



Universidade Federal de Minas Gerais

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Arlindo Follador Neto

**Uma Avaliação dos Mecanismos de Transmissão
de Tráfego para *IPTV***

Belo Horizonte

Junho - 2009

Universidade Federal de Minas Gerais
Programa de Pós-Graduação em Engenharia Elétrica

Arlindo Follador Neto

**Uma Avaliação dos Mecanismos de Transmissão
de Tráfego para *IPTV***

Dissertação submetida à banca examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais, como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Engenharia de Computação e Telecomunicações.

Orientador: Prof. Dr. Luciano de Errico

Co-orientador: Prof. Dr. Hani Camille Yehia

Belo Horizonte

Junho - 2009

Dedico este trabalho à toda minha família. Que durante toda a minha vida mantiveram a presença, o apoio e a compreensão inclusive nos momentos mais difíceis.

Agradecimentos

Agradeço ao Prof. Hani Camille Yehia pela confiança, apoio e por todas as orientações, incluindo as acadêmicas ;) Ao Prof. Luciano de Errico pelo paciente acompanhamento de todo o trabalho realizado. À empresa ComunIP que tornou possível essa pesquisa. E finalmente à todos os amigos dos laboratórios CEFALA, LabCOM, GOPAC e MACSIM, sem estes a passagem por esta etapa sem dúvida seria mais difícil e menos divertida.

“... a vida, como a antiga Tebas, tem cem portas. Fechas uma, outras se abrirão. És o último da tua espécie? Virá outra espécie melhor, não feita do mesmo barro, mas da mesma luz...”

Viver!, Várias Histórias
por Machado de Assis

Resumo

A oferta de serviço de IPTV (*IP Television*) é o foco atual de interesse de diversas empresas prestadoras de serviços de telecomunicações. Este serviço consiste na transmissão de programação contínua de TV, estruturada em canais, similar ao serviço de TV aberta ou de TV a cabo, mas transmitido através de uma rede IP. O objetivo é oferecer a mesma qualidade da TV convencional (tela cheia, em média ou alta resolução), acrescida da interatividade da *Internet*. Entre os diversos desafios que esta tecnologia enfrenta está a necessidade de otimizar a distribuição do tráfego de vídeo na rede IP, devido à sua grande demanda por largura de banda. Determinar qual é a melhor tecnologia de distribuição para esta tarefa, utilizando uma rede corporativa, foi o objetivo principal deste trabalho.

Para esse estudo foi desenvolvido um ambiente de rede no simulador ns-2, a partir da topologia e do levantamento do tráfego de uma rede real de provedor. Para as simulações foram injetados *traces* de tráfego de diferentes tipos de vídeo, criando um cenário característico de tráfego IPTV. A fim de eleger o mecanismo de distribuição de tráfego que exigisse da rede menos largura de banda na transmissão dos canais de IPTV, foram realizadas simulações com *unicast*, *multicast* por PIM-DM, *multicast* por PIM-SM e *multicast* por PIM-SSM. Também foi avaliado o desempenho de cada um desses mecanismos de *multicast* no tempo de mudanças de canal (*zapping*). O melhor resultado foi obtido com o PIM-SSM (*Protocol Independent Multicast - Source-Specific Multicast*), que demandou menor carga da rede para a transmissão do tráfego IPTV e permitiu o menor tempo de *zapping* (0, 2 s). A não disponibilidade de implementações P2P (*Peer to Peer*) para o simulador ns-2 impossibilitou a realização de simulações semelhantes para estes mecanismos. Contudo, um estudo teórico foi realizado, mostrando a superioridade do *multicast* sobre o P2P, para o cenário de IPTV.

Os resultados sugerem a possibilidade de implementação de serviços IPTV nas redes IP corporativas atuais, apesar da largura de banda exigida, desde que utilizados mecanismos de otimização da distribuição de tráfego.

Abstract

The IPTV service offer is currently the main interest of many ISPs (Internet Service Providers). Such service comprises the continuous transmission, via an IP network, of TV programs structured into channels, like the open TV ou cable TV services. The goal is to reach the same quality of conventional TV (full screen, medium or high definition), adding the interactivity of the Internet. Among the challenges, there is the need of optimizing the video traffic distribution in the IP network, given its heavy bandwidth demand. Finding which is the best distribution technology for this purpose, in a corporative network, was the main objective of the present work.

A network was implemented with the ns-2 network simulator, using real world topology and traffic from an ISP. Simulations used different video traffic traces, characterizing an IPTV scenario. To identify the traffic distribution mechanism that imposed less bandwidth burden during IPTV transmission, the work tested different approaches using unicast and multicast (with PIM-DM, PIM-SM, and PIM-SSM). Additionally, the zapping (channel change time) performance of each multicast approach was evaluated. The best result was achieved by the PIM-SSM (Protocol Independent Multicast - Source-Specific Multicast), which imposed a reduced load in the network and produced the smaller zapping time (0.2 s). As no P2P (Peer to Peer) implementation was available for the ns-2, this mechanism could not be simulated. However, a teoretical study showed that multicast is superior to P2P, for an IPTV scenario.

The results suggested that, using proper traffic distribution mechanisms, the implementation of IPTV services on current corporative IP networks is feasible, in spite of the required bandwidth.

Lista de Figuras

2.1	Arquitetura de um sistema multimídia IPTV.	7
2.2	Funcionamento do método de transmissão <i>unicast</i>	9
2.3	Funcionamento do método de transmissão <i>broadcast</i>	10
2.4	Funcionamento do método de transmissão <i>multicast</i>	11
2.5	Mapeamento de IP para MAC <i>multicast</i>	14
2.6	Delimitação dos protocolos PIM e IGMP em uma rede com suporte multicast.	18
2.7	Exemplo de comunicação entre <i>peers</i> P2P.	19
2.8	Funcionamento do método de transmissão P2P.	20
2.9	P2P com servidor centralizado (<i>Napster</i>).	22
2.10	P2P com método descentralizado (<i>Gnutella</i>).	23
2.11	Rede de sobreposição hierárquica P2P (<i>KaZaA</i>).	25
3.1	Conteúdo do arquivo de <i>trace</i> do ns-2.	32
3.2	Exemplo de custos em uma rede de distribuição simples utilizando <i>unicast</i>	35

3.3	Exemplo de custos em uma rede de distribuição simples utilizando <i>multicast</i>	36
3.4	Exemplos de custos em redes de distribuição simples utilizando P2P.	36
3.5	Topologia de rede com três fontes e um gateway (<i>screenshot</i> do ns-2 com as capacidades iniciais estabelecidas).	37
3.6	Perfil de tráfego coletado na rede do Grupo Mega.	39
3.7	Variação do Bit Rate para o filme.	41
3.8	Variação do Bit Rate para o jogo de futebol.	42
3.9	Variação do Bit Rate para o jornal.	42
4.1	Total de pacotes na transmissão (simulação com um único programa e protocolos <i>unicast</i> e <i>multicast</i>).	45
4.2	Total de bytes na transmissão (simulação com um único programa e protocolos <i>unicast</i> e <i>multicast</i>).	46
4.3	Total de pacotes perdidos na transmissão (simulação com um único programa e protocolos <i>unicast</i> e <i>multicast</i>).	46
4.4	Soma cumulativa do tráfego na transmissão em função do tempo na simulação com um único programa e protocolos <i>unicast</i> e <i>multicast</i>	47
4.5	Vazão da rede na simulação com um único programa e protocolos <i>unicast</i> e <i>multicast</i>	48
4.6	Total de pacotes na transmissão (simulação com três programas ocupando a rede simultaneamente).	49
4.7	Total de bytes na transmissão (simulação com três programas ocupando a rede simultaneamente).	50

4.8	Soma cumulativa do tráfego da transmissão em função do tempo na simulação com três programas ocupando a rede simultaneamente.	50
4.9	Vazão da rede na simulação com três programas ocupando a rede simultaneamente. .	51
4.10	Caminho tomado pelas transmissões <i>multicast</i> na topologia simulada.	52
4.11	Total de pacotes perdidos na simulação com três programas ocupando a rede simultaneamente.	53
4.12	Vazão entre os nós 4 e 8 com a soma dos FIDs relativos aos três programas simulados.	54
4.13	Topologia com a linha do tempo de início e fim de cada programa.	55
4.14	Tempo de <i>zapping</i> na simulação com <i>multicast</i> PIM-DM.	56
4.15	Tempo de <i>zapping</i> na simulação com <i>multicast</i> PIM-SM.	57
4.16	Tempo de <i>zapping</i> na simulação com <i>multicast</i> PIM-SSM.	57

Lista de Tabelas

3.1	Estatísticas dos programas utilizados para simulação.	41
4.1	Tempo de <i>zapping</i> para os protocolos <i>multicast</i>	58
4.2	Valores obtidos para o instante de tempo de chegada do primeiro pacote referente a um programa entregue após a requisição <i>join IGMP</i>	59
4.3	Pontuação na escala MOS segundo tempo de <i>zapping</i>	60
4.4	Avaliação da escala MOS segundo pontuação obtida pelo tempo de <i>zapping</i>	60

Lista de Abreviaturas

AS	<i>Autonomous System</i>
BASH	<i>Born Again SHell</i>
CBR	<i>Constant Bit Rate</i>
CONSER	<i>Collaborative Simulation for Education and Research</i>
DARPA	<i>Defense Advanced Research Projects Agency</i>
DRM	<i>Digital Rights Management</i>
DVMRP	<i>Distance Vector Multicast Routing Protocol</i>
FTP	<i>File Transfer Protocol</i>
GNU	<i>GNU is Not Unix</i>
HD	<i>High Definition</i>
HDTV	<i>High Definition TeleVision</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IANA	<i>Internet Assigned Numbers Authority</i>
ICMP	<i>Internet Control Message Protocol</i>
ICSI	<i>International Computer Science Institute</i>
IDE	<i>Integrated Development Environment</i>
IGMP	<i>Internet Group Management Protocol</i>
IP	<i>Internet Protocol</i>
IPTV	<i>Internet Protocol TeleVision</i>
IPv4	<i>Internet Protocol version 4</i>
IPv6	<i>Internet Protocol version 6</i>
ISDN	<i>Integrated Services Digital Network</i>
LAN	<i>Local Area Network</i>

LBL	<i>Lawrence Berkeley National Laboratory</i>
MAC	<i>Medium Access Control</i>
MLD	<i>Multicast Listener Discover</i>
MOS	<i>Mean Opinion Score</i>
MOSPF	<i>Multicast Extensions for Open Shortest-Path First</i>
ns-2	<i>network simulator 2</i>
NSF	<i>National Science Foundation</i>
OSPF	<i>Open Shortest-Path First</i>
OTCL	<i>Object-oriented Tool Command Language</i>
P2P	<i>Peer to Peer</i>
PARC	<i>Palo Alto Research Center</i>
PIM	<i>Protocol Independent Multicast</i>
PIM-DM	<i>Protocol Independent Multicast - Dense Mode</i>
PIM-SM	<i>Protocol Independent Multicast - Sparse Mode</i>
PIM-SSM	<i>Protocol Independent Multicast - Source-Specific Multicast</i>
QoE	<i>Quality of Experience</i>
QoS	<i>Quality of Service</i>
RIP	<i>Routing Information Protocol</i>
RP	<i>Rendezvous Point</i>
SAMAN	<i>Simulation Augmented by Measurement and Analysis for Networks</i>
SD	<i>Standart Definition</i>
TCL	<i>Tool Command Language</i>
TCP	<i>Transmission Control Protocol</i>
TKN	<i>Telecommunication Networks Group</i>
TTL	<i>Time-To-Live</i>
TV	<i>TeleVision</i>
UDP	<i>User Datagram Protocol</i>
VBR	<i>Variable Bit Rate</i>
VoD	<i>Video on-Demand</i>

VoIP *Voice over Internet Protocol*

WISP *Wireless Internet Service Provider*

Sumário

1	Introdução	1
1.1	Objetivos	3
1.2	Estrutura do Trabalho	4
2	Mecanismos de Transmissão de Tráfego para IPTV	5
2.1	IPTV (<i>Internet Protocol TeleVision</i>)	6
2.2	Comunicação <i>unicast</i>	8
2.3	Comunicação <i>broadcast</i> por difusão	9
2.4	Comunicação <i>multicast</i>	11
2.4.1	Modelo <i>multicast</i> de serviço	12
2.4.2	Iniciando o tráfego <i>multicast</i>	12
2.4.3	Endereçamento <i>multicast</i>	13
2.4.4	Protocolos de roteamento <i>multicast</i>	15
2.4.5	Protocolos de gerenciamento <i>multicast</i>	17

2.5	Comunicação P2P	19
2.5.1	P2P centralizado	21
2.5.2	P2P distribuído	22
2.5.3	P2P hierárquico	24
2.6	Distribuição de Tráfego IPTV: estado da arte	26
3	Metodologia	30
3.1	Ambiente	30
3.2	Simulador de Rede ns-2	31
3.3	Simulação de protocolos no ns-2	33
3.3.1	PIM-SSM no Network Simulator (ns-2)	33
3.3.2	Protocolo P2P	34
3.4	Topologia e ocupação da rede	37
3.5	Geração de tráfego IPTV	40
3.6	Experimentos	41
4	Resultados e Discussão dos Resultados	44
4.1	Resultados para simulação com um único programa e protocolos <i>unicast</i> e <i>multicast</i>	44
4.2	Resultados para simulação com três programas ocupando a rede simultaneamente	48
4.3	Resultados para <i>zapping</i> em simulação com três programas	53

4.4	Protocolos P2P	61
4.5	Considerações finais	62
5	Conclusões	64
5.1	Trabalhos Futuros	69
	Referências Bibliográficas	71

Capítulo 1

Introdução

A transmissão de vídeo digital de alta qualidade em tempo real em redes IP, seja na *Internet* ou em redes corporativas, é o foco atual de interesse de diversas empresas, tanto provedores de *Internet* (ISPs - *Internet Service Providers*) quanto operadoras de redes de telecomunicações. Adicionando este serviço aos já existentes serviços de dados (*e-mail*, *web*, acesso à *Internet*, dentre outros) e serviços de voz (*VoIP*, *Telefonia IP*), as operadoras passariam a oferecer um serviço *triple play*: dados, voz e vídeo, através do mesmo acesso em banda larga (Yarali and Cherry, 2005). Este cenário concretiza uma antiga meta em redes de telecomunicações, que é a de tornar possível ao usuário obter diferentes serviços de comunicação a partir de um único ponto de acesso, conceito tentado anteriormente com a ISDN (*Integrated Services Digital Network*) (ITU-T, 1991), em suas versões banda estreita e banda larga.

O serviço *triple play* ganha mais força com o surgimento do conceito de IPTV (*Internet Protocol TeleVision*). Entende-se aqui por IPTV a transmissão de programação contínua de TV, estruturada em canais da mesma forma que os serviços de TV aberta e de TV a cabo, através da *Internet* ou em redes corporativas (Shihab and Cai, 2007). Essa tecnologia deve oferecer a mesma qualidade da TV convencional (vídeo em tela cheia, em média ou em alta resolução), além de permitir novas funções, como uma maior interação com os usuários devido à utilização do protocolo IP.

O desenvolvimento das tecnologias de redes IP tornou possível a criação de uma infra-estrutura capaz de suportar aplicações multimídia, tais como mídia contínua, vídeo, voz, videoconferência e ambientes virtuais. Tais aplicações possuem requisitos diferentes das aplicações tradicionais, tais como os serviços de *web* (texto e imagem), a transferência de arquivos e o *e-mail* (Cardozo, 2002). Mais especificamente, a transmissão de vídeo através de redes IP exige grande demanda por largura de banda. Atualmente os serviços de *Video on Demand* (VoD), mesmo utilizando uma resolução baixa, são responsáveis por um aumento considerável na ocupação de redes de distribuição (Uzunalioglu, 2009). No IPTV, a demanda por largura de banda de uma programação que por ventura utilize HDTV (*High Definition TeleVision*) aumenta aproximadamente 25 vezes em relação à das aplicações multimídias atuais, (Yarali and Cherry, 2005; Gill et al., 2007). Considera-se HD (*High Definition*) o serviço de TV que oferece imagem de 1080 linhas na razão de aspecto 16:9 (widescreen), superior ao padrão tradicional de SD (*Standard Definition*) de 525 ou 625 linhas na razão de aspecto 4:3 (?).

Introduzir sem otimização um tráfego (MPEG-2) que pode variar entre 3 (em SDTV) e 9 Mbps (em HDTV) (Yarali and Cherry, 2005) nas atuais redes IP, que apenas suportam serviços *triple-play* com fluxos VoD de baixa resolução de cerca de 394 Kbps (Gill et al., 2007), pode causar o congestionamento tanto da fonte quanto da rede de distribuição. Realizar a entrega de serviços de TV em redes IP de forma escalável, com qualidade aceitável, é um grande desafio devido ao alto requisito por largura de banda. Uma das soluções adotadas tem sido a distribuição de fluxos de vídeo por múltiplas conexões *unicast*, uma para cada um dos clientes requisitantes (Gill et al., 2007). Este tipo de solução, obviamente não escalável, tende a causar o congestionamento da rede de distribuição, pois com a popularização do serviço e, conseqüentemente, a requisição deste por diversos novos clientes, a carga ¹ na rede atinge valores consideráveis.

Além disso, a implantação de serviços IPTV apresenta outros desafios operacionais para as operadoras e provedores, tais como: a escolha adequada dos codificadores; a gerência de direitos digitais (DRM - *Digital Rights Management*); a implementação de uma infra-estrutura adequada de rede; o uso de soluções de segurança; a garantia de eficiência do serviço IPTV; a entrega de conteúdo aos usuários finais com uma alta qualidade de experiência (QoE - *Quality of Experience*).

¹Nesta dissertação a palavra “carga” será utilizada para descrever o quanto uma aplicação multimídia, vídeo digital ou transmissão demanda de largura de banda de uma rede de computadores.

Com o propósito de otimizar a transmissão de programas ² de um sistema IPTV em uma rede de distribuição, diversas técnicas estão sendo aplicadas com a expectativa de redução da carga imposta na rede durante as transmissões. A distribuição *unicast*, *multicast* e P2P são os principais mecanismos considerados em estudos que visam otimizar as transmissões em um sistema IPTV (Gill et al., 2007; She et al., 2009; Uzunalioglu, 2009). A literatura mostra diversas implementações que otimizam a transmissão de tráfego IPTV em redes de distribuição reais (Komosny and Novotny, 2007; She et al., 2009; Begen et al., 2009; Gallo et al., 2009; Hei et al., 2008; Tian et al., 2008). Muitas destas otimizações trabalham com a *Internet* como o cenário de rede. No entanto, nessa dissertação o cenário é uma rede corporativa, inspirada na rede de distribuição da empresa Mega Grupo, que provê acesso à sub-redes para o atendimento de clientes.

