se pouco sensível ao conluio dos participantes. Finalmente, a abordagem simplificada apresenta o melhor resultado entre as três abordagens. Como não apresenta o componente de reputação baseado na opinião da rede P2P, essa abordagem mostra-se insensível ao conluio dos poluidores. Durante o período inicial de ataque, a abordagem teve uma média de sobrecarga inferior a 33% e depois desse período, a sobrecarga média foi de cerca de 1.75%. O tempo decorrido entre o início do ataque e a sua contenção foi inferior a 5 minutos.

8.4 Análise dos Resultados Utilizando o Sistema Simplificado

A seção anterior mostrou os resultados para o modelo simplificado, utilizando parâmetros aleatórios. Como descrito naquela seção, o critério de aceitação das

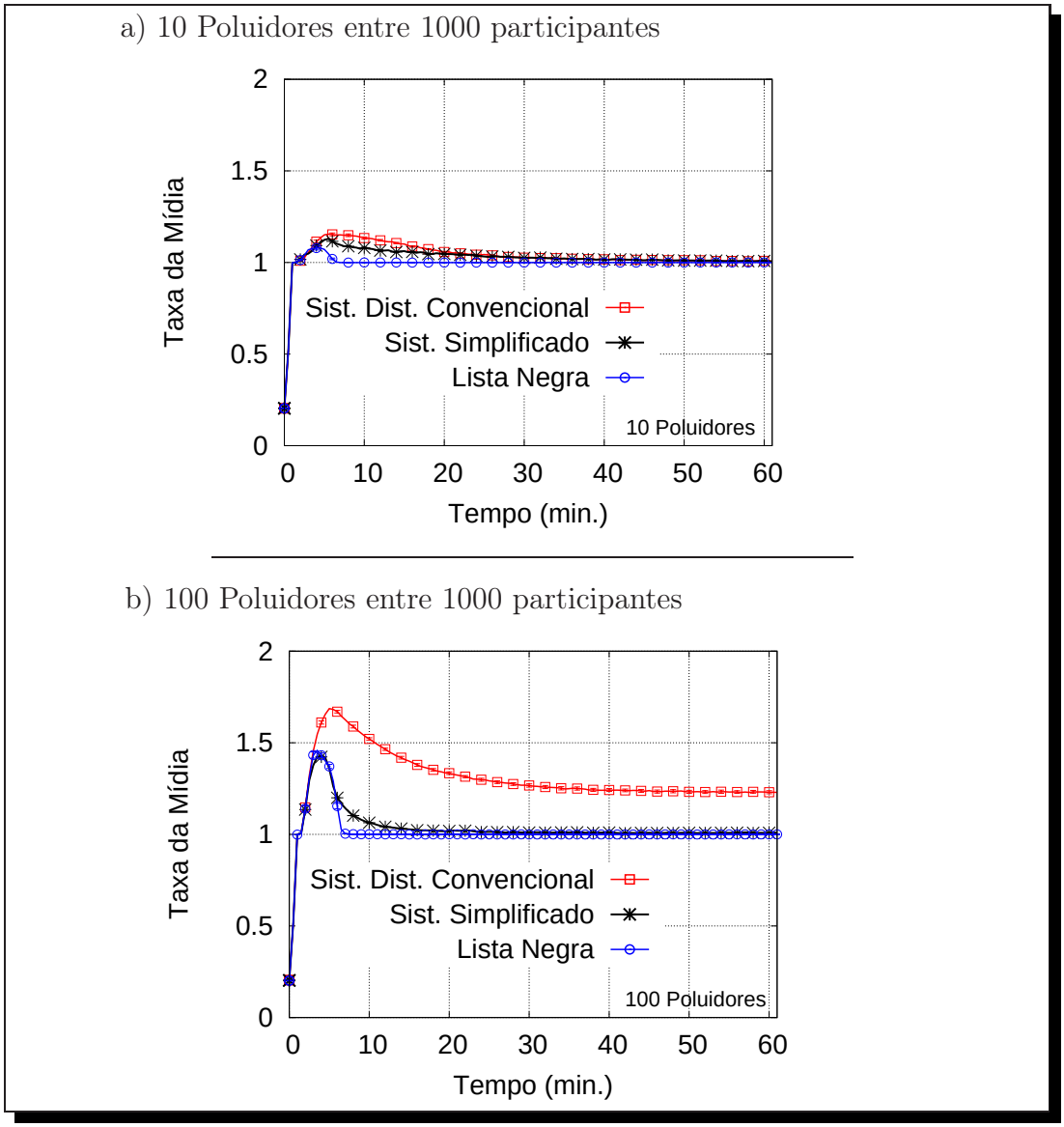


Figura 8.6: Comparação entre os métodos sem conluio dos poluidores.

simulações geradas com os parâmetros escolhidos era atender um acordo de nível de serviço em torno da taxa de mídia correta descarregada da rede P2P.

Esta seção destaca alguns dos valores de parâmetros aceitos naquela seção e verifica o impacto na variação desses parâmetros. O propósito é mostrar o quão sensível é o novo modelo proposto.

8.4.1 Variação dos Parâmetros Exponencial y_i

A figura 8.8 mostra o sistema simplificado em situações de diferentes valores para o parâmetro y_i da equação 8.1. Essa figura mostra dois comportamentos distintos,