1.1 Objetivos

O principal objetivo deste trabalho foi determinar, dentre os principais mecanismos disponíveis para transmissão de tráfego IPTV em uma rede IP corporativa, qual seria o que melhor reduz a carga imposta à rede. Uma rede IP corporativa representa aqui uma rede de um provedor de *Internet* ou de uma operadora de telecomunicações. Para tanto foram considerados os seguintes mecanismos de transmissão:

- *unicast*;
- *Protocol Independent Multicast - Dense Mode* (PIM-DM) ;
- *Protocol Independent Multicast - Sparse Mode* (PIM-SM);
- *Protocol Independent Multicast - Source-Specific Multicast* (PIM-SSM);
- *peer-to-peer* (P2P).

²A palavra “programas” nesta dissertação refere-se a programas televisivos (filmes, jornais, jogos, dentre outros).

Através de um ambiente de simulação desenvolvido com base nas características (topologia e ocupação) de uma rede de distribuição real, o desempenho desses mecanismos foi avaliado em diferentes cenários. Desejou-se investigar quais dos métodos relacionados de fato otimizaram a carga. As avaliações visaram medir a demanda por carga, o atraso e a qualidade nas transmissões com diferentes mecanismos. Bases de informações sobre a demanda por carga de programas reais num sistema IPTV foram utilizadas, a fim de aproximar as simulações da realidade (Lanfranchi and Bing, 2008).

Esta dissertação foi desenvolvida dentro do escopo do projeto de pesquisa intitulado “*Desenvolvimento de um Sistema de Vídeo sobre IP: Codificação e Arquitetura.*” (?). O projeto, resultado da parceria entre a UFMG e a empresa mineira ComunIP, objetivou estudar os métodos de aquisição, compressão e distribuição em rede IP de conteúdo televisivo em média e alta definição, a fim de implementar um sistema IPTV.

1.2 Estrutura do Trabalho

Este trabalho apresenta mais quatro capítulos. No Capítulo 2 é feita uma revisão bibliográfica sobre IPTV, protocolos de distribuição *unicast*, *multicast* e P2P, e sobre o estado da arte da distribuição de tráfego IPTV em redes.

No Capítulo 3 são apresentadas as metodologias para a utilização e a simulação dos protocolos. São detalhadas a forma de geração de tráfego, o simulador usado, a topologia da rede e a argumentação teórica da não adoção de protocolos P2P.

A avaliação do desempenho da metodologia é feita nos Capítulos 4 e 5. No Capítulo 4 são apresentados e discutidos os resultados de desempenho obtidos para cada mecanismo nas diferentes simulações. As impressões gerais e conclusões estão disponíveis no Capítulo 5.

Capítulo 2

Mecanismos de Transmissão de Tráfego para IPTV

Este capítulo abordará o sistema IPTV, do inglês *Internet Protocol TeleVision*, e os diferentes métodos de transmissão em redes de computadores que podem ser utilizados para otimizar a distribuição de conteúdo neste sistema. Tais métodos de transmissão diferenciam-se principalmente na taxa de ocupação da rede. Alguns tipos de fluxos de dados, como por exemplo *streams* de vídeo, requerem muita largura de banda. Neste caso é importante que o método de transmissão utilizado seja otimizado o bastante para sobrecarregar o mínimo possível a rede. Do contrário, isto pode causar o congestionamento e atrasos na rede de distribuição.

Os modelos de transmissão em rede mais comuns e que serão utilizados neste trabalho são:

- *unicast*: modelo básico de transmissão, no qual uma única origem transfere dados individualmente para cada destino;
- *broadcast*: uma única origem transfere dados para todos os destinos em uma rede, mesmo que estes não os solicitem;
- *multicast*: uma única origem transfere dados para múltiplos destinos em uma rede;

- P2P (*peer-to-peer*): uma transmissão colaborativa entre nós em uma rede. Cada nó transfere parte dos dados que recebe de uma origem ou nó para um outro nó na rede (Rocha et al., 2004).

As próximas seções apresentam mais detalhadamente: o sistema IPTV (Seção 2.1); os métodos de transmissão citados e suas técnicas de otimização de tráfego (Seções 2.2, 2.3, 2.4 e 2.5); e, ao final, um levantamento sobre o desenvolvimento atual de métodos para otimização da distribuição de tráfego num sistema IPTV (Seção 2.6).

2.1 IPTV (*Internet Protocol TeleVision*)

A transmissão de vídeo em redes IP pode ser realizada de diferentes formas. A mais conhecida delas, o VoD (*Video on Demand*), caracteriza um serviço no qual a reprodução do conteúdo de vídeo é feita no momento escolhido pelo usuário. O conteúdo pode então ser descarregado totalmente no computador (*download*) ou ser exibido à medida em que é transmitido (*streaming*). O exemplo mais popular deste tipo de serviço é o YouTube ¹, que atualmente oferece *streaming* de vídeos em janelas pequenas com baixa e média resolução com uma largura de banda média de aproximadamente 394 Kbps (Gill et al., 2007).

O IPTV, segundo Lee (2007), é um sistema multimídia no qual canais de TV digital são disponibilizados através de uma rede IP, mantendo um nível satisfatório de QoS ² (*Quality of Service*) e QoE ³ (*Quality of Experience*). Uma rede de distribuição que atenda seus clientes com essa modalidade de serviço, acrescida dos serviços convencionais de dados e voz, é conhecida como rede *triple play*.

A diferença fundamental entre o IPTV e as demais formas de *download* ou *streaming* de vídeo em redes IP, é que o IPTV oferece um serviço de transmissão contínua de canais de TV por uma rede

¹YouTube - www.youtube.com.br: site gratuito que oferece *streaming* de vídeos.

²QoS (*Quality of Service*) refere-se a uma vasta coleção de tecnologias e técnicas em redes de computadores. Tais tecnologias e técnicas incluem frequentemente agir sobre a disponibilidade do serviço, banda, latência e a taxa de erro, a fim de prover garantias sobre a capacidade de uma rede na entrega de serviços.

³QoE (*Quality of Experience*) refere-se à percepção do usuário sobre a qualidade de um serviço. Exprime-se em parâmetros subjetivos como “excelente”, “bom”, “regular”, “ruim” e “péssimo”.

IP. No IPTV, como pode ser visto na Figural 2.1, existe uma rede de distribuição corporativa que liga o provedor de serviço IPTV ao cliente final (Lee, 2007).

O **Provedor de Conteúdo**, representado por um ou mais produtores de vídeo, é o responsável pela gravação, compressão e distribuição da programação televisiva para o **Provedor de Serviço**. Esse por sua vez, tem por função armazenar e retransmitir esse conteúdo para o **Consumidor**, através da **Rede de Distribuição**. A **Rede de Distribuição**, foco central deste estudo, é a responsável por otimizar a transmissão feita a partir do **Provedor de Serviço** para todos os **Consumidores**.

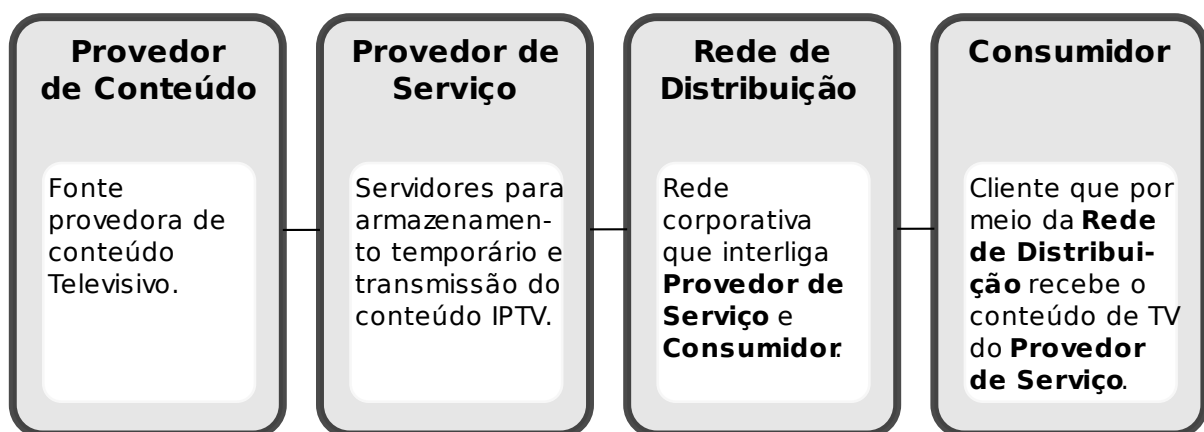


Figura 2.1: Arquitetura de um sistema multimídia IPTV.

De acordo com o apresentado por Yarali and Cherry (2005), uma programação produzida na qualidade HDTV (*High Definition TeleVision*) pode demandar da **Rede de Distribuição** uma largura de banda de até 9 Mbps. No levantamento feito por Gill et al. (2007), a média de largura de banda utilizada nos vídeos acessados no YouTube, maior site de VoD existente atualmente, é de 394 Kbps. Logo, o aumento da carga imposta em uma rede distribuição pelo IPTV é de aproximadamente 20 vezes se comparado com o VoD atual.

Mesmo com uma qualidade reduzida e, conseqüentemente, a largura de banda também reduzida se comparada à necessária para um conteúdo HDTV em rede IP, sites que oferecem conteúdo VoD (a exemplo do YouTube) também acabam demandando muita largura de banda. Por oferecerem seu conteúdo em forma de VoD, são criados canais exclusivos de transmissão entre a fonte e cada cliente. Isso faz com que a necessidade de largura de banda para um determinado conteúdo seja multiplicada pelo número de clientes interessados (Gill et al., 2007).

Em um sistema IPTV, se o método de transmissão fosse semelhante ao utilizado em serviços de VoD, rapidamente a **Rede de Distribuição** seria congestionada. No entanto, no sistema IPTV, assim como na TV, o programa transmitido para um **Consumidor** é o mesmo que o transmitido simultaneamente para todos os demais **Consumidores**. Isso permite a otimização da carga imposta na **Rede de Distribuição**, durante a transmissão entre **Provedor de Serviço** e **Consumidor**, devido ao comportamento multidestinatário de cada programa.

A transmissão de conteúdo televisivo em redes IP também traz desafios que vão além da demanda por largura de banda. A fim de otimizar as transmissões, um cliente recebe apenas o fluxo referente ao canal que selecionou, diferente da TV aberta, onde o equipamento se ajusta para reproduzir apenas um canal mas esse recebe o fluxo de todos os canais disponíveis. A mudança entre canais (*zapping*) em um sistema IPTV implica na solicitação de cancelamento da inscrição feita em um fluxo para a inscrição em outro (Sasaki et al., 2008). O cliente pára de receber o fluxo do canal atual para dar início ao recebimento de um novo fluxo, reproduzindo assim a programação de um novo canal. O tempo de *zapping* corresponde ao tempo que se leva para trocar de um canal para outro em um sistema IPTV. A redução no tempo de *zapping* dará ao cliente uma melhor percepção (QoE) do serviço de IPTV.

Dada as características de tempo real e multidestinatários das transmissões num sistema IPTV, como visto em Kukhmay et al. (2006), vários métodos de transmissão podem ser utilizados a fim de otimizar essa distribuição, reduzindo a carga imposta na rede e reduzindo o tempo de *zapping*. Tais métodos serão descritos a seguir.

2.2 Comunicação unicast

O método *unicast* pode ser utilizado na comunicação de uma fonte para um cliente. A fonte cria um canal de comunicação individual para cada cliente. Se a transmissão é feita entre uma fonte e três clientes, a fonte enviará três fluxos idênticos, uma para cada cliente (Tanenbaum, 1997).

Um exemplo de implementação do mecanismo *unicast* é o TCP (*Transmission Control Protocol*). O TCP é um protocolo de comunicação sobre o qual assenta a maioria das comunicações utilizadas

na *Internet*. Uma das características do TCP é a orientação à conexão, ou seja, antes de qualquer transmissão é feita uma conexão entre fonte e cliente. Essa conexão é utilizada para controlar e assegurar que os pacotes que são enviados chegarão ao destino. O protocolo TCP especifica três fases durante uma conexão: estabelecimento, transferência de dados e término. Essas fases acontecem sempre entre uma única fonte e um único cliente, o que garante outra característica do TCP, uma conexão TCP é estabelecida entre dois pontos, ou seja, utilizando o mecanismo (*unicast*).

Um exemplo do funcionamento do método de distribuição *unicast* pode ser visto na Figura 2.2. Analisando a figura, é possível perceber que nesse método de distribuição há uma alta carga de tráfego entre a **Fonte** e o **Roteador**. Os fluxos entre **Fonte** e **Roteador** aumentam linearmente com o número de clientes.

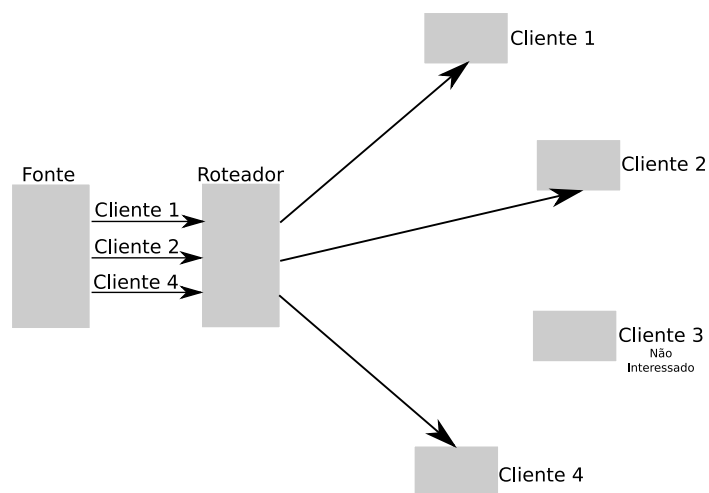


Figura 2.2: Funcionamento do método de transmissão *unicast*.

2.3 Comunicação broadcast por difusão

Em algumas aplicações, os *hosts*⁴ precisam enviar mensagens a muitos outros *hosts* (ou a todos). Um exemplo são as distribuições de relatórios sobre o tempo, atualizações do mercado de ações, programas de rádio e televisão ao vivo. Tais aplicações poderiam funcionar melhor se a difusão fosse

⁴Um **host** é um computador, ou outro tipo de unidade de processamento, ligado a uma rede.

feita para todas as máquinas, permitindo que aquelas que estivessem interessadas lessem os dados. O envio de um pacote a todos os destinos simultaneamente é chamado de *broadcasting* ou difusão.

O ARP (*Address Resolution Protocol*) é um exemplo do uso do mecanismo *broadcast*. O ARP é um protocolo essencial para o funcionamento de redes de computadores. Ele é usado para encontrar o endereço MAC (*Media Access Control*) de uma estação a partir de um endereço IP desta.

Um endereço MAC diz respeito ao endereço físico da interface de rede de um *host*, composto de 48 *bits* representados em formato hexadecimal. O MAC é responsável pelo controle de acesso de cada estação à rede *Ethernet*⁵.

A Figura 2.3 ilustra o funcionamento da distribuição em *broadcasting*. Assim como acontece no mecanismo de difusão chamado *encaminhamento pelo caminho inverso*, visto em Tanenbaum (1997), a **Fonte** encaminha um pacote para o **Roteador**, que o retransmite para todos os clientes. No método *broadcasting*, os pacotes são enviados inclusive para as estações que não estão interessadas em recebê-los, como é o caso do **Cliente 3**.

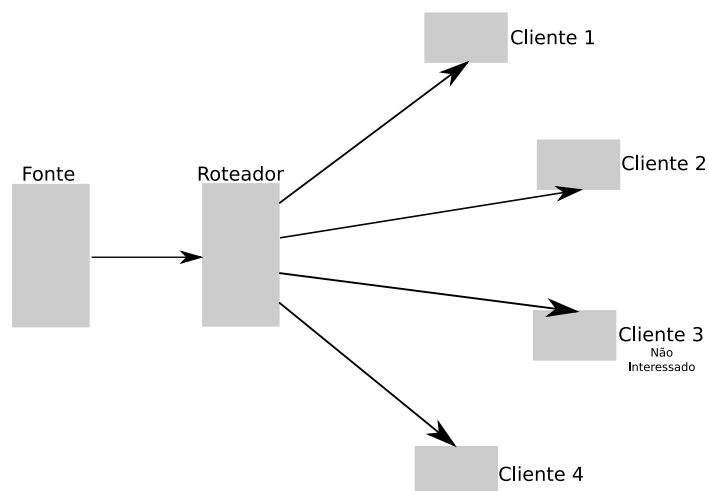


Figura 2.3: Funcionamento do método de transmissão *broadcast*.

⁵*Ethernet* é uma tecnologia de rede local. Padronizado pelo IEEE como 802.3, a *Ethernet* define o cabeamento e os sinais elétricos para a Camada Física e o formato dos pacotes e os protocolos para a Camada de Enlace do modelo OSI (*Open Systems Interconnection*).

2.4 Comunicação multicast

Aplicações como rádio digital, vídeo-conferência, TV, entre outras, pertencem à categoria de comunicação de grupo, diferente das demais aplicações que utilizam comunicação ponto-a-ponto. Essas aplicações podem ter várias fontes e milhões de clientes. Um serviço *multicast* é importante pois reduz a ocupação na rede, transmitindo fluxos apenas para os clientes interessados. Isto é diferente de múltiplos fluxos, como visto no *unicast*, ou de fluxos para todos os clientes indiscriminadamente, como no *broadcast* (Tanenbaum, 1997).

Por enviar um fluxo para cada cliente interessado, o método *unicast* tende a consumir rapidamente todos os recursos da rede e a capacidade de processamento dos servidores (Figura 2.2). Já o *broadcast* faz o envio indiscriminado dos dados, fazendo com que até os clientes não interessados recebam o tráfego. Isso também põe em risco a confiabilidade da transmissão, pois os dados ficam acessíveis a qualquer cliente da rede (Figura 2.3). O *multicast*, apresentado na Figura 2.4, é um mecanismo de distribuição que une as qualidades do *unicast* e do *broadcast*, fazendo com que a **Fonte** envie apenas um fluxo para o **Roteador** e este o replique somente para os **Clientes** interessados.

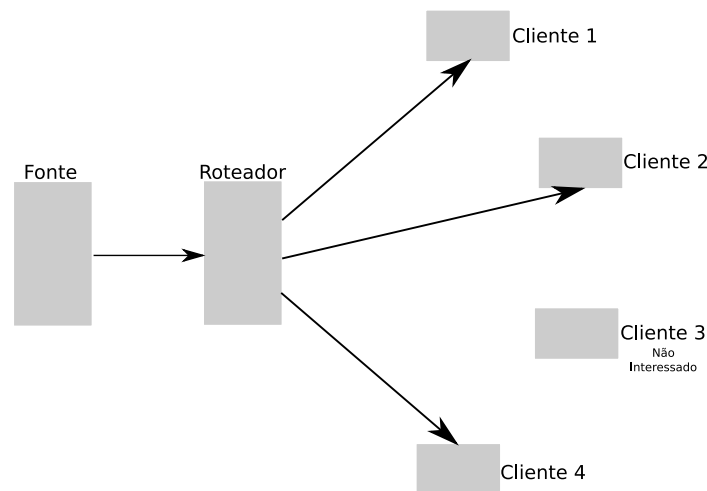


Figura 2.4: Funcionamento do método de transmissão *multicast*.

O IP *multicast* implementa seu modelo de distribuição utilizando a camada de rede, sendo composto por um modelo de serviço, pelo gerenciamento de grupo e por protocolos de roteamento (Diot et al., 1997). Esta seção apresenta os conceitos fundamentais do IP (*Internet Protocol*) *multicast*.

2.4.1 Modelo *multicast* de serviço

No modelo *multicast* é definido um grupo aberto, onde:

- qualquer estação pode participar sem necessidade de autorização;
- uma estação pode pertencer a diversos grupos *multicast* existentes;
- um servidor *multicast* pode transmitir dados para determinado grupo sem fazer parte deste;
- os grupos são dinâmicos (estações podem entrar ou sair de determinados grupos a qualquer momento);
- as informações dos membros de um grupo, tais como número e identidade, não são conhecidos nem pelo servidor *multicast*, nem pelas estações.

2.4.2 Iniciando o tráfego *multicast*

Como visto em Costa and Duarte (2003) e seções anteriores, para iniciar um tráfego *multicast* a fonte do tráfego que detém um endereço *multicast* temporário escolhe um endereço *multicast* permanente e passa a anunciar sua transmissão. No momento em que os roteadores começam a receber os dados do novo endereço *multicast*, estes mudam de estado (estado de *forward*) e uma árvore para distribuição em grupo *multicast* é criada.

Existem diferentes protocolos para criação da árvores de distribuição de tráfego *multicast*. Exemplos desses protocolos são o DVMRP (*Distance Vector Multicasting Routing Protocol*), um protocolo de roteamento por fonte, o PIM-SM (*Protocol Independent multicast - Sparse Mode*), um protocolo de roteamento com árvores centralizadas (Tanenbaum, 1997). Após a criação da árvore de distribuição, se não houver clientes para o grupo criado, no DVMRP a distribuição é limitada no roteador correspondente, exatamente no primeiro salto da árvore. No PIM-SM a distribuição é limitada por um roteador central, o RP (*Rendezvous Point*).

Uma estação interessada em uma transmissão de um determinado grupo deve explicitar seu desejo de receber os dados desse grupo ao seu roteador vizinho, por meio de uma mensagem IGMP (*join*). Essa mesma estação pode deixar de receber a transmissão de um grupo por meio de uma mensagem (*prune*) IGMP, que deve ser enviada, também, para seu roteador vizinho.

Um tráfego *multicast* é interrompido na rede quando a fonte pára de enviar dados para o endereço *multicast*. Os roteadores que fazem parte da árvore de distribuição *multicast*, mantêm apenas informação volátil (*soft-states*), ou seja, descartam a informação de um grupo de transmissão pré-existente, caso sua atividade cesse, de acordo com o *timeout* dos temporizadores existentes em cada roteador, desfazendo gradativamente a árvore de distribuição *multicast*.

2.4.3 Endereçamento multicast

No momento em que uma aplicação sinaliza à camada de rede a necessidade de se conectar a determinado grupo *multicast*, é verificado primeiramente se a estação já está conectada ao grupo. Em caso negativo, uma mensagem IGMP é enviada à rede local (um roteador vizinho) solicitando o ingresso da estação no grupo *multicast*. Além dessa solicitação, o endereço de rede IP *multicast* é utilizado para criar um endereço MAC (*Medium Access Control*) correspondente. O endereço MAC é *multicast* quando o *bit* menos significativo do *byte* mais significativo é igual a 1. Dessa forma a estação é reprogramada a aceitar pacotes *multicast* enviados para o grupo.

O novo endereço MAC é formado por uma operação lógica “ou” entre os 23 *bits* menos significativos do endereço MAC atual (ex. 01.00.5E.00.00.00) e os 23 *bits* menos significativos do IP *multicast* (ex. 244.0.2.2). O resultando desta operação gera o MAC 01.00.5E.00.02.02, como pode ser visto na Figura 2.5. Caso a diferenciação seja apenas nos 9 *bits* mais significativos do IP *multicast*, todos os IPs serão representados em um único MAC (Costa and Duarte, 2003).

Segundo (Tanenbaum, 1997), o IP aceita distribuição em *multicast* utilizando endereços da classe D que identificam um grupo de *hosts*. Estão disponíveis 28 *bits* para identificar grupos, portanto, podem haver mais de 250 milhões de grupos ao mesmo tempo. Os endereços designados à classe D

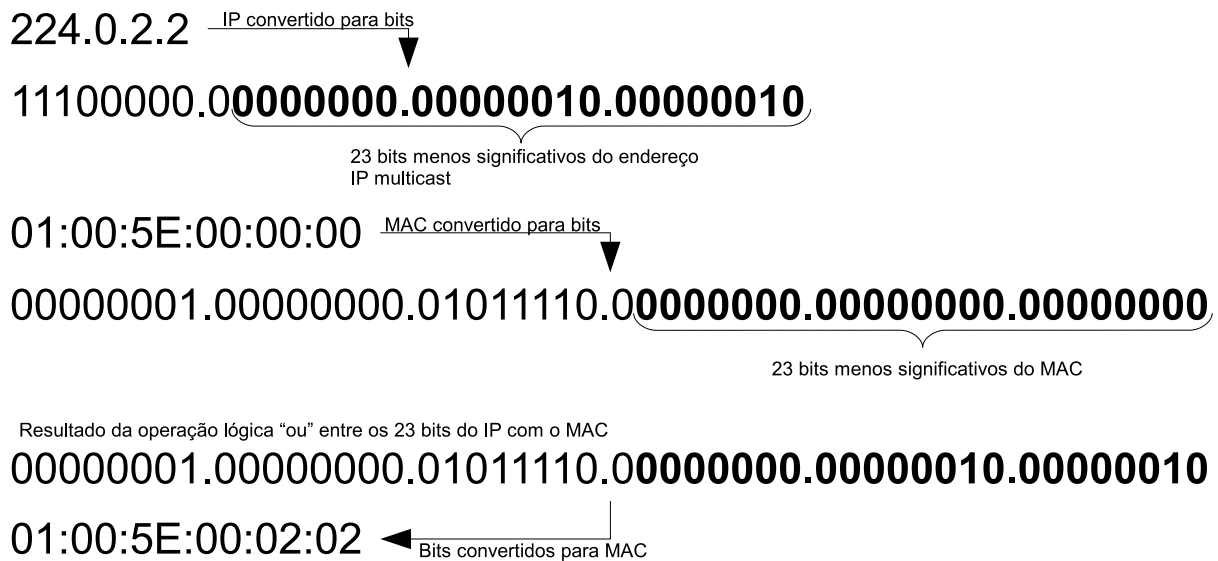


Figura 2.5: Mapeamento de IP para MAC *multicast*.

começam em 224.0.0.0 e terminam em 239.255.255.255. Quando um processo envia um pacote para um endereço classe D, é feita uma tentativa de entregá-lo a todos os membros do grupo endereçado, mas não há qualquer garantia de que isso realmente acontecerá. É possível que alguns membros não obtenham o pacote. Isto se deve ao fato de nem todos os roteadores de uma rede implementarem o protocolo *multicast*.

São aceitos dois tipos de endereços de grupo: endereços permanentes e endereços temporários. Um grupo permanente está sempre presente e não precisa ser configurado. Cada grupo permanente tem um endereço de grupo permanente. Alguns exemplos de grupos permanentes são:

- 224.0.0.1 Todos os sistemas de uma LAN.
- 224.0.0.2 Todos os roteadores de uma LAN.
- 224.0.0.5 Todos os roteadores *OSPF* de uma LAN.
- 224.0.0.6 Todos os roteadores *OSPF* designados de uma LAN.

Diferentemente dos grupos permanentes, que não precisam ser criados, para se usar um grupo temporário é preciso primeiramente criá-lo. Um processo pode solicitar que seu *host* se conecte a

um grupo específico ou solicitar que seu *host* saia de um grupo. Quando o último processo de um *host* deixa um grupo, o grupo passa a não mais existir no *host*. Cada *host* controla os grupos a que pertencem seus processos atuais (Tanenbaum, 1997). Segundo o IANA (*Internet Assigned Numbers Authority*) a faixa 239.0.0.0/8 é definida para um escopo *multicast* administrativo, o qual deve ser utilizado para configurar endereços de grupos temporários em redes locais *multicast*.

2.4.4 Protocolos de roteamento *multicast*

Os protocolos de roteamento utilizados em redes *multicast* são responsáveis, dentre outras atribuições, por otimizar as transmissões identificando o melhor caminho para a entrega de datagramas através dos roteadores que suportam esse modo de operação. Esta seção apresenta os principais protocolos de roteamento utilizados em redes *multicast*.

Segundo Tanenbaum (1997), o DVMRP (*Distance Vector Multicast Routing Protocol*) foi o primeiro protocolo de roteamento *multicast* a ser desenvolvido para grandes ambientes. O DVMRP utiliza um algoritmo de inundação (*flooding*) e poda (*prune*) em uma árvore de distribuição. O funcionamento do algoritmo consiste em, após receber um pacote *multicast* pela primeira vez, o roteador reenvia para todas as interfaces de saída menos a interface pela qual o pacote chegou. No DVMRP, o primeiro datagrama para cada fonte e grupo, canal (*Source, Group*) em *multicast*, é enviado para toda a árvore de distribuição tendo seu alcance limitado pelo campo TTL⁶ (*Time-to-Live*) do cabeçalho IP. Quando o datagrama enviado atinge um roteador folha que não tem, ao seu redor, nenhuma estação interessada no canal (S,G), este roteador dispara mensagens de poda na direção da fonte. Com esse método, utilizando mensagens de poda, é obtida a árvore com o caminho mais curto entre as estações interessadas em cada um dos canais (S,G) existentes na rede.

O MOSPF (*Multicast Extensions for Open Shortest-Path First*), é baseado no algoritmo de roteamento OSPF (*Open Shortest-Path First*) assim como o DVMRP é baseado no algoritmo de roteamento RIP (*Routing Information Protocol*). O MOSPF, assim como o OSPF, é um protocolo baseado

⁶TTL (*Time-to-Live*) diz respeito ao número de saltos que um pacote pode ter em uma rede de computadores, antes de ser descartado.

no estado de enlace. Neste método, cada roteador possui uma base de dados com o estado de todos os enlaces da rede. Quando um roteador detecta uma mudança do estado de um enlace, envia uma mensagem de atualização em *flooding* para todos os demais roteadores. O MOSPF diferencia-se do OSPF por adicionar um novo tipo de anúncio, utilizado para identificar a localização de membros dos grupos *multicast*. Os roteadores podem então utilizar o mesmo algoritmo para computar de forma local e independente a árvore de menor caminho para o *multicast*. Considerando uma rede dividida em ASs ⁷ (*Autonomous System*), o MOSPF permite *multicast* restrito a um único AS ou abrangendo múltiplos ASs (Tanenbaum, 1997).

O PIM (*Protocol Independent Multicast*) é um protocolo desenvolvido para rotear pacotes *multicast*. Ele é independente dos esquemas de roteamento IP *unicast* básicos. Como visto anteriormente, exemplos que utilizam IP *unicast* básicos são o DVMRP (*Distance Vector Multicasting Routing Protocol*) e o MOSPF (*Multicast Open Shortest Path First*). Esses mecanismos não apresentam boa eficiência quando os membros e os transmissores de um grupo estão distribuídos esparsamente numa ampla área (Gonçalves and Niles, 1998). O PIM se limita a criar caminhos entre os roteadores *multicast*, separando as atividades de protocolo de roteamento e protocolo de difusão. Os protocolos de difusão que utilizam o PIM são:

- PIM-DM (*Protocol Independent Multicast - Dense Mode*) (Adams et al., 2003): faz passar todo o tráfego *multicast* para todos os nós de uma rede. Por este motivo, é considerado um método pesado, já que difunde a sua mensagem para todas as sub-redes. No entanto, em determinadas situações, poderá ser um mecanismo eficiente caso existam clientes ativos em muitas das sub-redes existentes. Inicialmente o PIM-DM difunde todo o tráfego *multicast* em todas as sub-redes. No caso de existirem roteadores que não tenham na sua rede nenhum nó cliente ativo, os próprios roteadores enviam uma mensagem do tipo *prune* (poda) para informar que não desejam mais receber a referida informação. Para suportar grupos com elementos dinâmicos, todo o processo é repetido em intervalos regulares.
- PIM-SM (*Protocol Independent Multicast - Sparse Mode*) (Estrin et al., 1998): apresenta um

⁷AS (*Autonomous System*) é um grupo de redes IP gerenciada por uma ou mais operadoras de rede, que compartilham uma mesma política de roteamento.

funcionamento contrário ao do PIM-DM. Para o PIM-SM apenas as redes que possuem clientes ativos, que explicitamente requisitaram a informação pretendida, recebem o tráfego *multicast*. Este protocolo utiliza uma tabela de encaminhamento partilhada, para distribuir a informação de emissores ativos. Quando o tráfego começa a fluir pela árvore de roteadores, cada um ao longo do percurso determina se existe um melhor caminho para um determinado emissor. Se existir um caminho mais direto, o respectivo roteador irá enviar uma mensagem do tipo *join* ao emissor e, em seguida, o caminho é reconstruído. O protocolo PIM-SM introduz o conceito de Rendezvous Point (RP). O RP é um roteador designado, para onde as mensagens *multicast* fluem e a partir do qual se faz a entrega de pacotes através da árvore de distribuição *multicast*.

- PIM-SSM (*Protocol Independent Multicast - Source-Specific multicast*): é uma variante do PIM-SM. No protocolo PIM-SSM (Holbrook and Cheriton, 1999) a entrega de pacotes é baseada em canais (S,G). O tráfego dirigido a um canal (S,G) consiste em datagramas com endereço de origem S e endereço de destino o grupo *multicast* G. Os clientes que desejam receber determinada informação têm de se inscrever no canal (S,G). Ao fazer a inscrição em um canal (S,G), o cliente além de especificar qual grupo é do seu interesse, pode também especificar de qual fonte deseja receber o fluxo. No SSM, devido à natureza descentralizada e da possibilidade de existirem várias fontes para o mesmo grupo, o cliente, além de especificar o grupo também especifica uma fonte, o fluxo então será enviado a partir da fonte selecionada pelo cliente.

2.4.5 Protocolos de gerenciamento *multicast*

Como visto na Seção 2.4.4 e em Tanenbaum (1997), os algoritmos de roteamento *multicast* são responsáveis pela construção de uma árvore de distribuição, a fim de otimizar a entrega de pacotes a clientes interessados em determinado grupo.

O protocolo *multicast* é baseado no conceito de grupo. O protocolo é implementado por roteadores *multicast*, que também podem ser usados como roteadores comuns. Aproximadamente uma vez por minuto cada roteador *multicast* envia uma mensagem para os *hosts* de sua LAN, solicitando que eles informem os grupos aos quais seus processos pertencem atualmente. Com isso, cada roteador terá o

registro de todos os fluxos *multicast* existentes na LAN.

Os pacotes de consulta e resposta utilizam o protocolo IGMP (*Internet Group Management Protocol*), para gerenciamento de grupos *multicast*. O protocolo IGMP tem apenas dois tipos de pacotes: consulta e resposta. Cada pacote tem um formato simples, contendo algumas informações de controle na primeira palavra do campo de carga útil e um endereço da classe D (classe reservada para endereços *multicast*) na segunda palavra (Tanenbaum, 1997).

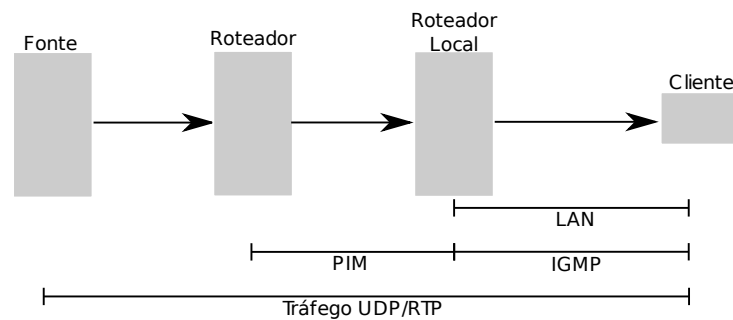


Figura 2.6: Delimitação dos protocolos PIM e IGMP em uma rede com suporte *multicast*.

A Figura 2.6 ilustra a delimitação dos protocolos utilizados numa rede *multicast*. Para receber um conjunto de dados de uma fonte é necessário que o cliente se inscreva em um grupo, por meio do protocolo IGMP (Cain et al., 2002), regido por roteadores *multicast* vizinhos. O protocolo de gerenciamento de grupo IGMP é utilizado pelo **Cliente** para reportar ao **Roteador Local** o interesse em participar do grupo de transmissão criado pela **Fonte**.

O protocolo IGMP é utilizado na versão quatro do protocolo IP (IPv4). Na versão seis (IPv6) o IGMP mudou para MLD (*multicast Listener Discover*) (Deering et al., 1999). Além do IGMP, todos os nós da rede (servidores, estações e roteadores) devem suportar o formato de endereçamento e comunicação *multicast* (Seção 2.4.3), somente assim é possível a transmissão de dados neste mecanismo de distribuição.

2.5 Comunicação P2P

Diferentemente do modelo Cliente/Servidor ⁸, um sistema P2P (*peer-to-peer*) consiste em um *pool* de recursos (conteúdo, armazenamento, largura de banda de rede, largura de banda de disco, CPU) disponível a todos os seus elementos, implementada de forma descentralizada, auto-organizável e escalável. Cada elemento do sistema P2P pode atuar tanto como fornecedor quanto consumidor de um determinado recurso, podendo até mesmo fazer ambos os papéis (Peterson and Davie, 2004).