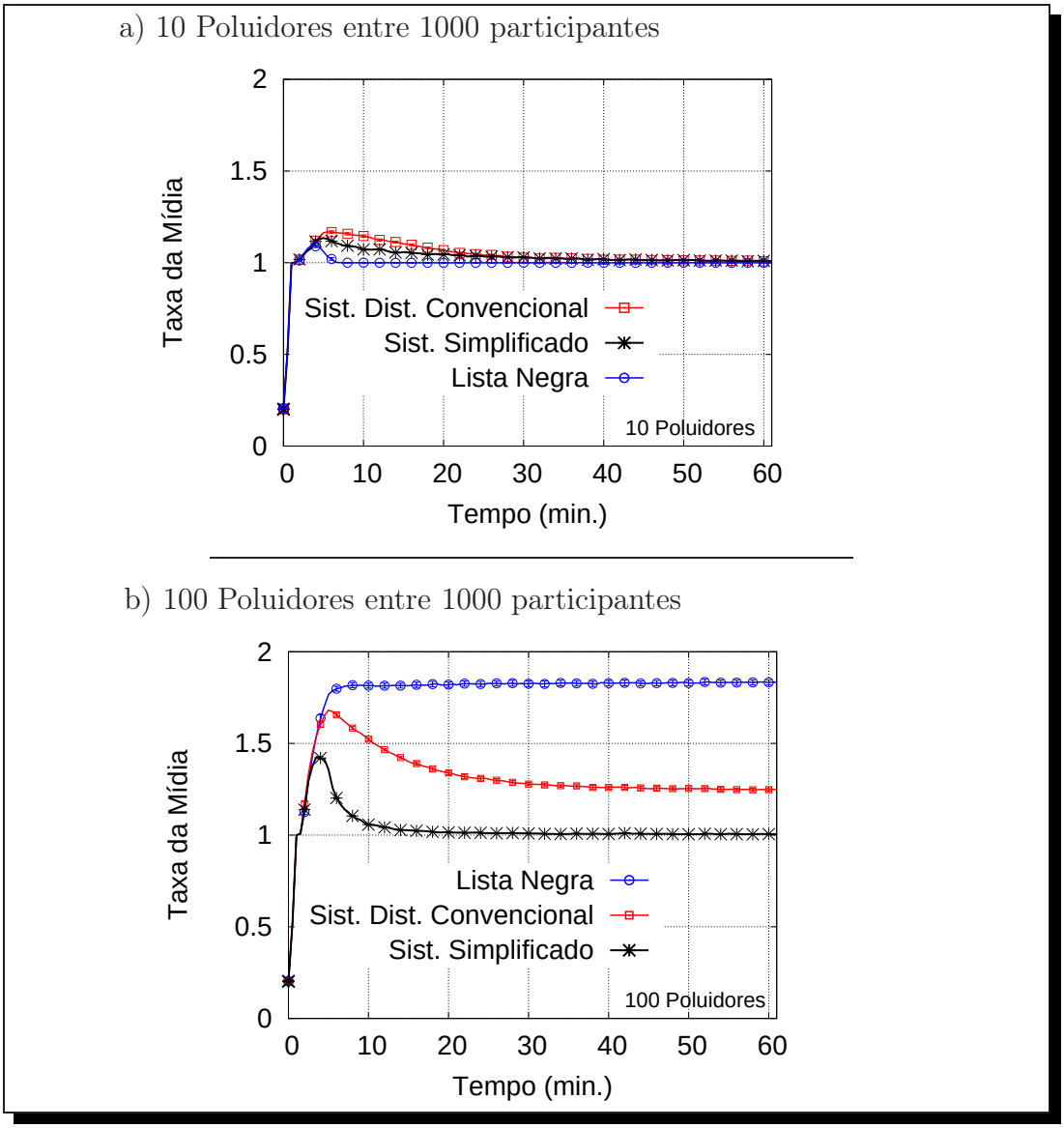


Figura 8.7: Comparação entre os métodos com conluio dos poluidores.

dependendo dos valores escolhidos para Y. Em um primeiro caso, os resultados são mais lentos e apresentam uma média de retransmissão maior. Isso ocorre porque, com a escolha de valores Y próximos de 1, os poluidores apresentam perdas lineares em sua reputação. Nesse caso, eles demoram a perder a reputação, e assim, causam maiores danos ao sistema. À medida que os valores de Y se aproximam de 2, o comportamento do sistema se estabiliza. As perdas são rápidas e, com poucas interações, um poluidor é identificado e penalizado.

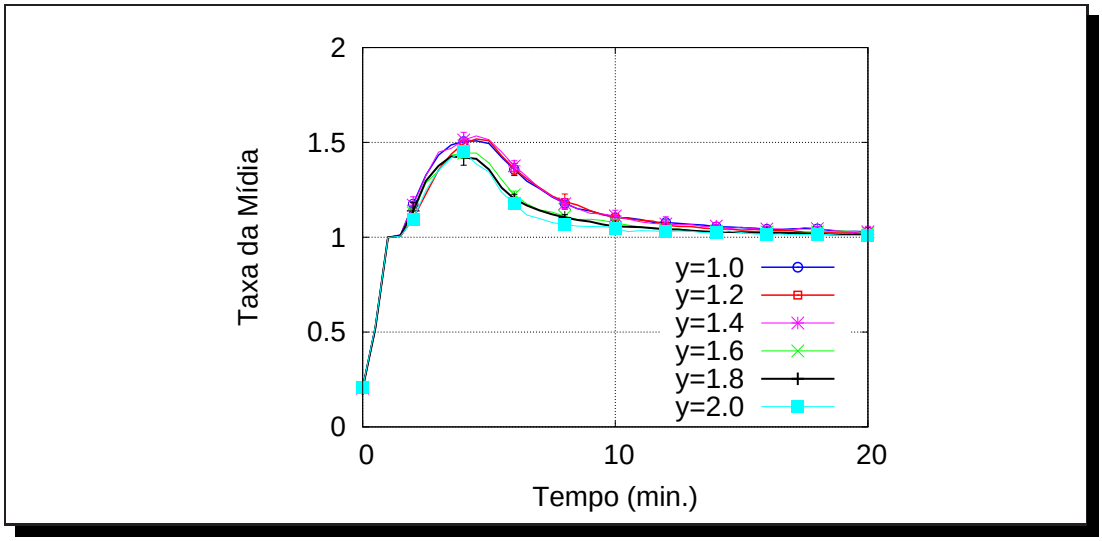


Figura 8.8: Variação do parâmetro y_i do modelo simplificado.