Um *peer* **P1**, que solicita determinado conteúdo a outro *peer* na *Internet*, **P6**, tem sua comunicação caracterizada como fornecedor/consumidor onde **P1** é consumir e **P6** é fornecedor. Nesta mesma comunicação, **P6** pode requisitar um conteúdo que **P1** possa fornecer e neste caso os papéis se invertem, **P1** passa a agir como fornecedor e **P6** cliente (Figura 2.7).

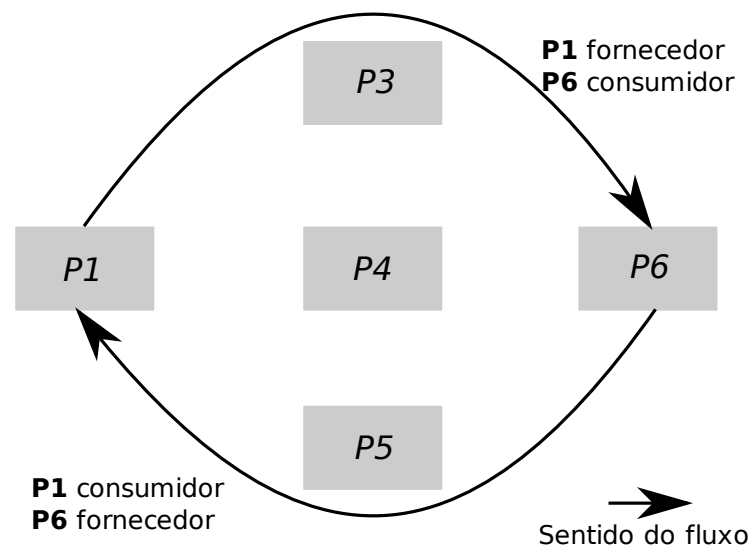


Figura 2.7: Exemplo de comunicação entre *peers* P2P.

No compartilhamento de arquivos P2P, a distribuição de conteúdo é feita diretamente entre os *peers* em comum, sem passar por servidores. Dessa forma, as aplicações P2P tiram proveito dos recursos (largura de banda, armazenamento e processamento) de um grande conjunto de *peers*. Em

⁸Cliente/Servidor é um modelo computacional que distingue clientes de servidores, sendo todos interligados utilizando uma rede de computadores. Cada instância de um cliente pode enviar requisições de dado para algum dos servidores conectados e esperar pela resposta. Por sua vez, algum dos servidores disponíveis pode aceitar tais requisições, processá-las e retornar o resultado para o cliente.

outras palavras, o compartilhamento de arquivos P2P é altamente escalável. A Figura 2.8 mostra um cenário onde **Fonte**, **Peer 1**, **2**, **3** e **4** implementam uma rede P2P. Nessa rede, todos os **Peers** desejam receber determinado conteúdo em comum, que é anunciado pela **Fonte**. Inicialmente **Peer 1** e **4** recebem um fluxo da **Fonte**, que é posteriormente compartilhado por P2P com os **Peers 2** e **3**. O P2P neste cenário economizou tráfego no **Roteador**. Os **Peers 2** e **3**, ao invés de obterem o conteúdo transmitido diretamente da **Fonte**, solicitaram o mesmo de seus vizinhos P2P (**Peer 1** e **4**), que já detinham uma cópia do conteúdo.

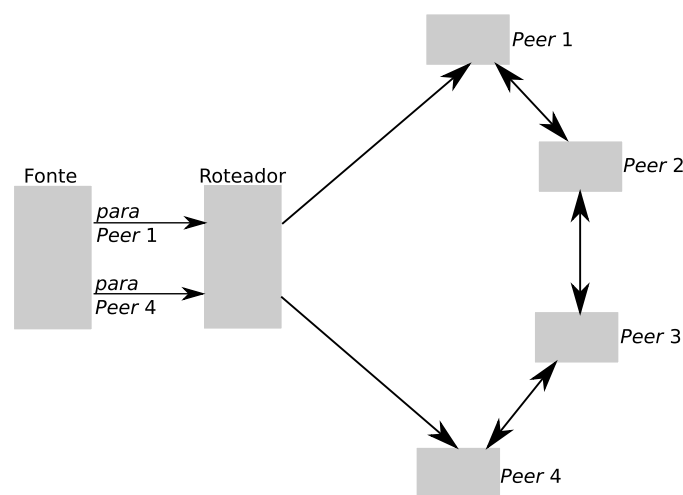


Figura 2.8: Funcionamento do método de transmissão P2P.

Mesmo não existindo servidores em meio à transferência de arquivos entre clientes no P2P, é importante ter em mente que o compartilhamento de arquivos P2P ainda se baseia no modelo cliente/servidor. O *peer* requisitante é o cliente e o *peer* escolhido é o servidor. O arquivo é enviado do *peer* servidor ao *peer* cliente, com um protocolo *unicast* de transferência de arquivos. Visto que qualquer *peer* pode requisitar ou ser requisitado, todos os *peers* podem executar tanto o lado cliente quanto o lado servidor do protocolo de transferência de arquivos (Kurose and Ross, 2006).

Normalmente há um grande número de *peers* conectados a um sistema P2P e cada um desses *peers* detém conteúdos diversos para compartilhar. Se um *peer* está interessado em obter um determinado conteúdo, ele deverá ter um modo de determinar os endereços IP dos *peers* que tenham a cópia de tal conteúdo e assim proceder à requisição para a transferência. Conectar-se a *peers* em um sistema P2P não é um problema trivial. As arquiteturas centralizadas, descentralizadas ou híbridas são métodos

que auxiliam a comunicação entre *peers* num sistema P2P e os modelos clássicos dessas arquiteturas são apresentados nas seções seguintes.

2.5.1 P2P centralizado

Diretório centralizado, ou P2P centralizado, segundo Kurose and Ross (2006), é uma das abordagens mais diretas na localização de conteúdo em uma rede com sistema P2P. Um exemplo clássico desse método é o *Napster*⁹, a primeira empresa comercial a disponibilizar uma aplicação P2P de grande escala para distribuição de conteúdo.

Neste método, o serviço de compartilhamento de arquivos P2P usa servidores para prover um serviço que armazene uma lista do conteúdo de cada *peer* ativo. Como mostra a Figura 2.9, no momento em que um *peer* é ativado na rede P2P, ele informa ao servidor da rede seu endereço IP e o conteúdo que tem. Deste modo, o servidor sabe qual conteúdo um *peer* tem disponível para compartilhamento em seu disco local. O servidor recebe estes dados de cada *peer* ativo na rede, criando assim um banco de dados centralizado que mapeia cada conteúdo para um conjunto de endereços IP. Periodicamente os *peers* enviam informações sobre o estado de seus objetos em disco, para que o servidor possa atualizar sua base. Para saber se cada *peer* está ou não ativo, o servidor monitora os *peers* por meio de mensagens. Caso o *peer* tenha sido desligado seu IP é removido da base de dados.

Existem desvantagens no uso de servidores centralizados:

- existe um único ponto de falha: caso o servidor central fique inoperante, toda a rede P2P cairá;
- há um gargalo de desempenho: em um sistema P2P muito grande, as centenas de milhares de consultas podem sobrecarregar o servidor.

⁹*Napster*: <http://www.napster.com>

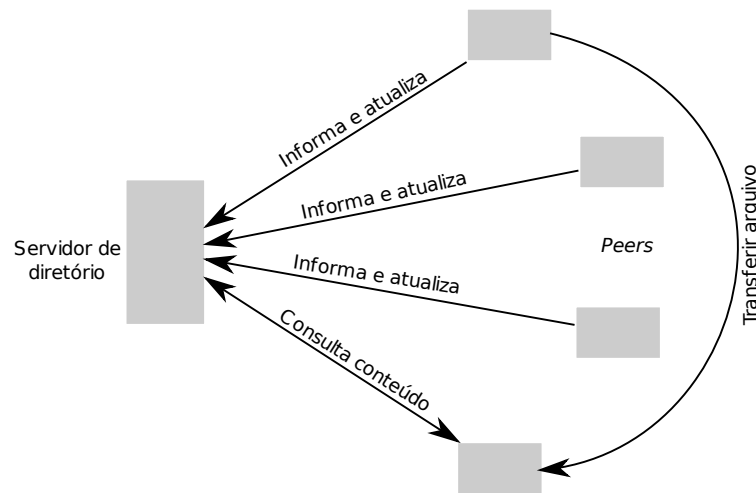


Figura 2.9: P2P com servidor centralizado (*Napster*).

2.5.2 P2P distribuído

No caminho oposto ao do P2P centralizado, o *Gnutella* (Gnutella, 2004), exemplo clássico de P2P distribuído, utiliza uma abordagem diversa para a busca de conteúdo em *peers* num sistema P2P. Diferentemente do *Napster*, que utiliza um servidor para centralizar todo o conteúdo existente nos *peers* da rede para assim prover um mecanismo de consulta, o *Gnutella* implementa suas consultas fazendo inundação (*flooding*) nos *peers* da rede P2P. O processo de inundação (*flooding*) em redes P2P *Gnutella* é descrito adiante.

Os *peers* em uma rede *Gnutella* formam uma rede lógica (virtual), denominada rede de sobreposição (*overlay network*). Se um *peer X* mantém conexão com um *peer Y*, então há uma aresta entre *X* e *Y*. O grafo que contém todos os *peers* ativos e arestas de conexão (no caso do *Gnutella*, conexões TCP em curso) define a rede de sobreposição *Gnutella* existente. A conexão entre as arestas, neste caso, é apenas virtual, pois não existe enlace físico entre elas. Entre duas conexões virtuais podem existir dezenas de elances físicos (Kurose and Ross, 2006).

Em uma rede *Gnutella*, *peers* enviam mensagens a outros *peers* próximos na rede de sobreposição. Quando um *peer* deseja localizar determinado conteúdo na rede *Gnutella*, este *peer* envia a todos seus *peers* vizinhos uma mensagem *Gnutella Query* (Gnutella, 2004). Essa mensagem inclui as palavras-chave a serem pesquisadas. Os vizinhos, por sua vez, retransmitem a mensagem a todos os seus

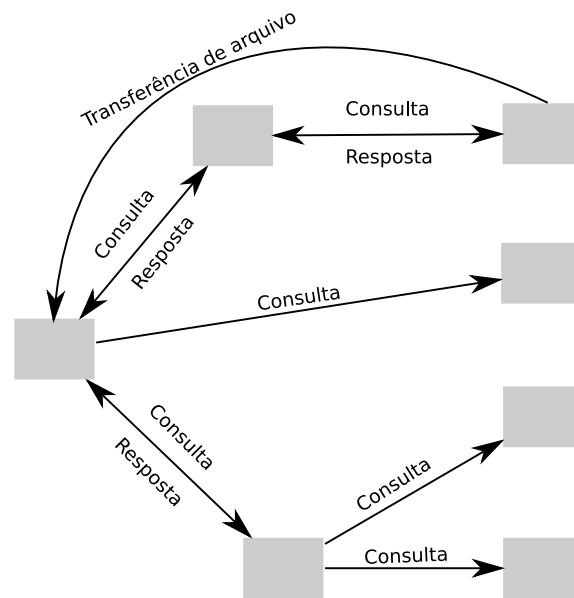


Figura 2.10: P2P com método descentralizado (*Gnutella*).

vizinhos e assim por diante. A este processo se dá o nome de inundação de consultas e é ilustrado na Figura 2.10. Quando um *peer* recebe uma mensagem *Query*, este verifica se existe algum conteúdo que se assemelha a palavra-chave. Se houver compatibilidade, o *peer* então devolve uma mensagem *QueryHit* pelo caminho inverso à mensagem *Query*, utilizando assim a mesma conexão TCP.

Para manipular as junções e saídas de *peers* em um sistema P2P *Gnutella*, o protocolo implementa o seguinte método (Gnutella, 2004):

1. Um *peer X* que deseja se conectar a rede de sobreposição, deve primeiramente encontrar um *peer* já ativo nesta rede. O *bootstrap* no método *Gnutella* é resolvido armazenando uma lista de *peers* que estão frequentemente ativos na rede.
2. De posse dessa lista, *X* faz tentativas seqüenciais para estabelecer uma conexão com *peers* presentes nesta lista, até ser criada uma conexão com algum *peer Y*.
3. Após estabelecida uma conexão TCP entre *X* e *Y*, o *peer X* envia ao *peer Y* uma mensagem *Gnutella Ping*, que também inclui um campo de contagem de *peers*, uma espécie de TTL (*Time To Live*), como utilizado no protocolo IP. Ao receber a mensagem *Ping*, *Y* decrementa o campo de contagem de *peers* e a transmite a todos os seus vizinhos na rede de sobreposição. Os *peers*

continuam a retransmitir a mensagem *Ping* até o campo de contagem de *peers* ser zerado. Este processo recebe o nome de inundação (*flooding*).

4. Sempre que um *peer Z* recebe uma mensagem *Ping*, este responde com uma mensagem *Pong* para *X*, pela própria rede de sobreposição. A mensagem *Pong* inclui o endereço IP de *Z*, o número de arquivos que ele está compartilhando e o número total de bytes ocupado pelos arquivos que está compartilhando.
5. Quando recebe as mensagens *Pong*, *X* sabe os endereços IP de muitos *peers*, além de *Y*, presentes na rede *Gnutella*. Pode então estabelecer conexões TCP com alguns desses outros *peers*, criando várias arestas ligadas a ele dentro da rede de sobreposição. O *peer X* também pode tentar estabelecer conexões TCP com outros nós que estão se juntando à rede. A especificação *Gnutella* não determina com quantos *peers X* deve se conectar.

Em resumo, o *Gnutella* é um exemplo clássico de P2P distribuído, que torna possível que um usuário consulte arquivos localizados em *peers* próximos, na rede de sobreposição (Gnutella, 2004; Kurose and Ross, 2006).

2.5.3 P2P hierárquico

As duas soluções que implementam métodos clássicos de P2P, *Napster* e *Gnutella*, utilizam abordagens completamente distintas, um inteiramente centralizado e outro descentralizado. Algumas limitações dessas abordagens motivaram a criação de um novo método P2P para distribuição. O novo método consiste em não concentrar toda a informação dos *peers* em um único ponto, mas também não ser descentralizado ao ponto de ser possível apenas pesquisar o conteúdo de *peers* mais próximos.

A nova aplicação surge com o *KaZaA* (KaZaA, 2008), que se parece com o *Gnutella* no sentido de não utilizar um servidor dedicado para monitorar e localizar conteúdo. No entanto, é diferente do *Gnutella* em que nem todos os *peers* são iguais. Os *peers* mais poderosos, os que tem mais conectividade e largura de banda na *Internet*, são designados líderes de grupo e têm maiores responsabilidades. Se um *peer* não for um líder de grupo, então este fica designado a um líder de grupo. Normalmente,

um líder de grupo terá até algumas centenas de *peers* filhos (Kurose and Ross, 2006). A Figura 2.11 apresenta um exemplo de uma rede P2P hierárquica.

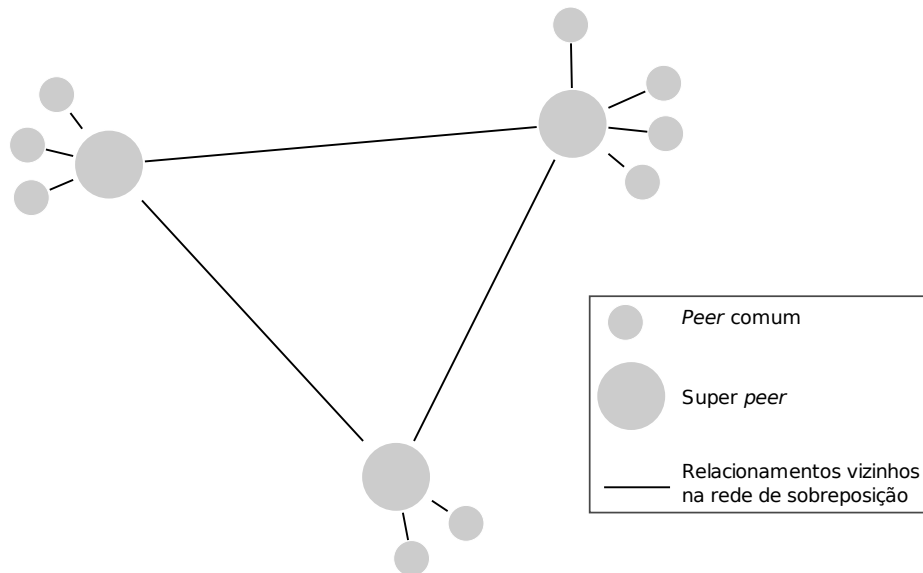


Figura 2.11: Rede de sobreposição hierárquica P2P (*KaZaA*).

Quando um *peer* lança a aplicação *KaZaA*, estabelece uma conexão TCP com um dos líderes de grupo. O *peer* então informa a seu líder de grupo todos os arquivos que está compartilhando. Isso permite ao líder manter um banco de dados que inclui os identificadores de todos os arquivos, os metadados¹⁰ e os endereços IP correspondentes que seus *peers* designados possuem. Cada líder de grupo torna-se uma espécie de *hub* (distribuidor). Um líder de grupo age como se fosse um pequeno servidor no *Napster*. No entanto, o líder neste caso é apenas um *peer* comum e não um servidor dedicado como no *Napster*.

Se cada um dos líderes, juntamente com seus *peers* designados, fossem entidades isoladas no *KaZaA*, a aplicação consistiria em milhares de servidores *Napster* espalhados. A diferença seria que o distribuidor é executado em um *peer*, ao invés de em um servidor dedicado. Essa abordagem limitaria seriamente a quantidade de conteúdo disponível em qualquer um dos *peers*. Para resolver esse problema, o *KaZaA* compartilha conteúdo de informação dos vários grupos por meio de conexões TCP que interligam todos os líderes de grupo em uma rede de sobreposição (Figura 2.11). Os líderes

¹⁰Metadado ou Metainformação é qualquer informação sobre determinada informação (“dados sobre dados”). A frase “a bola tem cor azul” é uma metainformação sobre a bola.

são ligados de uma forma semelhante a qual todos os *peers* são ligados na rede *Gnutella* (Kurose and Ross, 2006).

Para cada conteúdo que um cliente P2P possui no *KaZaA*, existe um descritor. Este descritor inclui o nome do arquivo e um texto descritivo não estruturado do objeto. Os *peers* fazem consultas por palavras-chave. Para procurar uma compatibilização para as palavras-chave, um *peer* as envia a seu líder de grupo como uma consulta. O líder, que pode re-encaminhar a consulta para os demais líderes na rede de sobreposição, responde com uma lista de *peers* (dentro ou fora de seu grupo) que possuem arquivos cujos descritores são compatíveis com a consulta.