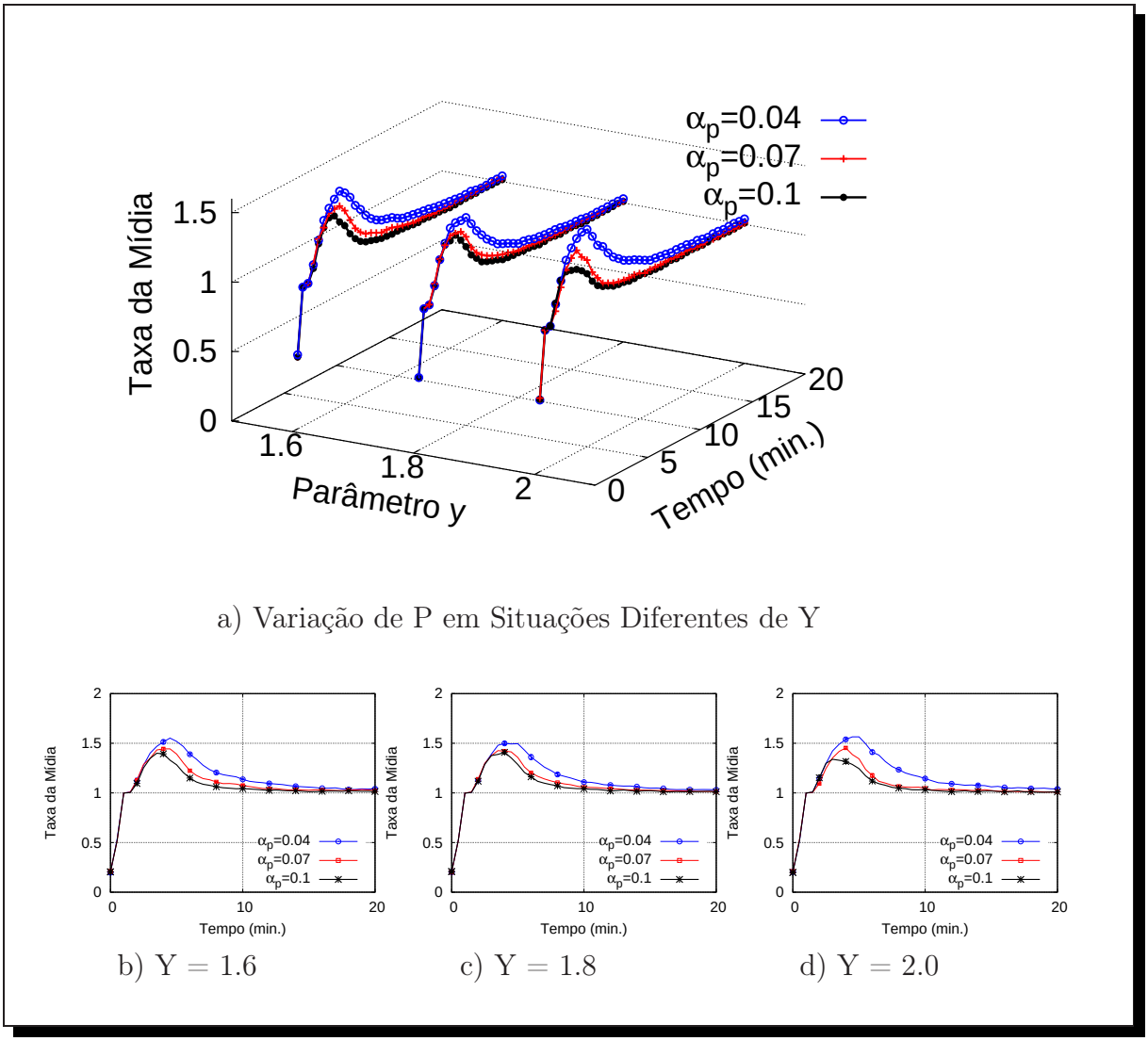
8.4.2 Variação dos Parâmetros α_p

A figura 8.9 destaca a variação do parâmetro α_p em diferentes situações para o sistema de reputação simplificado. Para deixar mais claro o efeito da variação desse parâmetro, a figuras 8.9-a traz 3 configurações de valores para o parâmetro Y no sistema simplificado. Em todos os casos, como detalhado nas figuras 8.9-b/d, a variação do parâmetro α_p gera o mesmo resultado.

Conforme o valor de α_p aumenta, as perdas são mais acentuadas em uma penalização por envio de conteúdo poluído. Como consequência, à medida que α_p aumenta, as punições são mais rápidas. Tanto o valor médio de retransmissões, quanto o tempo para o sistema atingir seu melhor resultado, são menores com valores maiores de α_p. Entretanto, quando α_p aumenta em demasia, o sistema não cumpre o nível de serviço adequado, e começa a ter uma transmissão com menor taxa. Provavelmente, isso ocorre, pois, há penalizações aos não poluidores (falso positivo).

8.4.3 Variação dos Parâmetros α_g

A variação do fator α_g é negligenciável, se comparado à variação dos demais fatores. A figura 8.10 mostra os efeitos da variação desse parâmetro, nas mesmas situações que se variou o fator α_p. No caso das simulações realizadas, esse comportamento de α_g foi observado para todos os valores que atendiam o nível de serviço acordado (taxa de transmissão da mídia sem poluição). Um valor comum, observado durante as simulações aleatórias, foi de α_g em torno da metade do valor de α_p. Um exemplo dessa situação são os valores: $\alpha_g = 0.045$ e $\alpha_{=}0.07$.



a) Variação de P em Situações Diferentes de Y

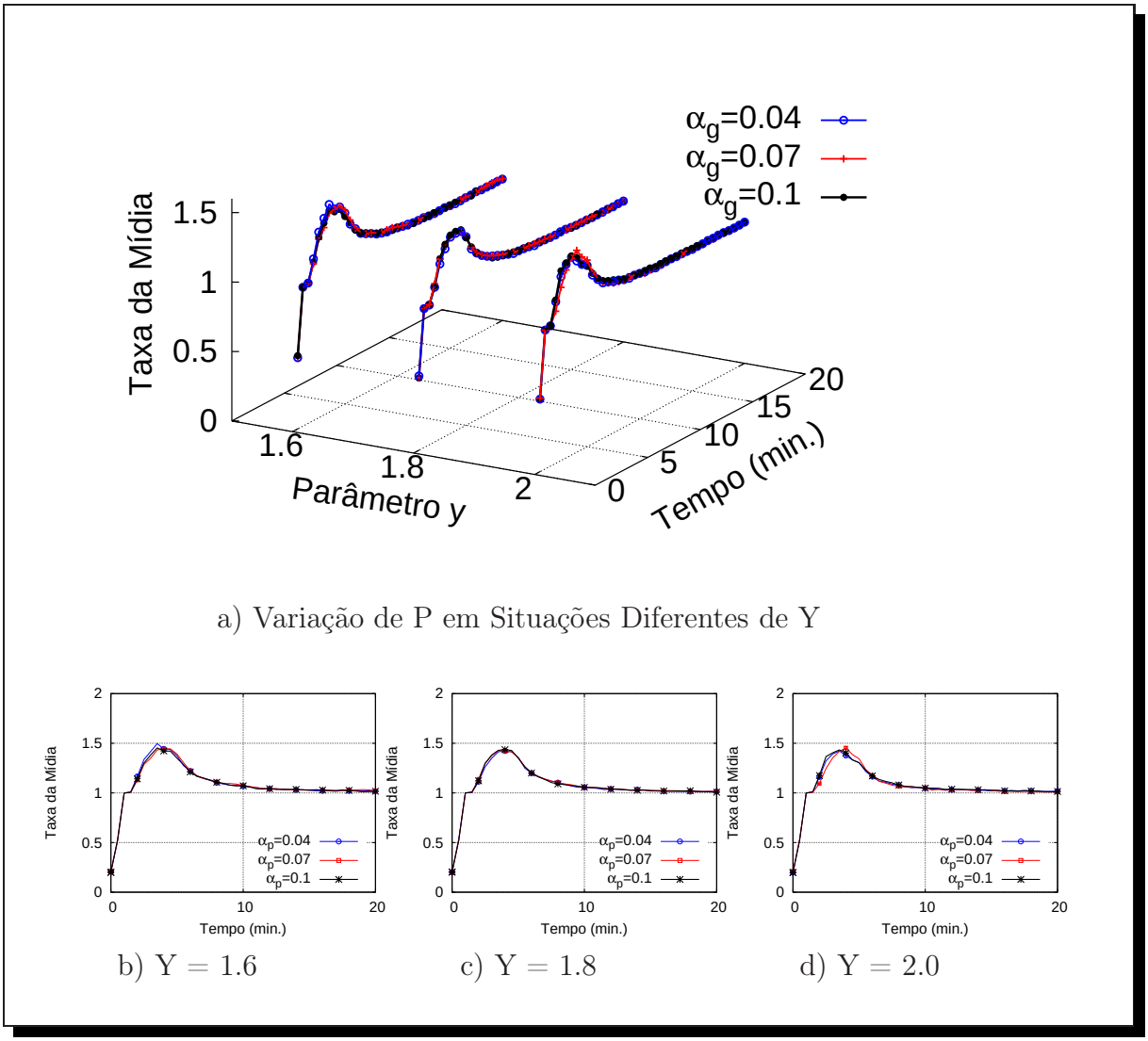
b) Y = 1.6 c) Y = 1.8 d) Y = 2.0

Figura 8.9: Variação do parâmetro P.