O *KaZaA* emprega ainda algumas técnicas que melhoram seu desempenho:

- **Enfileiramento de requisições:** é possível configurar um *peer* de modo a limitar o número de transferências simultâneas. Essa limitação garante que cada arquivo que está sendo transferido não receba somente uma quantidade desprezível de largura de banda.
- **Prioridades de incentivo:** esse mecanismo proporciona aos *peers* um incentivo para o envio de arquivos. É dada uma prioridade no enfileiramento de requisições para os *peers* que comumente enviam mais arquivos na rede.
- **Transferência paralela:** frequentemente a consulta por um arquivo retorna mais de uma fonte. Neste caso, é possível requisitar diferentes porções do arquivo de diferentes *peers*, baixando assim várias porções do arquivo em paralelo.

2.6 Distribuição de Tráfego IPTV: estado da arte

Muitos esforços estão sendo feitos para reduzir o impacto causado pela chegada do conteúdo de TV digital em redes IP, o sistema IPTV (Seção 2.1). O principal foco dos trabalhos citados nesta seção, assim como o desta dissertação, é de otimizar a carga imposta pelo conteúdo de TV na rede e/ou na fonte de distribuição. Um modelo de distribuição de conteúdo IPTV, com a finalidade de reduzir a carga imposta na rede de distribuição, utilizando *multicast*, pode ser visto em Caja (2006).

Neste trabalho (Caja, 2006) o autor descreve os desafios da prestação de serviços de IPTV utilizando redes metropolitanas com uma versão não especificada de protocolo *multicast*. O autor afirma que, para as empresas do ramo de IPTV aumentarem seus serviços e números de clientes, é necessário uma eficiência melhor na calibragem da largura de banda exigida por cada canal. Ressalta ainda a impossibilidade de enviar cada canal para a borda da rede utilizando fluxos únicos (*unicast*), pois isso sobrecarregaria a rede, segundo o autor. Este então sugere o protocolo *multicast* (sem uma versão definida do PIM) para otimizar a resposta da rede na distribuição de tráfego IPTV.

O trabalho proposto em Smith (2007) apresenta um panorama do tráfego num sistema IPTV para SD (*Standart Definition*) e HD (*High Definition*). Neste trabalho, o autor considerou os intervalos de uma programação (*channel surfing*, segundo o trabalho) como momentos nos quais o telespectador geralmente alterna entre diversos canais, ocasionando um aumento na demanda por largura de banda. Foi proposto então um modelo de demanda extra para uma rede que utilize protocolos *multicast*. O modelo visa quantificar a banda nos momentos onde o telespectador alterna entre canais. Os resultados deste trabalho, com demanda média de 400 espectadores, mostrou que durante intervalos na programação houve um curto tempo (670 ms) na troca entre canais. Em Uzunalioglu (2009) o autor também faz medições no tempo entre mudanças de canais em redes *multicast* e obtém resultado classificado como aceitável, cerca de 830 ms.

Em Komosny and Novotny (2007), os autores também fazem uso do protocolo *multicast* PIM-SSM, desta vez em transmissões utilizando RTP (*Real Time Protocol*)¹¹ e RTCP (*Real Time Control Protocol*)¹². Os autores desenvolveram um modelo, chamado de agregação hierárquica, para o envio de *feedbacks* das transmissões realizadas pelo RTCP. Segundo os autores, esses *feedbacks*, se enviados em larga escala (como o que acontece em um sistema IPTV) criam uma demanda extra de largura de banda (5% do total de uma transmissão utilizando o protocolo PIM-SSM e RTP). O modelo consiste em sumarizar os *feedbacks* nos roteadores da árvore de distribuição *multicast* de forma hierárquica,

¹¹RTP (*Real Time Protocol*) é um protocolo utilizado para aplicações de tempo real como Voz sobre IP (VoIP). Define como deve ser feita a fragmentação do fluxo de dados adicionando a cada fragmento informações de sequência e tempo de entrega

¹²O RTCP (*Real Time Control Protocol*) funciona juntamente com o RTP com a função de enviar pacotes de controle aos participantes de uma transmissão. O RTCP é encarregado principalmente de fornecer um *feedback* da qualidade dos serviços oferecidos pelo RTP.

partindo do cliente em direção à fonte.

Uma das empresas de maior abrangência no mercado de soluções para redes de computadores e telecomunicações, a *Cisco System Inc.*, também tem seu plano de serviço para o sistema IPTV em redes corporativas baseado em protocolos *multicast* (Cisco Systems Inc., 2007). Tanto nos trabalhos citados anteriormente como em She et al. (2009); Luo et al. (2008); She and Ho (2008); Begen et al. (2009), os autores utilizaram modelos semelhantes de *multicast* obtendo resultados próximos na economia de largura de banda utilizada na rede de distribuição para a entrega de serviços de IPTV.

A *Internet*, diferente de uma rede corporativa, é uma rede completamente heterogênea. Isto pode impossibilitar a transmissão utilizando somente o protocolo *multicast*. Este protocolo, para ter seu funcionamento garantido, deve contar com roteadores *multicast* contíguos ligando a fonte do tráfego a todos os clientes. No protocolo P2P, por utilizar fluxos *unicast*, há compatibilidade entre quaisquer roteadores. Por este motivo, muitos trabalhos utilizam o P2P para corrigir possíveis falhas no caminho *multicast*.

Os aplicativos (*PPlive*, *PPstream*, *TVants* e *Sopcast*), relacionados em Silverston and Fourmaux (2006), são exemplos de aplicações que utilizam o método P2P na distribuição de conteúdo em um sistema IPTV na *Internet*. A *Internet* tem um agravante de não garantir qualidade e largura de banda, no entanto, medições realizadas no tráfego de pacotes de *peers* que estavam na mesma rede durante transmissões da Copa do Mundo de Futebol em 2006, momentos de grande fluxo de dados nas aplicações abordadas, apontam a tecnologia P2P como promissora (Silverston and Fourmaux, 2006).

Aplicações P2P num sistema IPTV reduzem a carga imposta principalmente na fonte do tráfego. Contudo a rede de distribuição é sobrecarregada com fluxos *unicast*, pois os *peers* devem colaborar entre si para formar ou completar o conteúdo IPTV.

Em Wu et al. (2008) os autores mostram que alguns dos sérios problemas do P2P são a instabilidade de disponibilidade dos *peers* e a restrição quanto a sua banda de *upload*. Para reduzir este impacto negativo, o trabalho propõe um servidor online com a capacidade de provisionamento, que ajusta proativamente as capacidades disponíveis de largura de banda para cada um dos canais concorrentes, tendo em conta o número de *peers*, a qualidade e as prioridades dos canais. Trabalhos que

abordam IPTV com resultadores semelhantes utilizando métodos P2P podem ser vistos em Hei et al. (2008) e Mushtaq and Ahmed (2009).

Alguns métodos P2P reduzem a carga imposta não apenas na fonte do tráfego. Mesmo utilizando fluxos *unicast*, eles reduzem a carga também na rede de distribuição, como o visto em Cha et al. (2008). O método proposto reduz a carga na rede limitando as requisições somente entre *peers* vizinhos na rede física. Desta forma, o compartilhamento do conteúdo entre os *peers* é realizado através de poucos roteadores e isso faz com que a carga total da rede seja reduzida, pois as conexões entre *peers* ocupam menos enlaces. Os resultados do autor mostram que, mesmo com a redução do tráfego gerado pelo modelo P2P de distribuição proposto, o *multicast* ainda tem um fluxo de dados menor na rede de distribuição.

Outros trabalhos apostam em tecnologias híbridas para redução de carga num sistema IPTV. De acordo com o apresentado em Venkataraman et al. (2006) e Agarwal (2007), *multicast* e P2P são utilizados conjuntamente a fim de criar um sistema de distribuição capaz de superar a heterogeneidade da *Internet*, utilizando o protocolo P2P, e otimizar a transmissão do tráfego, utilizando o protocolo *multicast*, quando possível. Em Gallo et al. (2009) os autores propõem uma arquitetura *multicast* para a distribuição de conteúdo ao vivo, e o protocolo P2P para a oferta de VoD. Os resultados, segundo os autores, apontam para uma solução que é eficiente para todos os serviços relacionados com a televisão, seja ao vivo ou em VoD. Outros trabalhos que utilizam soluções híbridas (P2P e *multicast*) são vistos em Zeng and Strauss (2008) e Tian et al. (2008), com resultados semelhantes.

Capítulo 3

Metodologia

O objetivo deste capítulo é descrever como foram realizados os experimentos e as simulações que serviram como base de resultados para esta dissertação. Neste capítulo são apresentados o ambiente de trabalho (Seção 3.1), o simulador e as alterações necessárias, bem como os protocolos simulados (Seções 3.2 e 3.3, respectivamente), a topologia de rede de distribuição (Seção 3.4) e a geração do perfil de tráfego com base em programas televisivos (Seção 3.5).

3.1 Ambiente

O sistema operacional GNU/Linux foi utilizado durante todo o trabalho de desenvolvimento desta dissertação. A criação de aplicativos, análise de resultados, simulações e a escrita do texto tiveram como base a distribuição Gentoo Linux 2008, compilada em 64 bits. O sistema operacional Linux foi utilizado por oferecer um bom desempenho, frente ao grande volume de dados a ser processado pelas simulações, e por oferecer compatibilidade com as aplicações necessárias. O hardware utilizado nas simulações foi um Intel Core 2 Duo 1.66 GHz, com 4GB de memória RAM e dois HDs de 250 GB.

Para o ambiente de trabalho foram utilizadas diferentes linguagens de programação: C, BASH (*Born Again SHell*), *awk* e *perl* para a análise e processamento de arquivos e automatização das

simulações, que utilizaram o ns-2 descrito na Seção 3.2. O *vim* foi utilizado como IDE (*Integrated Development Environment*) para o desenvolvimento de todos os aplicativos que foram necessários para as simulações e o processamento de resultados. O *Matlab*, software de computação científica, foi utilizado para os cálculos finais das matrizes extraídas do simulador e a plotagem de gráficos estatísticos.

3.2 Simulador de Rede ns-2

O Network Simulator (ns-2) é um dos principais simuladores de redes de computadores atualmente em uso pela comunidade científica. Seu código foi desenvolvido pelo LBL (*Lawrence Berkeley National Laboratory*), Xerox PARC (*Palo Alto Research Center*), Universidade da Califórnia em Berkeley e Sun Microsystems. Atualmente o ns-2 é mantido pelo DARPA (*Defense Advanced Research Projects Agency*), através do projeto SAMAN (*Simulation Augmented by Measurement and Analysis for Networks*), e pela NSF (*National Science Foundation*), através do projeto CONSER (*Collaborative Simulation for Education and Research*), em colaboração com outros centros de pesquisa como o ICSI (*International Computer Science Institute*). O Network Simulator é descrito em ISI (2009).

O ns-2 é um simulador baseado em eventos discretos e orientado a objetos. Seu código foi desenvolvido utilizando linguagens C/C++, TCL (*Tool Command Language*) e OTCL (*Object-oriented Tool Command Language*). O objetivo do ns-2 é proporcionar um ambiente para o desenvolvimento de pesquisas em torno dos protocolos que constituem a *Internet*, isto é, que utilizam a pilha TCP/IP, tanto no contexto das redes fixas quanto móveis, com fio e sem fio (ISI, 2009). O ns-2 é uma ferramenta poderosa para configurar simulações complexas e também para comparação de resultados de pesquisas. Estão implementados no ns-2 os protocolos IP, TCP, UDP, FTP, HTTP, além dos protocolos de roteamento e *multicast*.

O formato dos resultados obtidos a partir de uma simulação feita com o ns-2 é o apresentado na Figura 3.1. O arquivo, em texto simples com a extensão *tr* (*trace*), tem sua estrutura dividida em


```
+ 6 10 0 encap 1020 ----- 0 10.4 0.0 -1 1835
+ 6 10 0 encap 92 ----- 0 10.4 0.0 -1 1837
r 6.000052 0 11 ack 40 ----- 7 9.1 11.1 480 1815
+ 6.000052 11 0 tcp 1040 ----- 7 11.1 9.1 500 1838
- 6.000052 11 0 tcp 1040 ----- 7 11.1 9.1 500 1838
- 6.000194 0 3 tcp 1040 ----- 7 11.1 9.1 498 1828
- 6.000453 10 0 encap 1020 ----- 0 10.4 0.0 -1 1835
```

Figura 3.1: Conteúdo do arquivo de *trace* do ns-2.

colunas como descrito abaixo:

1. “+” - enfileiramento, “-” - envio, “r” - pacote recebido, “d” - pacote descartado;
2. instante (em segundos) em que ocorreu o evento;
3. nó de origem do evento;
4. nó de destino do evento;
5. tipo de protocolo utilizado;
6. tamanho em bytes do pacote;
7. flags do evento;
8. id do fluxo;
9. endereço do nó de origem do pacote;
10. endereço do nó de destino do pacote;
11. número de sequência;
12. id do pacote.

3.3 Simulação de protocolos no ns-2

O Network Simulator (ns-2) permite que sejam simuladas redes de computadores com diferentes topologias, configurações de enlaces e tráfegos. Para este trabalho, que envolve simulações principalmente utilizando o protocolo *multicast*, o ns-2 tem uma limitação. Dos protocolos de distribuição que este trabalho se propôs a simular (*unicast*, *multicast* PIM-DM, PIM-SM, PIM-SSM e P2P), o ns-2 não conta com o suporte ao PIM-SSM e P2P. Para simular o protocolo PIM-SSM foi utilizado um novo módulo para o ns-2, descrito na Seção 3.3.1. E, de acordo com argumentação teórica apresentada na Seção 3.3.2, não foi realizada a simulação de P2P.

3.3.1 PIM-SSM no Network Simulator (ns-2)

Para simular o protocolo PIM-SSM foi utilizado um novo módulo para o ns-2, descrito em Camilo et al. (2004) e também utilizado em Silva et al. (2008) e Pereira et al. (2007). Segundo os autores, o protocolo PIM-SSM é, em parte, similar ao protocolo PIM-SM. Por este motivo o desenvolvimento do PIM-SSM para o ns-2 foi baseado no código do protocolo *centralized multicast*, versão do PIM-SM para o ns-2. As principais diferenças entre estes dois protocolos são:

- No PIM-SSM as mensagens de *join* e *prune* necessitam especificar o *Group* e a *Source* do canal *multicast*. Por este motivo os roteadores não aceitam mensagens do tipo (*,G), onde não é especificado *Source*, como ocorre no protocolo PIM-SM (Capítulo 2 e (Camilo et al., 2004));
- Em ambientes PIM-SSM, o nó RP deixa de ser necessário. A tabela de encaminhamento passa a ser descentralizada, o que não acontecia no protocolo PIM-SM, onde todos os pacotes teriam necessariamente de passar pelo nó RP.

O autor propõe uma nova classe a ser inserida no ns-2, que altera o comportamento do PIM-SM, classe esta que serviu de base para adequar-se ao PIM-SSM. A classe *centralized multicast* (PIM-SM) não é compatível com o comportamento do PIM-SSM. Na arquitetura do PIM-SM não é necessário guardar a relação entre *Group* e *Source*, já que para um *Group* específico podem existir

várias *Sources*, (*,G). O RP, no PIM-SM, centraliza todos os *Groups* sem fazer distinção das *Sources*. No PIM-SSM é necessário guardar relação entre *Source* e *Group* num *array*, pois diferentes *Groups* podem ser distribuídos por diferentes *Sources*. Este procedimento realizado pelo PIM-SSM permite, por exemplo, declarar vários grupos pertencentes a uma *Source* específica, considerando dois canais distintos, por exemplo (S_0, G_1) e (S_0, G_2) (Camilo et al., 2004).

É possível verificar a existência de apenas um elemento nas mensagens emitidas pelo ns-2 (Seção 3.2) durante uma simulação com o PIM-SM, o *Group*. Entretanto, na classe desenvolvida para representar o PIM-SSM existem dois elementos, o *Source* e o *Group*. A função responsável por criar toda a tabela de encaminhamento (função *compute-branch*) na versão do protocolo PIM-SM, obriga que a *Source* encaminhe a informação para o RP. Como no PIM-SSM o RP não é necessário, este comportamento foi modificado para uma ligação direta entre *Source* e o nó receptor. Em outras palavras, as alterações propostas em Camilo et al. (2004) eliminaram a presença do RP e encaminharam o tráfego da fonte diretamente para os receptores.

3.3.2 Protocolo P2P

A ideia central deste trabalho é determinar qual protocolo impõe menor carga na rede de distribuição durante a transmissão de conteúdo IPTV. Nesta seção é apresentada uma argumentação teórica para a não realização de simulação do protocolo P2P. De acordo com Agarwal (2007), é possível chegar a uma conclusão preliminar de que, como no *unicast*, se só o P2P for utilizado (sem suporte de um protocolo como o *multicast*) ele também não será uma boa alternativa para reduzir carga neste ambiente. O protocolo *unicast*, como visto no Capítulo 2, cria canais únicos de comunicação para cada cliente, a partir da fonte de transmissão. O P2P, por sua vez, cria múltiplas conexões entre os *peers* a fim de compartilhar o conteúdo recebido, seja por um servidor ou outro *peer*. O protocolo P2P, como no *unicast*, não otimiza o número de cópias a transitarem pela rede de distribuição como o *multicast*, mas, diferente do *unicast*, tenta sobrecarregar o mínimo possível a fonte de conteúdo, distribuindo a carga do tráfego entre os *peers* da rede.

Em Cha et al. (2008), como comentado na Seção 2.6, os autores desenvolvem um método que

reduziu o tráfego em redes P2P. Para isso, limitou-se as requisições entre *peers* somente para vizinhos na rede física. Mesmo com uma redução expressiva no tráfego P2P, é possível perceber neste trabalho que o fluxo de dados proveniente deste método P2P ainda é maior que o do *multicast*.

Os autores utilizaram um cenário-exemplo, simplificado, para determinar graficamente o custo de cada mecanismo de transmissão. Tal abordagem também é utilizada nesta dissertação para representar os custos dos mecanismos *unicast*, P2P e *multicast*.

O custo de distribuição, neste caso, deve ser considerado como o número de cópias de um determinado conteúdo que passa por um enlace. As Figuras 3.2, 3.3 e 3.4, apresentam uma topologia simplificada do núcleo de uma rede de distribuição onde são avaliados os custos que *unicast*, *multicast* e P2P, respectivamente, demandam para distribuir um conteúdo para as redes dos clientes (redes estas representadas por **Cliente 1**, **Cliente 2**, **Cliente 3** e **Cliente 4**). Seguindo as características de uma rede de distribuição típica, que em geral se estrutura na forma de uma árvore, não existem ligações intermediárias entre os ramos.

O protocolo *unicast*, de acordo com a Figura 3.2, obteve custo 8. Para este protocolo realizar uma transmissão é criado um canal único a partir da fonte para cada cliente. No exemplo da Figura 3.2, a fonte envia dois fluxos para cada roteador, que em seguida são entregues aos dois clientes que cada roteador possui. No protocolo *multicast* a ocupação na rede é reduzida, pois existe a transmissão de apenas um fluxo para todos os receptores interessados. De acordo com a Figura 3.3, o protocolo *multicast* demanda um custo 6 na transmissão do conteúdo para todos os clientes da rede de distribuição.

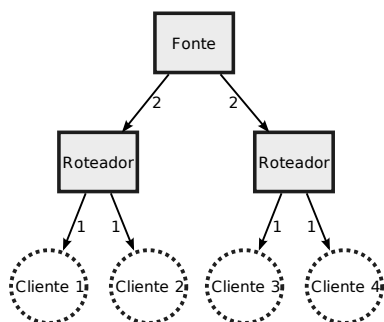


Figura 3.2: Exemplo de custos em uma rede de distribuição simples utilizando *unicast*.

Dois exemplos simples do funcionamento de protocolos P2P são apresentados na Figura 3.4, am-

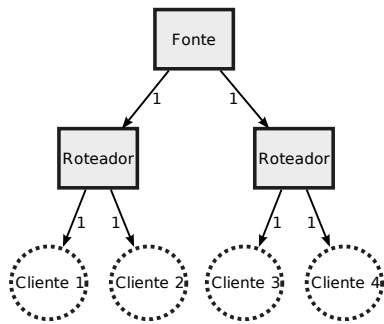


Figura 3.3: Exemplo de custos em uma rede de distribuição simples utilizando *multicast*.

bos distribuindo entre *peers* um conteúdo originalmente recebido a partir fonte. No primeiro exemplo, que demandou custo 8, a fonte entrega um fluxo para cada roteador, que por sua vez o repassa para seu primeiro cliente, este cliente então compartilha o fluxo recebido com o cliente vizinho. No segundo exemplo, que demandou custo 9, a fonte entrega um fluxo para o primeiro cliente do primeiro roteador, este cliente então inicia o compartilhamento de seu fluxo com os demais clientes através da rede de distribuição, com cada cliente distribuindo conteúdo para o próximo cliente.

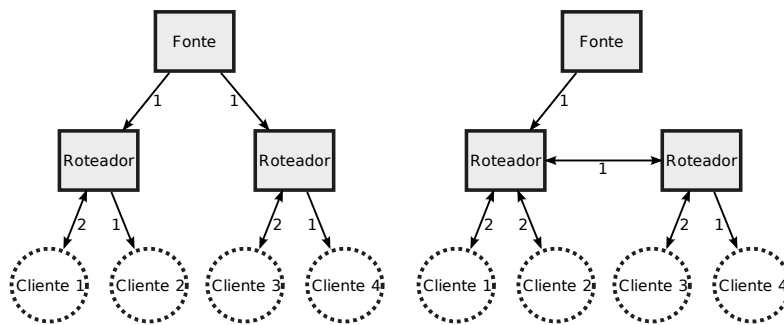


Figura 3.4: Exemplos de custos em redes de distribuição simples utilizando P2P.

O segundo exemplo da Figura 3.4 chama atenção para outro possível problema na distribuição de conteúdo IPTV através do protocolo P2P. Da forma com a qual os clientes estão dispostos neste exemplo, o fato de um cliente ter que receber uma cópia do conteúdo para depois colaborar transmitindo parte ou toda sua cópia para outros clientes, pode gerar uma grande variabilidade no atraso (*jitter*) em que cada cliente recebe o conteúdo transmitido (Ouali et al., 2008). Os protocolos *unicast* e *multicast*, por não possuírem intermediários em suas transmissões, estão menos sujeitos a gerar uma grande variação do atraso em clientes que estão próximos um do outro.

Devido a todos os motivos relacionados nesta seção, mas principalmente por não ter um bom

desempenho na otimização da carga imposta na rede, se comparado com o protocolo *multicast* (Cha et al., 2008), o protocolo P2P não faz parte das simulações realizadas neste trabalho.

3.4 Topologia e ocupação da rede

A topologia utilizada nas simulações foi inspirada na rede do Mega Grupo ¹. O grupo atende a aproximadamente seis mil clientes, entre empresas e clientes residenciais, com mais de 60 roteadores distribuídos em Minas Gerais e Espírito Santo. A partir das informações cedidas sobre a topologia do Mega Grupo, foi possível implementar no simulador um modelo de rede de distribuição real.

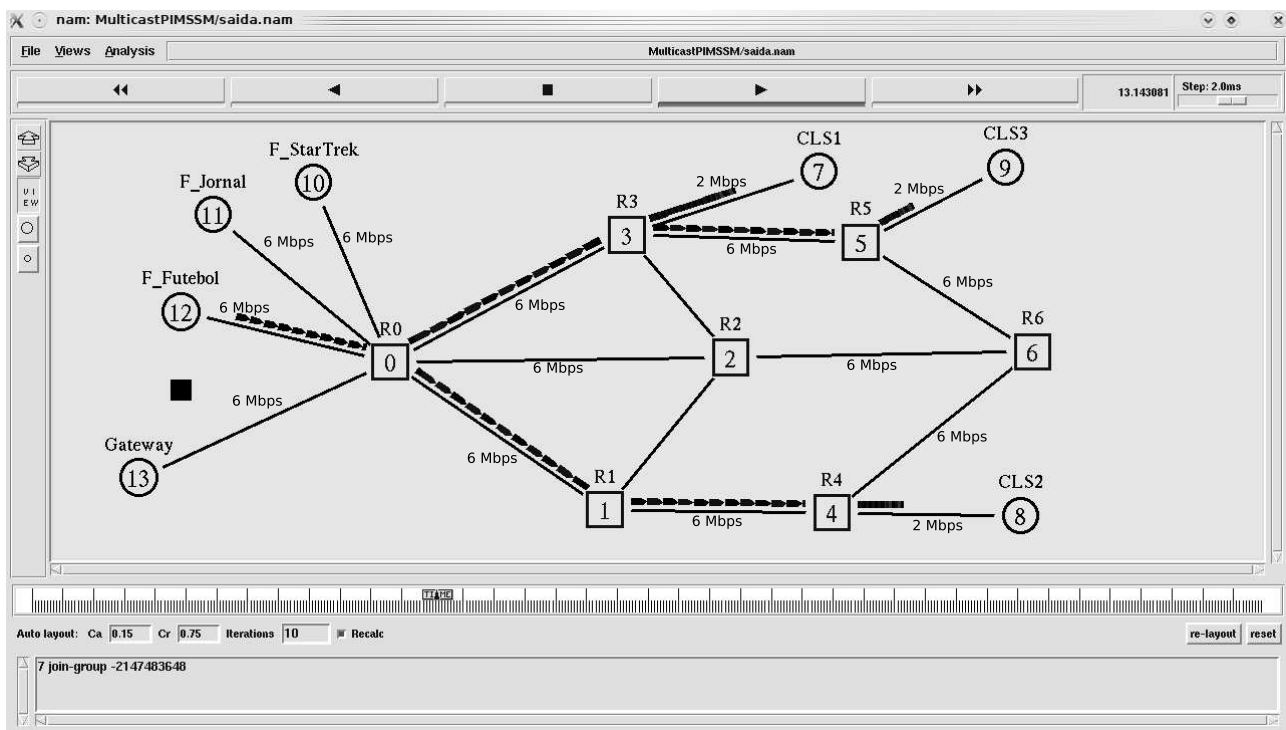


Figura 3.5: Topologia de rede com três fontes e um gateway (*screenshot* do ns-2 com as capacidades iniciais estabelecidas).

A topologia apresentada na Figura 3.6 é resultado de uma análise feita em parte da topologia da rede do grupo. Foram removidos os nós redundantes e mantido o núcleo central de distribuição

¹O Mega Grupo é uma associação que une vários WISP (*Wireless Internet Service Provider*) que atendem o leste de Minas Gerais e parte do Espírito Santo.

(*backbone*). O resultado final foi uma rede com 14 nós distribuídos entre roteadores, clientes e fontes de tráfego, como descritos abaixo:

- os nós 0, 1, 2, 3, 4, 5 e 6 são roteadores que, interligados, criam a rede de distribuição (núcleo da rede);
- os nós 7, 8 e 9 representam sub-redes que dão acesso a clientes (redes dos clientes);
- os nós 10, 11, e 12 são fontes de tráfego IPTV que provêm, respectivamente, programas de filme, jornal e futebol;
- o nó 13 representa o *gateway* padrão da rede, por onde flui todo o tráfego *background*, proveniente em sua maioria de acessos à *Internet* e VoIP.

Medições foram feitas na rede do Mega Grupo, para descobrir o perfil de tráfego existente. De posse da topologia e do perfil de tráfego desta rede real, foi possível, através de sua reprodução no simulador, estudar o comportamento da mesma quando submetida a distribuição de tráfego IPTV.

Os resultados obtidos nas medições realizadas mostraram que a rede de distribuição tem uma média de ocupação de 40% de sua capacidade máxima e que essa ocupação média representa o tráfego de *background*. O grupo não tem IPTV inserido em sua rede. Dos 40% da ocupação média, 85% é tráfego TCP proveniente dos protocolos HTTP e FTP, entre outros. Os demais 15% são UDP, a maioria proveniente de VoIP (*Voice over Internet Protocol*).

O perfil de tráfego *background* levantado foi reproduzido na rede a ser simulada, sempre do nó 13 (*gateway*) para os nós 7, 8 e 9 (clientes). Em todas as simulações realizadas foram mantidas as proporções de ocupação da rede mencionadas anteriormente. O diferencial da realidade para a simulação foi a injeção de tráfego IPTV. Para não manter um ambiente subdimensionado, onde há pouca ocupação da rede, o que não causaria necessidade alguma de otimização, o tráfego IPTV inserido correspondeu então à largura de banda ociosa. Correspondendo ao levantamento feito na rede real, o tráfego de *background* foi fixado em 40% da ocupação total da capacidade da rede, sendo o restante, 60%, utilizado para as transmissões de IPTV.

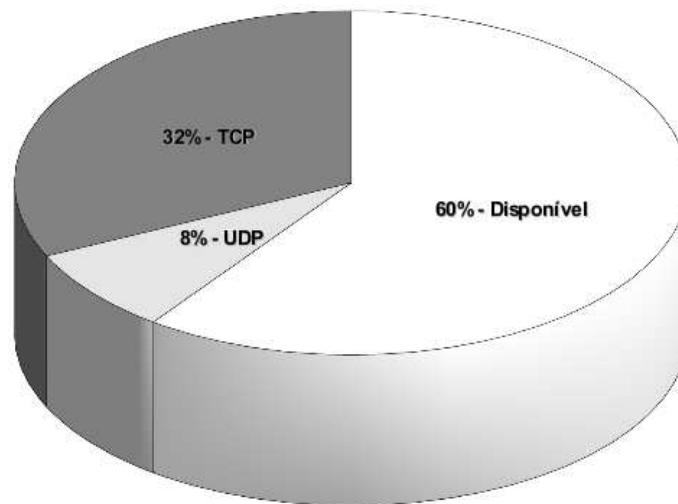


Figura 3.6: Perfil de tráfego coletado na rede do Grupo Mega.

O tráfego IPTV tem sua largura de banda impossibilitada de alteração, por ser um agregado de fluxos de tempo real, ou seja, com bit rate e demais informações medidos a partir de programas gerados continuamente (Seção 3.5). Com o intuito de sempre utilizar a capacidade máxima dos enlaces da rede de distribuição a fim de ter a possibilidade de observar detalhes do desempenho de cada protocolo, a soma dos tráfegos de IPTV e de *background* devem atingir sempre da capacidade máxima dos enlaces. Como o perfil do tráfego IPTV não pode de sofrer alterações, este corresponde à capacidade ociosa (60%) enquanto que o tráfego *background* é dimensionado a fim de demandar o restante da largura de banda (40%). Por exemplo: se a média do tráfego IPTV corresponder a 2 Mbps, o tráfego *background* deverá ser de 1,2 Mbps, onde 1,02 Mbps será TCP e 0,18 Mbps UDP. A capacidade dos enlaces da rede de distribuição é ajustada para 3,2 Mbps, ou seja, existe a possibilidade de utilização de 100% destes enlaces. Com a natural variabilidade do tráfego IPTV e a percentagem da capacidade total da rede reservada para o tráfego de *background*, ocorrem perdas e congestionamento em momentos de pico, pois o tráfego ultrapassa a capacidade da rede. Este foi considerado um bom cenário para medir a eficiência de otimização dos protocolos de distribuição abordados por este trabalho.

3.5 Geração de tráfego IPTV

A expressão VBR, do inglês *Variable Bit Rate*, diz respeito a *codecs*² que utilizam uma taxa de bits variável na compressão e descompressão de vídeo. Tais *codecs* otimizam a transmissão de vídeo digital, diminuindo ou aumentando a taxa de bits, respectivamente, para momentos de pouca ou muita movimentação no vídeo (Fan, 2007).

No simulador ns-2 existem módulos específicos para geração sintética de tráfego CBR (*Constant Bit Rate*) e VBR. Entretanto, a fim de aproximar as simulações realizadas com o ns-2 da realidade de uma transmissão de conteúdo televisivo IPTV, a geração do tráfego principal utilizou informações de bit rate que foram extraídas de programas reais de TV.

O grupo TKN (*Telecommunication Networks Group*), especializado em pesquisas sobre protocolos e arquiteturas de redes de computadores, da Faculdade de Engenharia Elétrica e Ciência da Computação da Universidade Técnica de Berlin, Alemanha, desenvolveu (Fitzek and Reisslein, 2001) a partir de programas reais de TV estatísticas para a geração de tráfego de rede, como visto em Lanfranchi and Bing (2008); Teh et al. (2007). As estatísticas extraídas dos programas são apresentadas em arquivos de texto simples, contendo o tempo em milissegundos, o valor em bytes e o tipo de cada frame. Uma rotina escrita em TCL para o ns-2 é responsável por ler o arquivo de texto, que contém uma matriz tempo por bit rate, e criar o perfil de tráfego que será inserido na rede simulada.

Três programas de categorias diferentes foram selecionados para servirem como fonte de tráfego: um filme (Star Trek First Contact), Figura 3.7, um jornal (ARD Talk), Figura 3.9, e um jogo de futebol, Figura 3.8. Os dados desses programas são apresentados na Tabela 3.1. As diferentes categorias foram selecionadas por possuírem diferentes padrões na atualização de tela, o que difere no bit rate final do tráfego: um jornal com pouca atualização, um filme com atualização média e um jogo de futebol com muita atualização.

²*Codec*: programa que permite codificar e decodificar vídeo digital.

Programa	Tempo (ms)	Total Bits	Bit Rate Médio (bps)	Bit Rate Máximo (bps)*
Filme (Star Trek - First Contact)	3603320	589335872	163568	1056368
Futebol (Soccer)	3603320	2147483647	716679	2019648
Jornal (ARD Talk)	2702200	642384920	237744	1153032

* **Bit Rate Máximo** para os intervalos de tempo utilizados neste trabalho.

Tabela 3.1: Estatísticas dos programas utilizados para simulação.

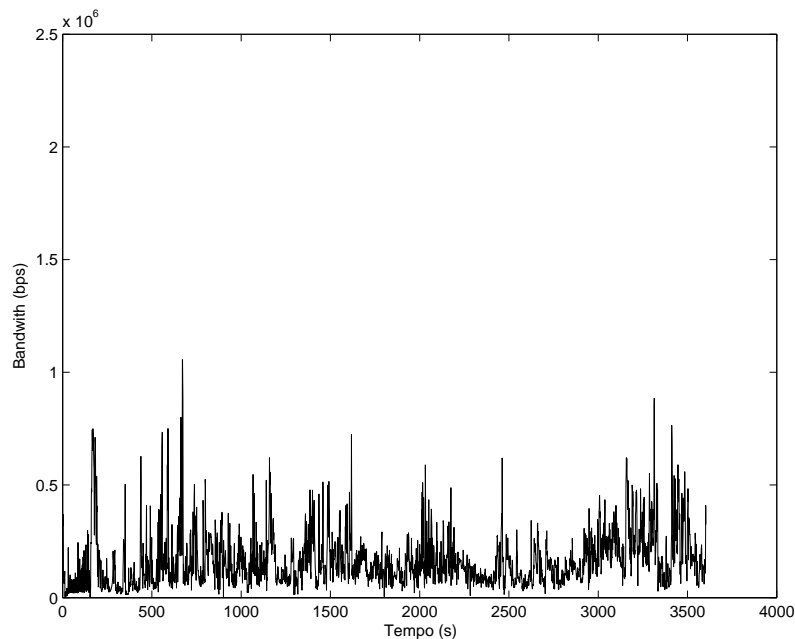


Figura 3.7: Variação do Bit Rate para o filme.

3.6 Experimentos

De posse de um ambiente de simulação capaz de aproximar a realidade de um sistema IPTV (Capítulo 3) as próximas etapas deste trabalho consistiram em avaliar cada protocolo (*unicast*, *multi-cast* PIM-DM, PIM-SM e PIM-SSM) quanto ao seu desempenho, principalmente na redução da carga imposta na rede de distribuição e também nos tempos de *zapping*. Todos os protocolos testados utilizaram suas respectivas configurações padrão no ns-2. Os experimentos realizados nesta dissertação foram divididos em três etapas a saber:

1. A primeira etapa de simulações consistiu em avaliar a discrepância de desempenho esperada en-

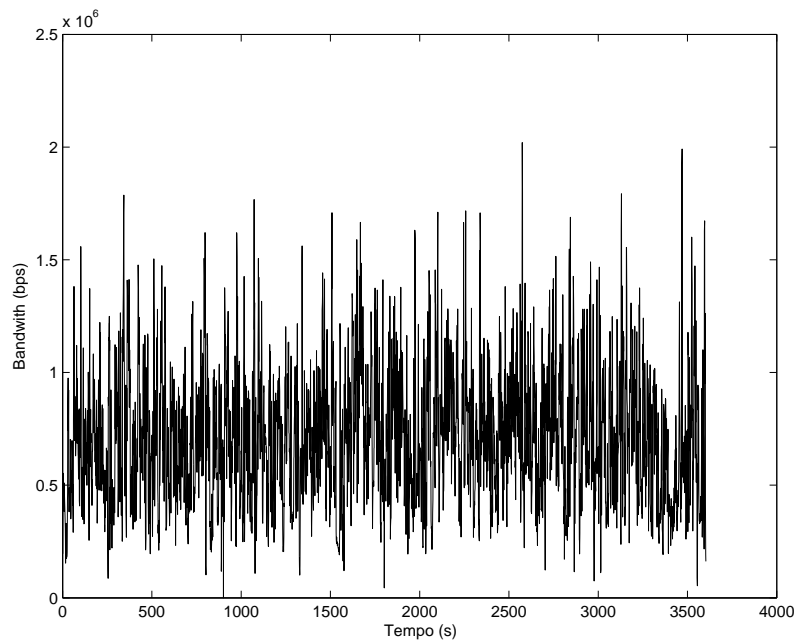


Figura 3.8: Variação do Bit Rate para o jogo de futebol.