8.4.4 Ataques Dissimulados

Um poluidor pode tentar dissimular o seu ataque para ganhar a confiança de seus parceiros no sistema P2P. Uma vez que ganha a confiança, ele pode tentar ataques dissimulados, alternando seu comportamento entre um atacante e um participante bom. Esse tipo de comportamento pode dificultar a ação dos mecanismos de defesa contra ataques. No pior dos casos, a dissimulação de um atacante pode enganar, por completo, o sistema de defesa, tornando-o totalmente ineficiente.

No caso do sistema de reputação simplificado, os ataques dissimulados não conseguem atingir o seu objetivo. A figura 8.11 destaca 3 situações de ataque dissimulado. Em todas essas situações, o sistema simplificado conseguiu identificar



a) Variação de P em Situações Diferentes de Y

b) Y = 1.6 c) Y = 1.8 d) Y = 2.0

Figura 8.10: Variação do parâmetro G.

e punir os poluidores do sistema. Nessa figura, cada valor de D representa um perfil de dissimulação. Por exemplo, para $D = 0.50$, um poluidor fica alternando entre um estado de *não ataque* e um *estado de ataque*, com probabilidade de 50%. Um período de ataque dura alguns minutos, tempo suficiente para causar dano a um sistema sem proteção. Após o período de ataque, o poluidor volta ao estado de não ataque.

Durante o período inicial de ataque, os poluidores com perfil mais agressivo causam um dano maior ao sistema. A figura 8.11 mostra que, durante o período inicial de ataque, o perfil de $D = 0.75$ causa um dano maior que $D = 0.5$, que causa um dano maior que $D = 0.25$. Porém, tão logo os poluidores sejam identificados em seu primeiro ataque dissimulado, os 3 perfis assumem o mesmo comportamento. Isso pode ser explicado pelo fato de que, após a identificação inicial, os poluidores terão uma

nota baixa de reputação. Ficarão próximo ao limiar de serem penalizados. Assim, caso sejam reabilitados, dissimulem, ganhem confiança, tão logo comecem um ataque, eles serão banidos. Dessa maneira, os participantes do sistema se aproveitam dos poluidores enquanto dissimulam, e os abandonam rapidamente quando o ataque se inicia.

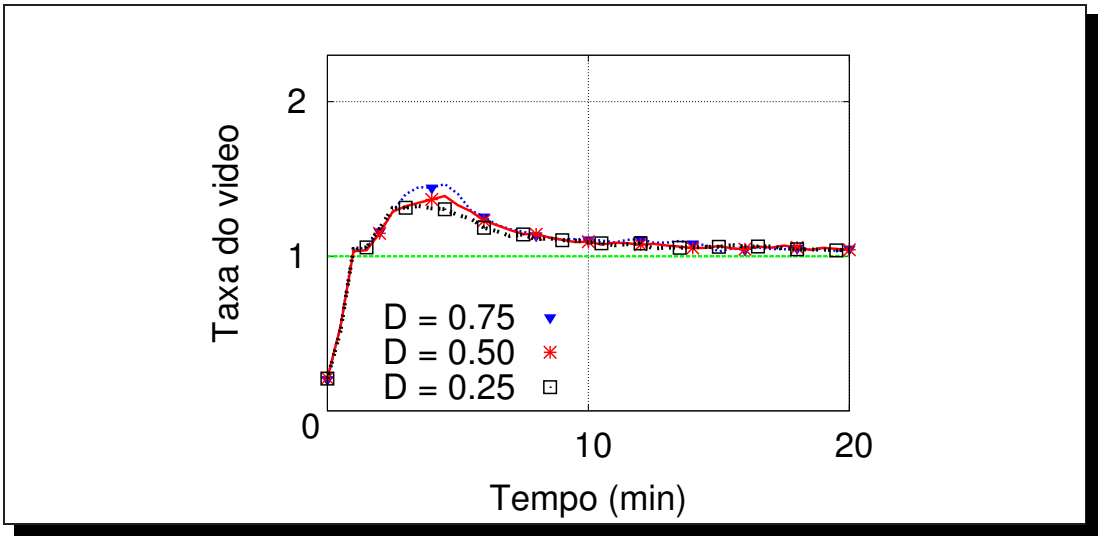


Figura 8.11: Ataques dissimulados com o sistema simplificado.

8.5 Resumo do Capítulo

Neste capítulo foram apresentadas as soluções para combater os ataques realizados aos sistemas de transmissão ao vivo em P2P. Foi dada atenção especial ao ataque de poluição, principalmente, porque os danos causados por esse tipo de ataque podem ter um grande efeito negativo no sistema de transmissão, como a perda dos participantes [20].

Atualmente, os trabalhos que tratam poluição de conteúdo em sistemas de transmissão ao vivo em P2P [20, 28–31], sugerem que se faça marcação do conteúdo para posterior verificação de sua integridade. Caso o conteúdo esteja poluído, esse é descartado e um pedido de retransmissão é feito. Junto a esse mecanismo, são sugeridas maneiras de penalização aos geradores de conteúdo poluído, como uma lista negra.

A adoção de tais medidas não é suficiente para eliminar os efeitos negativos de um ataque de poluição ao sistema. Os resultados do capítulo 6 mostram que a simples verificação do conteúdo não é suficiente, e leva a uma sobrecarga de mais de 100% à banda de rede do sistema P2P. O uso de uma lista negra centralizada só se mostrou eficiente quando os poluidores não se combinam para trapacear o sistema. Os resultados

do presente capítulo mostram que, quando há conluio dos poluidores, um sistema de lista negra centralizada perde seu propósito. Nesse caso, a sobrecarga imposta à rede praticamente se iguala a um sistema sem nenhum tipo de proteção.

O sistema de reputação descentralizado, baseado nas maneiras convencionais de se reputar conteúdo e usuários de sistemas de compartilhamento de arquivos P2P, mostra-se capaz de identificar um ataque de poluição. Mesmo em situações de conluio, esse sistema isola parte dos efeitos causados por esse tipo de ataque. Porém, é um sistema que apresenta baixa eficiência. Durante o período inicial de ataque, há uma sobrecarga de cerca de 50% na banda de rede necessária para a transmissão na rede P2P. Mesmo após o período de detecção dos poluidores, o sistema se estabiliza, com uma sobrecarga média, cerca de 30% de banda adicional.