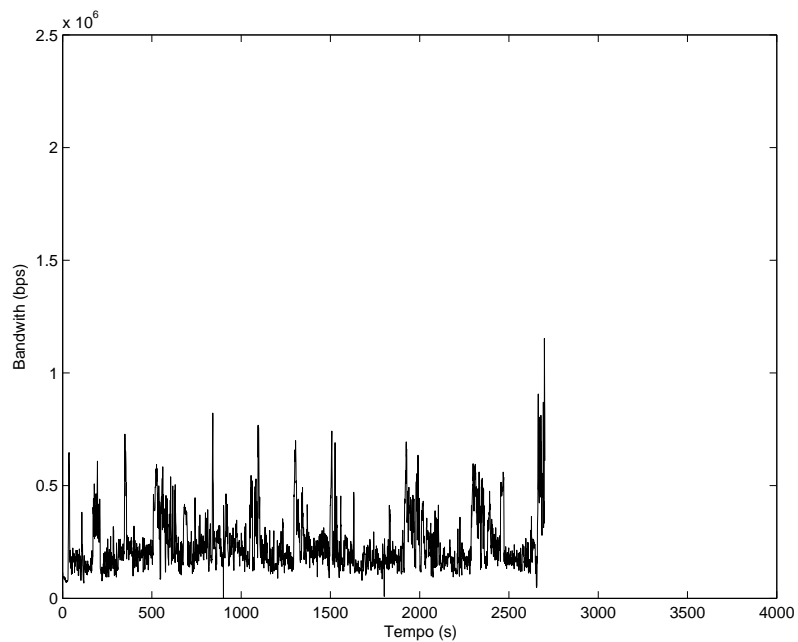


Figura 3.9: Variação do Bit Rate para o jornal.

tre os protocolos *unicast* e *multicast*. Os protocolos *multicast* otimizam a rede de distribuição, fazendo com que apenas uma cópia do fluxo transite por toda rede atendendo seus clientes. No protocolo *unicast*, o número de clientes de um fluxo corresponde ao número de cópias deste

que transitará na rede de distribuição, durante as transmissões de um programa IPTV.

2. Identificada a discrepância no desempenho do protocolo *unicast*, a próxima etapa de simulações consistiu em avaliar somente os protocolos *multicast* relacionados neste trabalho (PIM-DM, PIM-SM e PIM-SSM). Cada protocolo tem uma dinâmica diferenciada de anúncio e atendimento à requisições. Como visto nos capítulos anteriores, o PIM-DM inclui em seu anúncio parte da transmissão a ser realizada, o que aumenta a carga deste tipo de anúncio. O PIM-SM centraliza em um RP as atividades de anúncio (que não inclui um trecho da transmissão) e atendimento à requisições, fazendo com que a fonte entregue para o RP continuamente o fluxo de dados referentes aos grupos que este anuncia. O PIM-SSM também não inclui partes da transmissão em seus anúncios e tão pouco utiliza um RP. Neste modelo de serviço *multicast* o cliente faz o ingresso no grupo de transmissão enviando uma requisição diretamente à fonte escolhida. Esperou-se então identificar, nesta etapa de simulações, a dinâmica de serviço *multicast* que menos impõe carga na rede de distribuição.
3. Uma vez selecionado o protocolo (dentre os utilizados nesta dissertação) que menos impõe carga na rede de distribuição, uma nova etapa foi requerida a fim de identificar os tempos de *zapping* para os protocolos que otimizam a distribuição do tráfego em sistemas IPTV (*multicast* PIM-DM, PIM-SM e PIM-SSM). Esta simulação teve o intuito de medir o tempo entre a solicitação de um fluxo e o recebimento deste, observado a partir de um cliente. Para esta etapa, cada programa foi iniciado após o término do anterior, dando a entender que o cliente está alternando entre canais (*zapping*). Foram coletados então, em um cliente, para os momentos de *zapping*, o tempo entre a requisição de ingresso em um grupo transmissão e chegada do fluxo de dados referente à cada programa.

Uma etapa que não envolveu simulações determinou a diferença de desempenho entre protocolos P2P e *multicast*. No protocolo P2P a carga, antes imposta na fonte, é transferida para a rede de distribuição. Este protocolo utiliza uma dinâmica de compartilhamento de dados entre clientes que devolvem para a rede de distribuição parte daquilo que receberam de outros clientes, no intuito de colaborar com os demais. Os resultados de todas as etapas descritas são apresentados no Capítulo 5.

Capítulo 4

Resultados e Discussão dos Resultados

Os resultados coletados neste trabalho visam principalmente avaliar, em um primeiro momento, a carga que cada método de distribuição impõe sobre uma topologia de rede e, em um segundo momento, o tempo entre a mudança de um programa para outro, chamado tempo de *zapping*. A apresentação dos resultados então é dividida em três seções. A Seção 4.1 discute os resultados obtidos em uma simulação que envolve todos os protocolos de distribuição mencionados no Capítulo 3, com apenas um programa de TV (o filme da Seção 3.5). Na Seção 4.2 são discutidos os resultados de uma simulação que envolve os protocolos de distribuição *multicast* PIM-DM, PIM-SM, PIM-SSM e utiliza os programas apresentados na Seção 3.5. A Seção 4.3 aborda o tempo de mudança de programa, em uma simulação que também envolve os protocolos de distribuição *multicast* PIM-DM, PIM-SM, PIM-SSM e os programas da Seção 3.5.

4.1 Resultados para simulação com um único programa e protocolos *unicast* e *multicast*

Além das características do experimento citadas no Capítulo 3, esta primeira simulação envolveu:

- a simulação com o método de distribuição de tráfego *unicast* e com os métodos *multicast* PIM-DM, PIM-SM e PIM-SSM;
- a utilização apenas do filme para a geração de tráfego VBR, de acordo com os programas listados na Seção 3.5.

Nesta simulação era esperada uma discrepância no desempenho do método *unicast* em relação aos demais métodos de distribuição simulados. Devido ao método *unicast* criar um canal exclusivo para transmitir dados da fonte para cada cliente, como visto no Capítulo 2, este tem um desempenho muito inferior se comparado com o *multicast*, que compartilha um único canal para diversos clientes.

A Figura 4.1 mostra o número total de pacotes gerados na transmissão do programa (filme) durante o período de simulação. O *unicast* sobrecarregou a rede de distribuição, pois gerou múltiplos fluxos para a transferência do filme. Em comparação com a transmissão multidesinatário do *multicast* houve um aumento expressivo no número de pacotes gerados (Figura 4.1). O comportamento do protocolo *unicast* também fez com que o valor total em bytes da transmissão (Figura 4.2) e o número de pacotes perdidos (Figura 4.3) também se elevassem em comparação com o *multicast*.

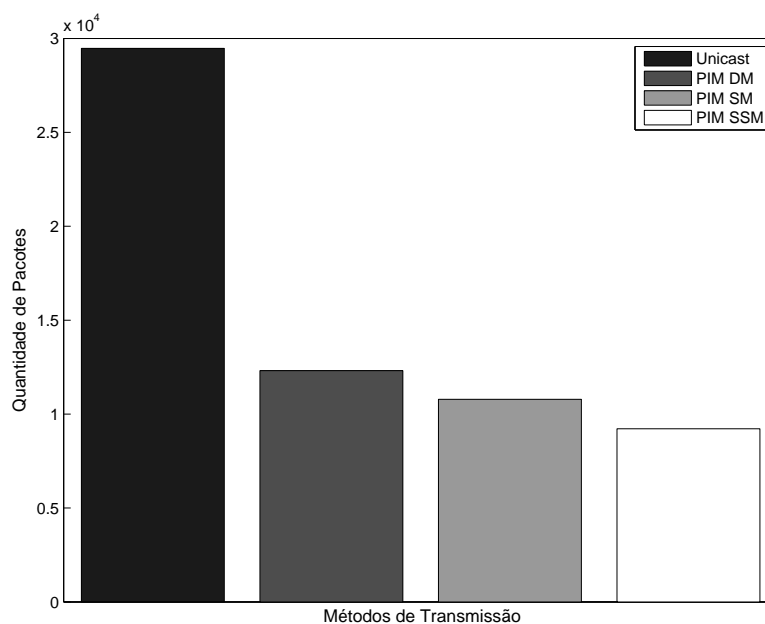


Figura 4.1: Total de pacotes na transmissão (simulação com um único programa e protocolos *unicast* e *multicast*).

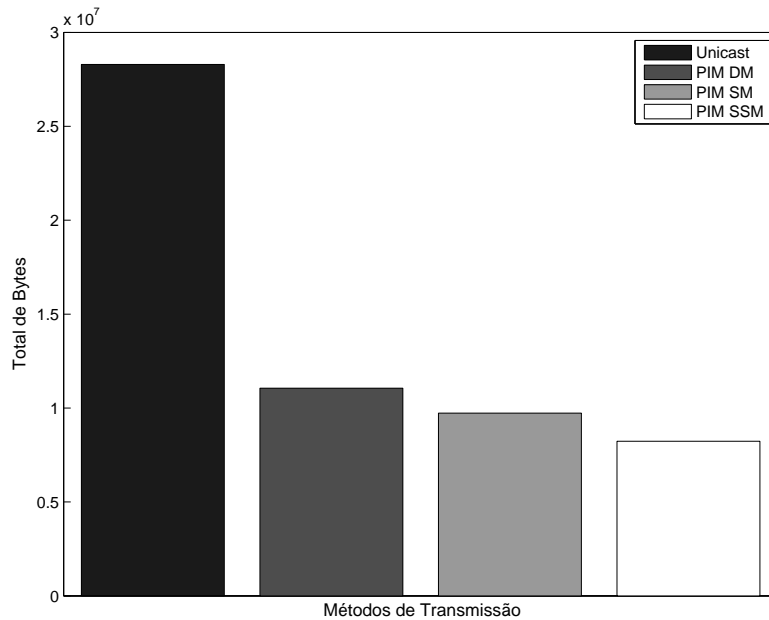


Figura 4.2: Total de bytes na transmissão (simulação com um único programa e protocolos *unicast* e *multicast*).

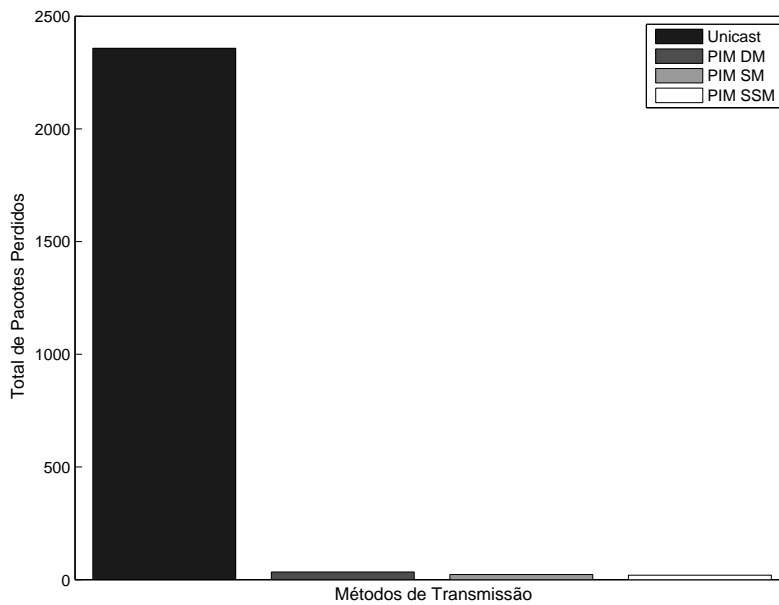


Figura 4.3: Total de pacotes perdidos na transmissão (simulação com um único programa e protocolos *unicast* e *multicast*).

A Figura 4.4 mostra a soma cumulativa, em bytes, da transmissão do programa em função do tempo. Um comportamento relevante visto nesta figura é o desempenho do *unicast* em momentos onde não há requisições do programa por parte dos clientes. A transmissão do programa é iniciada no

momento 10 e termina no momento 30 de uma simulação de 40 segundos, como visto no Capítulo 3.

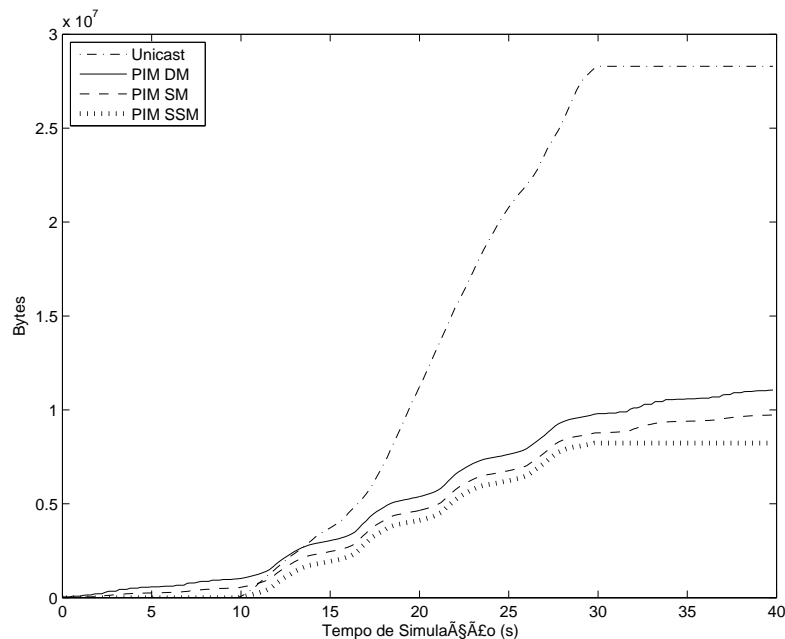


Figura 4.4: Soma cumulativa do tráfego na transmissão em função do tempo na simulação com um único programa e protocolos *unicast* e *multicast*.

Nos momentos de 0 a 10 e de 30 a 40 não há nenhuma requisição ao programa, o que no caso do *unicast* e do PIM-SSM foi representado por um período sem nenhuma transferência de fluxo, Figura 4.5. A Figura 4.5 apresenta a vazão da rede em função do tempo para todos os protocolos. Enquanto os protocolos *multicast* tendem a seguir um padrão, diferenciando-se apenas em momentos onde não houve requisição ao filme, o *unicast*, por ter gerado múltiplos fluxos, ocasionou uma utilização maior da rede durante o tempo de requisição ao filme.

Nas Figuras 4.4 e Figura 4.5 é possível perceber o melhor desempenho do *unicast* e do PIM-SSM em relação aos demais protocolos de distribuição nos períodos onde não há requisição. Nos casos do *unicast* e do PIM-SSM não há necessidade de anúncios constantes, como no PIM-DM, tão pouco a existência de um RP, presente no PIM-SM. Isso faz com que estes dois protocolos não realizem nenhum tipo de transferência em períodos sem requisições a um programa. Mesmo com o bom desempenho em períodos sem requisições, o *unicast* tem um péssimo desempenho com o nível de sobrecarga que impõe à rede. Na simulação realizada, o *unicast* ocupou a rede cerca de três vezes mais que os demais protocolos, deixando claro que não é uma boa opção para a distribuição de tráfego

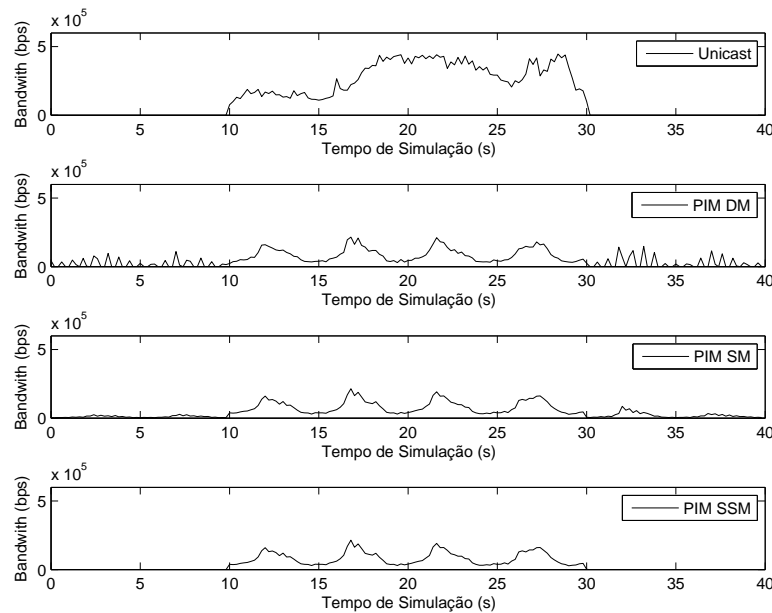


Figura 4.5: Vazão da rede na simulação com um único programa e protocolos *unicast* e *multicast*.

IPTV.

A escolha da largura de banda entre os nós da rede é feita a partir da média de ocupação que cada protocolo exige na transmissão dos programas. Na simulação apresentada na Seção 4.1, o protocolo *unicast*, por ocupar a rede cerca de três vezes mais que o protocolo *multicast*, superou a largura de banda definida na topologia. A largura de banda definida entre a média de tráfego *unicast* e *multicast*, fez com que a rede ficasse subdimensionada para o *unicast* e superdimensionada para o *multicast*. Um reflexo deste superdimensionamento pode ser visto na Figura 4.3, onde o protocolo *unicast* teve dificuldades na entrega de todos os pacotes enquanto os demais protocolos tiveram perdas irrisórias.

4.2 Resultados para simulação com três programas ocupando a rede simultaneamente

A simulação apresentada nesta seção seguiu exatamente as normas apresentadas no Capítulo 3. O dimensionamento da rede foi feito com base apenas na necessidade dos protocolos *multicast* simula-

dos, PIM-DM, PIM-SM e PIM-SSM. Isso fez com que a rede reagisse a qualquer transferência fora da média estabelecida, resultando principalmente em perdas de pacotes.