Finalmente, o novo sistema implementado apresenta ganhos evidentes em sua utilização. Além de ser um mecanismo simples, permite reabilitação de participantes que eventualmente poluem ou tem más condições de conexão. Ele mostra-se indiferente ao conluio dos atacantes e consegue lidar com a dissimulação dos ataques.

Os resultados encontrados mostram que o sistema simplificado necessita de uma baixa banda de rede adicional. Durante o período inicial de ataque, a sobrecarga média foi de cerca de 30% na banda de rede. Os demais sistemas, na mesma situação, podem necessitar de mais de 100%. Após o curto período para atingir o melhor resultado, o sistema simplificado necessita de menos de 2% de adicional na banda de rede. O sistema tradicional, na mesma situação, tem uma sobrecarga de cerca de 30%. Mesmo quando o ataque é dissimulado, os participantes que utilizam o sistema simplificado podem identificar os poluidores. Mais ainda, podem tirar proveito deles enquanto agem dissimulando e, tão logo o ataque se inicie, eles podem isolar rapidamente os atacantes.

Capítulo 9

Conclusões

Este capítulo apresenta as principais contribuições da pesquisa desta tese. Além das contribuições, serão apontados as limitações e os problemas ainda em aberto.

9.1 Resumo

O problema básico tratado nesta tese é a segurança em transmissão de mídia contínua ao vivo em arquiteturas P2P. Em uma transmissão ao vivo em P2P, um conjunto de participantes colaboram entre si para obter os dados de uma transmissão ao vivo. O objetivo é que os participantes obtenham os dados da mídia transmitida com baixa latência e alta qualidade. Com relação à segurança, o objetivo deste trabalho é permitir que os participantes do sistema recebam uma transmissão limpa e sem a presença de dados alterados ou forjados, mantendo o baixo custo intríseco aos sistemas P2P.

Nesse sentido, o principal resultado é a implementação de um sistema de reputação que é simples e eficiente para tratar os ataques de poluição diferidos contra os sistemas de transmissão ao vivo em P2P. A idéia chave do novo sistema é realizar uma verificação de cada interação de um participante com seus parceiros. Tão logo haja suspeita de ataques, o novo sistema cria bloqueios para evitar os participantes suspeitos de gerar ataques. Para possibilitar reabilitação dos participantes que foram bloqueados e encerraram as atitudes maliciosas, tão logo as suspeitas de ataquem desapareçam, o sistema de reputação reage, flexibilizando as interações no sistema. Essa flexibilização também permite um fluxo de dados contínuo e suave, sem que um participante perca todos os seus parceiros por suspeita de ataque.

O novo sistema de reputação implementado é capaz de lidar com os ataques com um baixo custo. O novo sistema mostra-se eficiente mesmo em ataques massivo, comuns em sistema distribuídos como a *web* e compartilhamento de arquivos. Em um ataque

assim, a sobrecarga necessária durante o período inicial de ataque é cerca de 33%. Após um curto período de tempo, o sistema criado atinge o seu melhor resultado e a sobrecarga média é cerca de 1.7%. Além disso, o sistema criado mostra-se indiferente ao conluio dos atacantes.

Como resultados secundários desta tese destacam-se:

- Um modelo de sistemas de transmissão ao vivo em P2P;
- A caracterização dos participantes de um sistema de transmissão ao vivo em P2P;
- A criação de um modelo formal para verificar o impacto causado por um ataque a sistemas de transmissão ao vivo em P2P.

9.2 Limitações

O novo sistema de reputação desenvolvido apresenta algumas limitações com relação à sua abrangência e a outros problemas envolvidos aos ataques aos sistemas de transmissão ao vivo em P2P. Entre as limitações citam-se:

- Identificação e autenticação dos participantes do sistema: o novo sistema proposto não faz autenticação ou lida com os problemas de identificação dos participantes do sistema. No trabalho corrente, considera-se que cada participante do sistema apresenta um identificador de fácil acesso. Porém, o problema de identificação dos participantes de um sistema P2P pode ser complexo e sujeito a falhas. Os participantes podem se impersonificar ou criar vários identificadores para tentar burlar o sistema de reputação P2P. Alguns trabalhos criam sistemas de autoridades certificadoras para minimizar esse tipo de problema.
- Verificação dos dados: apesar da verificação dos dados ser um ponto chave para poder detectar conteúdo poluído, não foi propósito deste trabalho criar novos mecanismos para verificar o conteúdo disseminado no sistema de transmissão ao vivo em P2P. Nesse sentido, assume-se que os mecanismos previamente propostos podem ser utilizados nos atuais sistemas de transmissão ao vivo em P2P. Trabalhos anteriores mostram mecanismos de verificação com baixo custo computacional e sobrecarga de rede por volta de 5%.

9.3 Problemas em Aberto

1. Tratamento de outros ataques ou comportamentos indesejados: em sistemas de transmissão ao vivo em P2P pode haver uma série de comportamentos maliciosos e indesejados. Não foi explorado, por exemplo, um ataque em que um participante usa a estrutura da rede de transmissão ao vivo para atacar algum sistema externo à aplicação. Também devem ser tratados ataques realizados a pontos importantes da rede, que tentam exaurir os recursos e assim, por consequência, destruir o sistema de transmissão ao vivo. Um exemplo típico desse último seria os ataques realizados aos *"super-nodos"* da rede.

2. Criação de novos algoritmos de seleção de parceiros: a seleção de parceiros tem um papel importante na escalabilidade e na qualidade do serviço da rede. As aplicações populares atualmente apresentam comportamento guloso e pouco seletivo. Para criar algoritmos eficiente de escolha de parceria podem-se explorar questões como tempo de permanência na rede de um participante, os recursos disponíveis e as próprias parcerias dele. Mais ainda, as parcerias criadas podem ser feitas de maneira a maximizar a contribuição uma com a outra, sem a necessidade de disputa por um recurso recém criado.

3. Criação de sistemas híbridos seguros: há uma tendência de se mesclar redes P2P de transmissão ao vivo com arquiteturas de distribuição de conteúdo tipo cliente-servidor. Nesse tipo de sistema híbrido, os ataques ainda não são abordados como um ponto de falha. Assim, podem-se tratar esses sistemas híbridos de maneira que, um auxilie o outro a manter a qualidade de serviço, baixa latência e a segurança do sistema e de seus participantes.

Referências Bibliográficas

[1] Stephanos Androutsellis-theotokis and Diomidis Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys*, 36:335–371, 2004.

[2] Nick Antonopoulos and James Salter. Efficient resource discovery in grids and P2P networks. *Internet Research, Emerald Group Publishing Limited*, 14(5):339–346, 2004.

[3] Albert L. Barabási. The physics of the web. *PhysicsWeb.ORG, online journal*, July 2001.

[4] Beverly and H. Garcia-Molina. Designing a super-peer network. In *Proceedings of the 19th International Conference on Data Engineering*, pages 49–60, March 2003.

[5] Bittorrent. `http://www.bittorrent.com`, 2008.

[6] Alex Borges, Jussara Almeida, and Sergio Campos. Combate a poluição em sistemas p2p de mídia contínua ao vivo. In *SBRC*. SBC, 2008.

[7] Alex Borges, Jussara Almeida, and Sergio Campos. Fighting pollution in p2p live streaming systems. In *ICME*. IEEE, 2008.

[8] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: High-bandwidth multicast in cooperative environments, 2003.

[9] Ilias Chatzidrossos and Viktoria Fodor. On the effect of free-riders in p2p streaming systems. In *Proc. of International Workshop on QoS in Multiservice IP Networks (QoSIP) 2008*, 2008.

[10] Nicolas Christin, Andreas S. Weigend, and John Chuang. Content availability, pollution and poisoning in file sharing peer-to-peer networks. In *EC '05:*

Proceedings of the 6th ACM conference on Electronic commerce, pages 68–77, New York, NY, USA, 2005. ACM.

[11] Brent Chun, David Culler, Timothy Roscoe, Andy Bavier, Larry Peterson, Mike Wawrzoniak, and Mic Bowman. PlanetLab: An Overlay Testbed for Broad-Coverage Services. *ACM SIGCOMM Computer Communication Review*, 33(3):00–00, July 2003.

[12] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science*, 2009, 2001.

[13] William Conner, Klara Nahrstedt, and I. Gupta. Preventing dos attacks in peer-to-peer media streaming systems. In *13th Annual Multimedia Computing and Networking Conference (MMCN'06)*, San Jose, CA, Jan 2006.

[14] Cristiano Costa and Jussara Almeida. Reputation systems for fighting pollution in peer-to-peer file sharing systems. In *P2P '07: Proceedings of the Seventh IEEE International Conference on Peer-to-Peer Computing (P2P 2007)*, pages 53–60, Washington, DC, USA, 2007. IEEE Computer Society.

[15] Cristiano Costa, Vanessa Soares, Jussara Almeida, and Virgilio Almeida. Fighting pollution dissemination in peer-to-peer networks. In *Proceedings of the 2007 ACM symposium on Applied computing*, pages 1586–1590, New York, NY, USA, 2007. ACM.

[16] Cristiano P. Costa, Italo S. Cunha, Alex Borges, Claudiney V. Ramos, Marcus M. Rocha, Jussara M. Almeida, and Berthier Ribeiro-Neto. Analyzing client interactivity in streaming media. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 534–543, New York, NY, USA, 2004. ACM.

[17] Steve Crocker. RFC 1: Host software, April 1969. Status: UNKNOWN.

[18] Yi Cui, Liang Dai, and Yuan Xue. Optimizing p2p streaming throughput under peer churning. In *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, pages 231–235, 2007.

[19] Ernesto Damiani, De C. di Vimercati, Stefano Paraboschi, Pierangela Samarati, and Fabio Violante. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *Proceedings of the 9th ACM conference on Computer and communications security*, New York, NY, USA, 2002. ACM Press.

[20] Prithula Dhungel, Xiaojun Hei, K. Ross, and N. Saxena. The pollution attack in p2p live video streaming: Measurement results and defenses. In *Proc. SIGCOMM Peer-to-Peer Streaming and IP-TV Workshop*, 2007.

[21] John R. Douceur. The sybil attack. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 251–260, London, UK, 2002. Springer-Verlag.

[22] Ellacoya networks. `http://www.ellacoya.com/`, 2008.

[23] Benny Fallica, Yue Lu, Fernando Kuipers, Rob Kooij, and Piet Van Mieghem. On the quality of experience of sopcast. *Next Generation Mobile Applications, Services and Technologies, International Conference on*, 0:501–506, 2008.

[24] L. Garcés-Erice, E.W. Biersack, P. A. Felber, K. Ross, and G. Urvoy-Keller. Hierarchical peer-to-peer systems. In *Proceedings of ACM/IFIP International Conference on Parallel and Distributed Computing (Euro-Par)*, 2003.

[25] Gnutella. http://gnutella.wego.com, 2006.

[26] GridMedia.
`http://www.gridmedia.com.cn`, 2008.

[27] Krishna P. Gummadi, Stefan Saroiu, and Steven D. Gribble. A measurement study of napster and gnutella as examples of peer-to-peer file sharing systems. *SIGCOMM Computer Communication Review*, 32(1):82, January 2002.

[28] Maya Haridasan and Robbert van Renesse. Defense against intrusion in a live streaming multicast system. In *Proc. 6th International Conference on Peer-to-Peer Computing (P2P2006)*. IEEE, September 2006.

[29] Maya Haridasan and Robbert van Renesse. Defense against intrusion in a live streaming multicast system. In Alberto Montresor, Adam Wierzbicki, and Nahid Shahmehri, editors, *Peer-to-Peer Computing*, pages 185–192. IEEE Computer Society, 2006.

[30] Maya Haridasan and Robbert van Renesse. Securestream: An intrusion-tolerant protocol for live-streaming dissemination. In *Journal of Computer Communications. Special issue on Foundation of Peer-to-Peer Computing.* Elsevier, 2007.

[31] Maya Haridasan and Robbert van Renesse. Securestream: An intrusion-tolerant protocol for live-streaming dissemination. *Computer Communications*, 2008.

[32] Oliver Heckmann, Michael Piringer, Jens Schmitt, and Ralf Steinmetz. Generating Realistic ISP-Level Network Topologies. *IEEE Communications Letters*, 7(7):335–337, July 2003.

[33] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross. Insights into pplive: A measurement study of a large-scale p2p iptv system. In *In Proc. of IPTV Workshop, International World Wide Web Conference*, 2006.

[34] Xiaojun Hei, Chao Liang, Jian Liang, Yong Liu, and Keith W. Ross. A measurement study of a large-scale p2p iptv system. *IEEE Transactions on Multimedia*, 2007.

[35] Xiaojun Hei, Yong Liu, and K. W. Ross. Iptv over p2p streaming networks: the mesh-pull approach. *Communications Magazine, IEEE*, 46(2):86–92, 2008.

[36] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. PRISM: A tool for automatic verification of probabilistic systems. In H. Hermanns and J. Palsberg, editors, *Proc. 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06)*, volume 3920 of *LNCS*, pages 441–444. Springer, 2006.

[37] Qi Huang, Hai Jin, and Xiaofei Liao. P2p live streaming with tree-mesh based hybrid overlay. In *ICPPW '07: Proceedings of the 2007 International Conference on Parallel Processing Workshops*, page 55, Washington, DC, USA, 2007. IEEE Computer Society.

[38] Qi Huang, Hai Jin, Ke Liu, Xiaofei Liao, and Xuping Tu. Anysee2: an auto load balance p2p live streaming system with hybrid architecture. In *InfoScale '07: Proceedings of the 2nd international conference on Scalable information systems*, pages 1–2, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[39] John Jannotti, David K. Gifford, Kirk L. Johnson, M. Frans Kaashoek, and Jr. James W. O'Toole. Overcast: reliable multicasting with on overlay network. In *OSDI'00: Proceedings of the 4th conference on Symposium on Operating System Design & Implementation*, Berkeley, CA, USA, 2000.

[40] Xing Jin, S.-H.G. Chan, W.-P.K. Yiu, Yongqiang Xiong, and Qian Zhang. Detecting malicious hosts in the presence of lying hosts in peer-to-peer streaming. In *IEEE ICME 2006*. IEEE, 2006.

[41] Murat Karakaya, Ibrahim Korpeoglu, and Özgür Ulusoy. Free riding in peer-to-peer networks. *IEEE Internet Computing*, 13(2):92–98, 2009.

[42] David Karger, Eric Lehman, Tom Leighton, Mathhew Levine, Daniel Lewin, and Rina Panigrahy. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web. In *In ACM Symposium on Theory of Computing*, pages 654–663, 1997.

[43] Kazaa. `http://www.kazaa.com`, 2008.

[44] M. Kelaskar, V. Matossian, P. Mehra, D. Paul, and M. Parashar. A study of discovery mechanisms for peer-to-peer applications. In *CCGRID '02: Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid*, page 444, Washington, DC, USA, 2002. IEEE Computer Society.

[45] M. Kwiatkowska, G. Norman, and D. Parker. Prism: Probabilistic model checking for performance and reliability analysis. *ACM SIGMETRICS Performance Evaluation Review*, 36(4):40–45, 2009.

[46] Zhenjiang Li, Yao Yu, Xiaojun Hei, and Danny H. K. Tsang. Towards low-redundancy push-pull p2p live streaming. In *in proceedings of 5th ICST International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine 2008)*, 2008.

[47] Chao Liang, Yang Guo, and Yong Liu. Hierarchically clustered p2p streaming system. In *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, pages 236–241, 2007.

[48] Jian Liang, J Liang, R Kumar, Y Xi, and Keith Ross. Pollution in p2p file sharing systems. In *Proceedings of IEEE Infocom*. IEEE, March 2005.

[49] Jian Liang, Naoum Naoumov, and Keith W. Ross. Efficient blacklisting and pollution-level estimation in p2p file-sharing systems. In *In AINTEC*, 2005.

[50] Yong Liu, Yang Guo, and Chao Liang. A survey on peer-to-peer video streaming systems. *In Peer-to-Peer Networking and Applications, by Springer New York*, 1:18–28, March 2008.

[51] R. Lo Cigno, A. Russo, and D. Carra. On some fundamental properties of p2p push/pull protocols. In *Communications and Electronics, 2008. ICCE 2008. Second International Conference on*, pages 67–73, 2008.

[52] Thomas Locher, Remo Meier, Stefan Schmid, and Roger Wattenhofer. Push-to-Pull Peer-to-Peer Live Streaming. In *21st International Symposium on Distributed Computing (DISC), Lemesos, Cyprus, Springer LNCS 4731*, September 2007.

[53] Keong Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *Communications Surveys & Tutorials, IEEE*, pages 72–93, 2005.

[54] Qin Lv, Pei Cao, Edith Cohen, Kai Li, and Scott Shenker. Search and replication in unstructured peer-to-peer networks. In *ICS '02: Proceedings of the 16th international conference on Supercomputing*, pages 84–95, New York, NY, USA, 2002. ACM.

[55] Qin Lv, Sylvia Ratnasamy, and Scott Shenker. Can heterogeneity make gnutella scalable? In *In Proceedings of the first International Workshop on Peer-to-Peer Systems*, pages 94–103, 2002.

[56] Nazanin Magharei and Reza Rejaie. Mesh or multiple-tree: A comparative study of live p2p streaming approaches. In *in Proceedings of IEEE INFOCOM*, pages 1424–1432, 2007.

[57] Nazanin Magharei, Reza Rejaie, and Yang Guo. Mesh or multiple-tree: A comparative study of live p2p streaming approaches. In *INFOCOM*, pages 1424–1432. IEEE, 2007.

[58] S. Mccanne, S. Floyd, and K. Fall. Network simulator. `http://www-nrg.ee.lbl.gov/ns/`.

[59] Alberto Medina, Anukool Lakhina, Ibrahim Matta, and John Byers. BRITE: Universal topology generation. Technical Report 2001-003, Computer Science Department at Boston University, 1 2001.

[60] J. J. D. Mol, J. A. Pouwelse, D. H. J. Epema, and H. J. Sips. Free-riding, fairness, and firewalls in p2p file-sharing. In *P2P '08: Proceedings of the 2008 Eighth International Conference on Peer-to-Peer Computing*, pages 301–310, Washington, DC, USA, 2008. IEEE Computer Society.

[61] Morpheus. `http://www.morpheus.com`, 2009.

[62] Napster. http://www.napster.com/, 2006.

[63] The new york times, 13 de fevereiro de 2009. `The New York Times Newspaper`, 2009.

[64] Esther Palomar, Juan M. Estevez-Tapiador, Julio C. Hernandez-Castro, and Arturo Ribagorda. A protocol for secure content distribution in pure p2p networks. In *DEXA '06: Proceedings of the 17th International Conference on Database and Expert Systems Applications*, pages 712–716, Washington, DC, USA, 2006. IEEE Computer Society.

[65] B. Pourebrahimi, K. Bertels, and S. Vassiliadis. A survey of peer-to-peer networks. In *Proceedings of the 16th Annual Workshop on Circuits, Systems and Signal Proessing*, 2005.

[66] PPLive. `http://www.pplive.com`, 2008.

[67] PPStreaming. `www.ppstreaming.com`, 2008.

[68] Prism.
`http://www.prismmodelchecker.org`, 2009.

[69] Darshan Purandare and Ratan Guha. An alliance based peering scheme for peer-to-peer live media streaming. In *P2P-TV '07: Proceedings of the 2007 workshop on Peer-to-peer streaming and IP-TV*, pages 340–345, New York, NY, USA, 2007. ACM.

[70] Sylvia Ratnasamy, Paul Francis, Scott Shenker, and Mark Handley. A scalable content-addressable network. In *In Proceedings of ACM SIGCOMM*, pages 161–172, 2001.

[71] M. Ripeanu. Peer-to-peer architecture case study: Gnutella network. In *Peer-to-Peer Computing, 2001. Proceedings. First International Conference on*, pages 99–100, August 2001.

[72] John Risson and Tim Moors. Survey of research towards robust peer-to-peer networks: search methods. *Comput. Netw.*, 50(17):3485–3521, 2006.

[73] Marcus Rocha, Marcelo Maia, Italo Cunha, Jussara Almeida, and Sergio Campos. Scalable media streaming to interactive users. In *In Proc. of ACM Multimedia*, pages 966–975, 2005.

[74] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. *Lecture Notes in Computer Science*, 2218:329–??, 2001.

[75] Stefan Saroiu, Krishna P. Gummadi, and Steven D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Multimedia Computing and Networking (MMCN)*, January 2002.

[76] Thomas Silerston and Olivier Fourmaux. Measuring p2p iptv systems. In *Proceedings of NOSSDAV'07*, June 2007.

[77] Thomas Silverston and Olivier Fourmaux. P2p iptv measurement: a case study of tvants. In *CoNEXT '06: Proceedings of the 2006 ACM CoNEXT conference*, pages 1–2, New York, NY, USA, 2006. ACM.

[78] Thomas Silverston, Olivier Fourmaux, Alessio Botta, Alberto Dainotti, Antonio Pescapé, Giorgio Ventre, and Kavé Salamatian. Traffic analysis of peer-to-peer iptv communities. *Comput. Netw.*, 53(4):470–484, 2009.

[79] Thomas Silverston, Olivier Fourmaux, and Kave Salamatian. Characterization of p2p iptv traffic: Scaling analysis, 2007.

[80] Atul Singh, Miguel Castro, Peter Druschel, and Antony Rowstron. Defending against eclipse attacks on overlay networks. In *EW11: Proceedings of the 11th workshop on ACM SIGOPS European workshop*, page 21. ACM, 2004.

[81] Sopcast. `http://www.sopcast.com`, 2008.

[82] Ion Stoica, Robert Morris, David Karger, Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the 2001 ACM SIGCOMM Conference*, pages 149–160, 2001.

[83] Daniel Stutzbach and Reza Rejaie. Understanding churn in peer-to-peer networks. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 189–202, New York, NY, USA, 2006. ACM.

[84] Daniel Stutzbach, Reza Rejaie, and Subhabrata Sen. Characterizing unstructured overlay topologies in modern p2p file-sharing systems. *IEEE/ACM Trans. Netw.*, 16(2):267–280, 2008.

[85] Y Tang, L Sun, M Zhang, S Yang, and Y Zhong. A novel distributed and practical incentive mechanism for peer to peer live video streaming. In *IEEE International Conference on Multimedia & Expo, Toronto, Canada*, Jul 2006.

[86] Yun Tang, Lifeng Sun, Jian-Guang Luo, and Yuzhuo Zhong. Characterizing user behavior to improve quality of streaming service over p2p networks. In Yueting Zhuang, Shiqiang Yang, Yong Rui, and Qinming He, editors, *PCM*, volume 4261 of *Lecture Notes in Computer Science*, pages 175–184. Springer, 2006.

[87] Hongsuda Tangmunarunkit, Ramesh Govindan, Sugih Jamin, Scott Shenker, and Walter Willinger. Network topologies, power laws, and hierarchy. *SIGCOMM Comput. Commun. Rev.*, 32(1):76–76, 2002.

[88] Hongsuda Tangmunarunkit, Ramesh Govindan, Sugih Jamin, Scott Shenker, and Walter Willinger. Network topology generators: Degree-based vs. structural. In *in ACM SIGCOMM*, pages 147–159, 2002.

[89] Duc A. Tran, Kien Hua, and Tai Do. Zigzag: An efficient peer-to-peer scheme for media streaming. In *Proceedings of IEEE Infocom*, 2003.

[90] Long H Vu, Indranil Gupta, Jin Liang, and Klara Nahrstedt. Measurement and modeling of a large-scale overlay for multimedia streaming. In *QShine: International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, 2007.

[91] Kevin Walsh and Emin Gün Sirer. Fighting peer-to-peer spam and decoys with object reputation. In *Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 138–143. ACM, 2005.

[92] Wenjie Wang, Yongqiang Xiong, Qian Zhang, and Sugih Jamin. Ripple-stream: Safeguarding p2p streaming against dos attacks. In *ICME*. IEEE, 2006.

[93] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, June 1998.

[94] Bernard M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, pages 1617–1622, 1988.

[95] Bernard M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, pages 347–352, 1991.

[96] Chung Kei Wong and Simon S. Lam. Digital signatures for flows and multicasts. *IEEE/ACM Trans. Netw.*, 7(4):502–513, 1999.

[97] Chuan Wu, Baochun Li, and Shuqiao Zhao. Exploring large-scale peer-to-peer live streaming topologies. *ACM Trans. Multimedia Comput. Commun. Appl.*, 4(3):1–23, 2008.

[98] Zhen Xiao and Fan Ye. New insights on internet streaming and iptv. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 645–654, New York, NY, USA, 2008. ACM.

[99] Susu Xie, Bo Li, Gabriel Y. Keung, and Xinyan Zhang. Coolstreaming: Design, theory, and practice. *IEEE Transactions on Multimedia*, 9(8):1661–1671, 2007.

[100] Li Xiong, Ling Liu, and Ieee Computer Society. Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering*, 16:843–857, 2004.

[101] Sirui Yang, Hai Jin, Bo Li, Xiaofei Liao, Hong Yao, and Xuping Tu. The content pollution in peer-to-peer live streaming systems: Analysis and implications. *Parallel Processing, International Conference on*, 0:652–659, 2008.

[102] Youtube. `http://www.youtube.com/`, 2008.

[103] Meng Zhang, Jian-Guang Luo, Li Zhao, and Shi-Qiang Yang. A peer-to-peer network for live media streaming using a push-pull approach. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 287–290. ACM, 2005.

[104] Meng Zhang, Qian Zhang, Lifeng Sun, and Shiqiang Yang. Understanding the power of pull-based streaming protocol: Can we do better? *Selected Areas in Communications, IEEE Journal on*, 25(9):1678–1694, 2007.

[105] Meng Zhang, Li Zhao, Yun Tang, Jian-Guang Luo, and Shi-Qiang Yang. Large-scale live media streaming over peer-to-peer networks through global internet. In *P2PMMS'05: Proceedings of the ACM workshop on Advances in peer-to-peer multimedia streaming*, pages 21–28. ACM Press, 2005.

[106] Xinyan Zhang, Jiangchuan Liu, Bo Li, and Y. S. P. Yum. Coolstreaming/donet: a data-driven overlay network for peer-to-peer live media streaming. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 2102–2111 vol. 3, 2005.

[107] Ben Y. Zhao, Ling Huang, Jeremy Stribling, Sean C. Rhea, Anthony D. Joseph, and John D. Kubiatowicz. Tapestry: A resilient global-scale overlay for service deployment. *Selected Areas in Communications, IEEE Journal on*, 22(1):41–53, 2004.

[108] Li Zhao, Jian-Guang Luo, Meng Zhang, Wen-Jie Fu, Ji Luo, Yi-Fei Zhang, and Shi-Qiang Yang. Gridmedia: A practical peer-to-peer based live video streaming system. In *Multimedia Signal Processing, 2005 IEEE 7th Workshop*, pages 1–4, Oct 2005.

[109] Yipeng Zhou, Dah-Ming Chiu, and John C. S. Lui. A simple model for analyzing p2p streaming protocols. In *ICNP*, pages 226–235. IEEE, 2007.