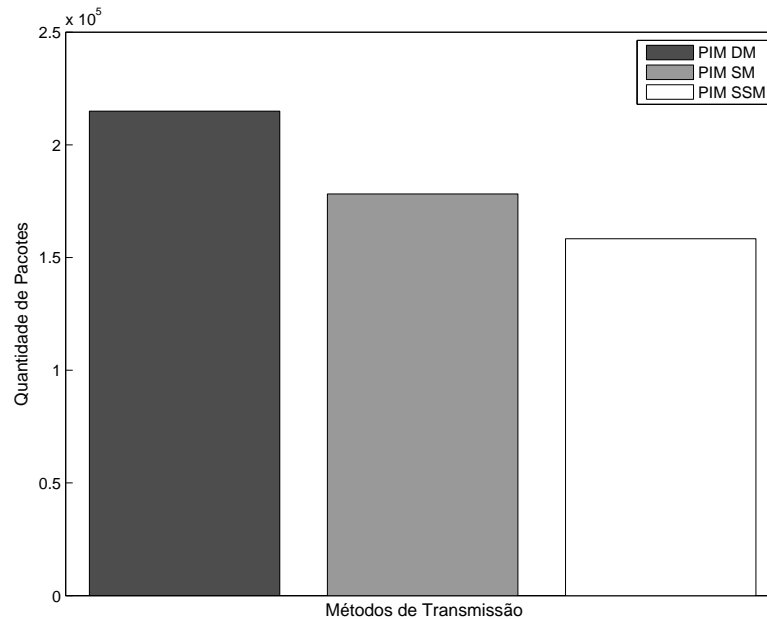


Figura 4.6: Total de pacotes na transmissão (simulação com três programas ocupando a rede simultaneamente).

Os resultados apresentados nas Figuras 4.6 e 4.7 mostram a carga que cada protocolo *multicast* impôs sobre a rede durante todo o tempo de simulação. A soma cumulativa de *bytes* transmitidos durante o período de simulação também pode ser vista na Figura 4.8. Esta figura deixa claro que a diferença de carga imposta na rede pelos protocolos *multicast*, acontece principalmente nos momentos onde só há o anúncio do serviço e não na transmissão do programa.

Na Figura 4.9 é possível observar detalhadamente o comportamento de cada protocolo nos momentos onde só existem os anúncios do serviço, do segundo 0 a 10 e de 30 a 40. O protocolo PIM-DM anuncia periodicamente para todos os nós da rede a existência de seus programas. Em cada anúncio, este protocolo envia um trecho atual da transmissão, pois assume que todos os nós da rede estão interessados no tráfego. No PIM-DM, como visto no Capítulo 2, o cliente deve enviar uma mensagem *prune* após receber o anúncio, para negar o ingresso ao grupo de transmissão. Caso a resposta seja uma mensagem *join*, o cliente passa a receber a transmissão por ter ingressado no grupo. No caso de uma mensagem *join*, o trecho que foi enviado junto ao anúncio pode ser usado para reduzir o tempo

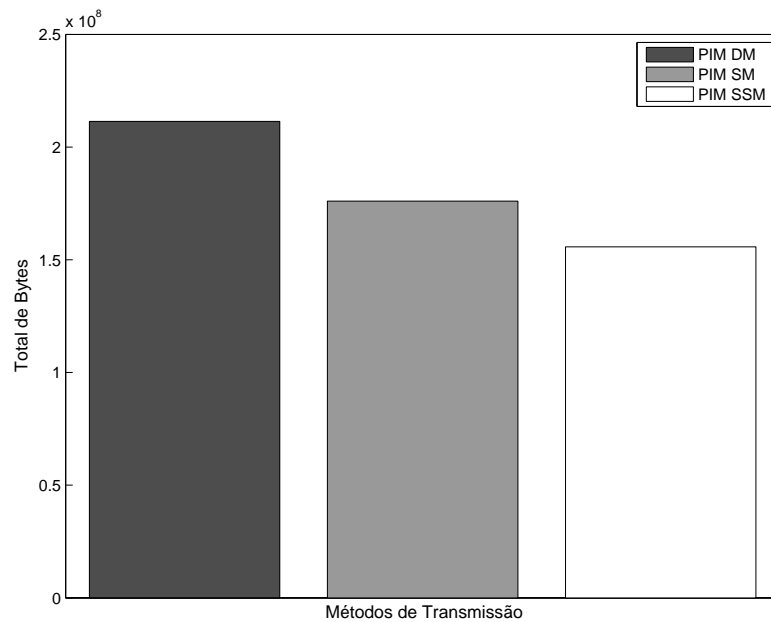


Figura 4.7: Total de bytes na transmissão (simulação com três programas ocupando a rede simultaneamente).

em que o cliente recebe a transferência vinda da fonte após seu ingresso no grupo.

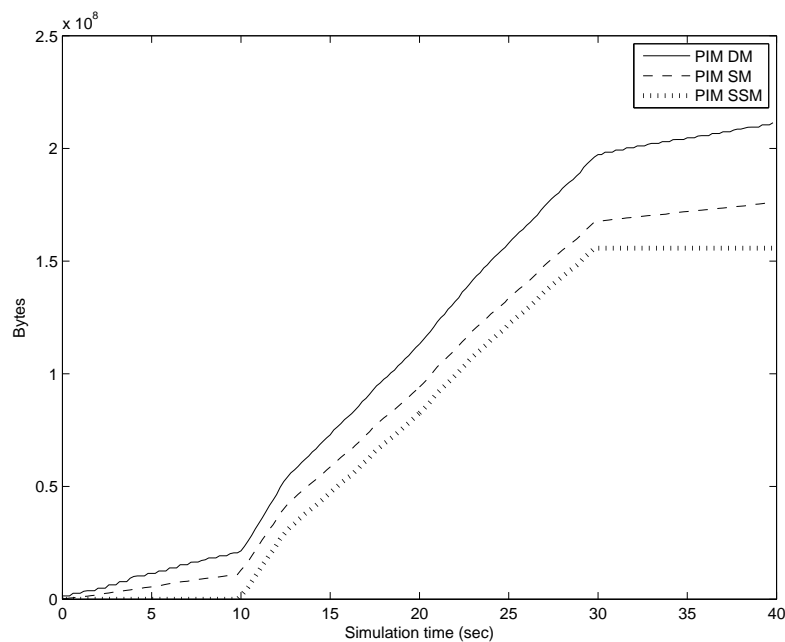


Figura 4.8: Soma cumulativa do tráfego da transmissão em função do tempo na simulação com três programas ocupando a rede simultaneamente.

A Figura 4.9 ainda exhibe o comportamento dos protocolos PIM-SM e PIM-SSM. No PIM-SM

(Capítulo 2), o anúncio não contém parte da transmissão como no PIM-DM. No PIM-SM um roteador é eleito para ser o RP e a fonte transmite continuamente seu programa para o RP eleito, que por sua vez atende aos clientes que fazem parte do grupo *multicast*. Na topologia adotada nas simulações, de acordo com o Capítulo 3, o RP é vizinho da fonte e a transmissão contínua da fonte para o RP ocupa irrisoriamente a rede (Figura 4.9, PIM-SM) devido à ocupação de apenas um enlace.

O PIM-SSM, conforme a Figura 4.9, tem o melhor desempenho nos momentos onde só existem anúncios. No PIM-SSM os anúncios não são acompanhados de trechos da transmissão e também não há a necessidade de um RP na rede de distribuição. Segundo explicado no Capítulo 3, no PIM-SSM a transmissão parte da fonte diretamente para o cliente, após este ingressar no grupo por meio de uma mensagem IGMP (*join*). A diferença de desempenho dos protocolos PIM-DM, PIM-SM e PIM-SSM, apresentadas nas Figuras 4.6, 4.7 e 4.8, é um reflexo do comportamento de cada protocolo nos momentos onde só há o anúncio do grupo na rede.

O caminho tomado pelas transmissões *multicast* é exibido na Figura 4.10. As três fontes de tráfego (filme, jornal e futebol) enviam seus fluxo *multicast* para o roteador R0 e a partir deste, o fluxo é repassado para os demais roteadores e posteriormente para os clientes.

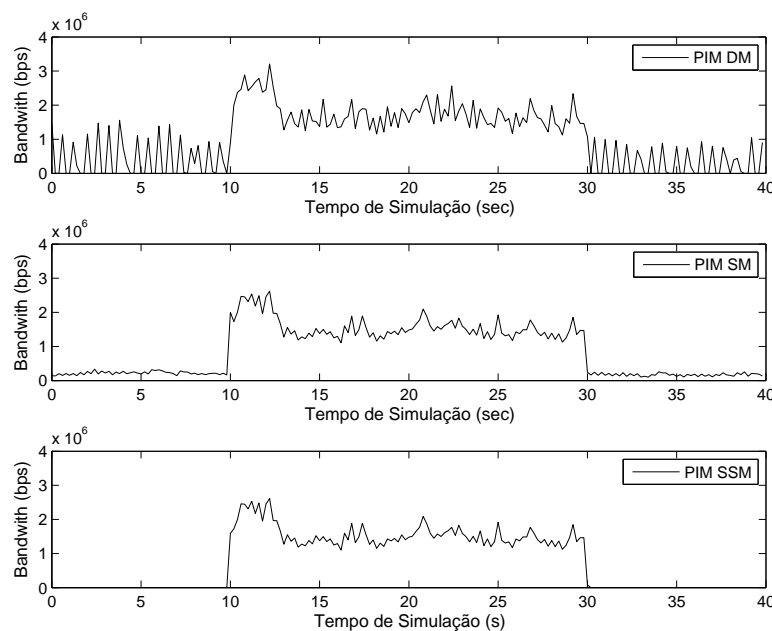


Figura 4.9: Vazão da rede na simulação com três programas ocupando a rede simultaneamente.

O protocolo PIM-SM, por transmitir continuamente seu programa para o RP, ocupou o roteador R0 (Capítulo 3) durante todo o período de simulação, ao contrário dos protocolos PIM-DM e PIM-SSM. A ocupação contínua do roteador R0, que é o roteador que dá acesso a todos os nós da topologia, aumenta a probabilidade de perda de pacotes em momentos de picos do tráfego VBR. A perda de pacotes ocorreu em todos os protocolos devido aos enlaces da topologia terem sido ajustados exatamente para transferir a média do tráfego VBR mais tráfego *background*, segundo Capítulo 3. O tráfego de *background*, além de contínuo, é uniforme. Ele parte do *gateway* em direção aos clientes e o roteador R0 é utilizado na ligação entre *gateway* e clientes. No PIM-SM, em momentos onde o tráfego VBR tem picos na transmissão, há a perda de pacotes no roteador R0, o que justifica uma perda maior de pacotes neste protocolo em relação aos demais, Figura 4.11.

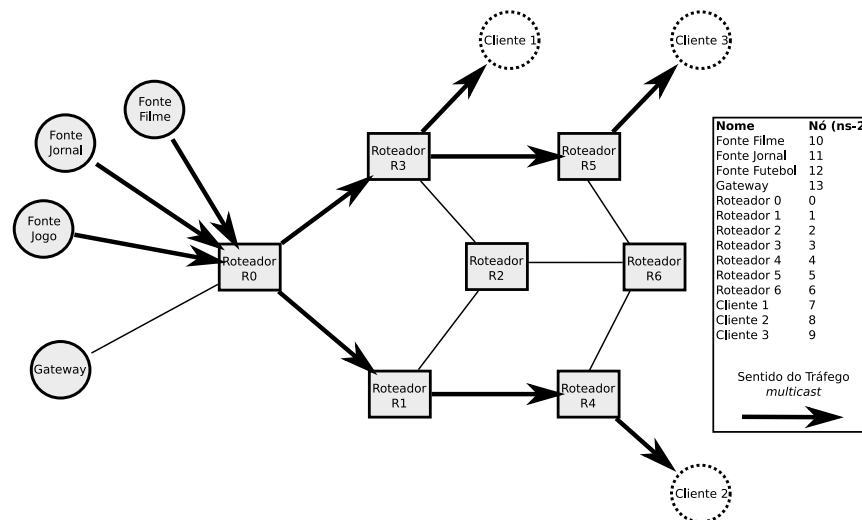


Figura 4.10: Caminho tomado pelas transmissões *multicast* na topologia simulada.

Um comportamento percebido nos protocolos *multicast* foi que a variação do perfil de tráfego dos métodos de difusão acontece principalmente em momentos onde só existem os anúncios dos programas. Durante uma transferência, como visto no período de 10 a 30 segundos da Figura 4.9, os protocolos *multicast* mantiveram um padrão de transmissão semelhante. Foi realizada então uma análise da vazão percebida por um cliente na rede, a fim de comparar o perfil das transmissões *multicast*. A Figura 4.12 apresenta a vazão entre o roteador R4 e o cliente 2, com apenas os FIDs¹ dos programas transmitidos, retirando totalmente o tráfego de *background*. É possível perceber que a

¹FID (*Flow ID*) diz respeito ao número de identificação de um fluxo de dados na rede. No contexto IPTV, um FID pode ser considerado como o fluxo de dados de um programa.

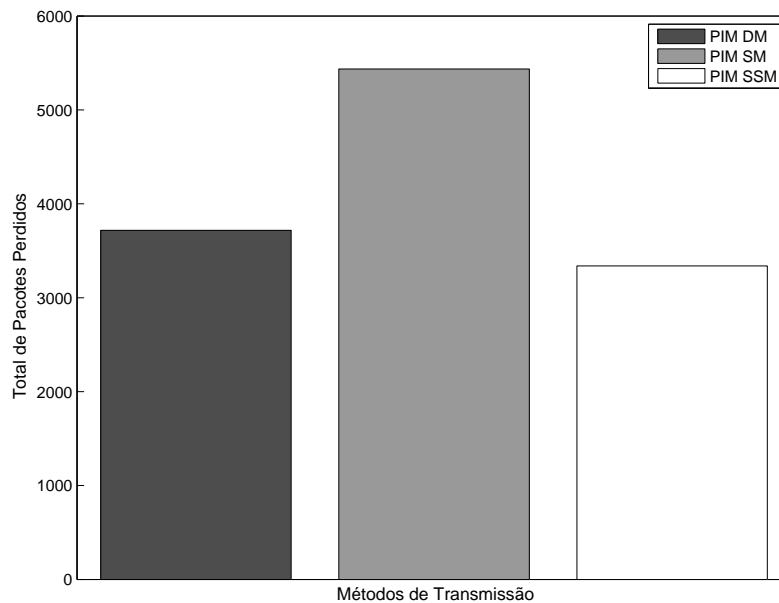


Figura 4.11: Total de pacotes perdidos na simulação com três programas ocupando a rede simultaneamente.

diferença entre PIM-DM, PIM-SM e PIM-SSM realmente só existe em momentos de anúncio. Uma vez que um cliente entra para o grupo de transmissão *multicast* existente na rede, o cliente deixa de receber o anúncio deste serviço e as transferências dos dados realizadas por PIM-DM, PIM-SM e PIM-SSM seguem exatamente o mesmo perfil.

4.3 Resultados para zapping em simulação com três programas

A simulação apresentada nesta seção foi realizada com o intuito de medir o tempo entre mudanças de canal, chamado *zapping*. Em uma rede com *multicast*, um exemplo de *zapping* consiste no cliente enviar uma mensagem *prune* para um grupo referente a um programa em que esteja inscrito e, em seguida, uma mensagem *join* para outro grupo, a fim de se inscrever em novo programa. A ordem das mensagens *join* e *prune* pode ser alterada. Dessa forma, como visto no Capítulo 2, através do protocolo IGMP o cliente encarregará o roteador vizinho de cancelar sua inscrição em um grupo *multicast* e em seguida inscrever o cliente em outro (Uzunalioglu, 2009; Sasaki et al., 2008).

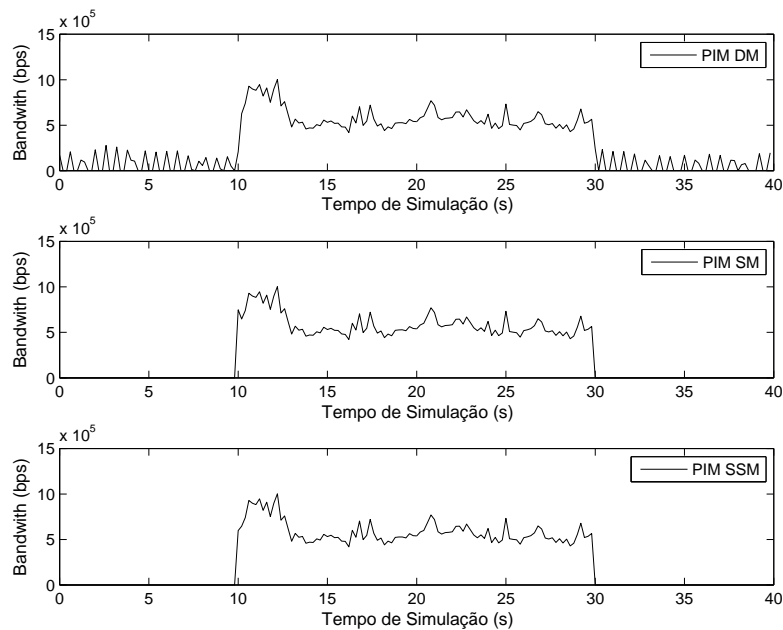


Figura 4.12: Vazão entre os nós 4 e 8 com a soma dos FIDs relativos aos três programas simulados.

Na simulação realizada, novamente seguindo os padrões estabelecidos no Capítulo 3, os programas filme, jornal e futebol (Seção 3.5) são iniciados nesta ordem, mas em intervalos distintos. Em uma simulação de 40 segundos, o filme é iniciado no momento 6 e encerrado no momento 16, o jornal iniciado no momento 16 e encerrado no momento 26 e o jogo de futebol iniciado no momento 26 e encerrado no momento 36. Para esta simulação, o tempo entre o início de um programa e a chegada da primeira rajada de tráfego referente a ele, corresponde ao tempo de *zapping*. A coleta dos dados referente à rajada de cada programa foi realizada no enlace que liga o roteador 4 ao cliente 2, de acordo com a topologia proposta no Capítulo 3 e vista na Figura 4.13. Na simulação de *zapping*, como apresentado na Figura 4.13, para os períodos de simulação de 0 à 6 e de 36 à 40 segundos, não há requisições a nenhum dos programas. Nesses períodos só existem os anúncios dos serviços.

As Figuras 4.14, 4.15 e 4.16 são resultados da coleta de tráfego realizada entre o roteador 4 e o cliente 2. Nessas figuras existem três diferentes perfis de tráfego presentes em intervalos distintos de tempo. Tomando como exemplo as Figuras 4.15 e 4.16, é possível perceber claramente o início e término de cada tráfego, referente aos programas filme, jornal e jogo, inicializados nesta sequência. Para o tráfego de cada programa foi identificado o momento onde o cliente 2 recebeu o início transferência após a solicitação de ingresso no grupo *multicast*.

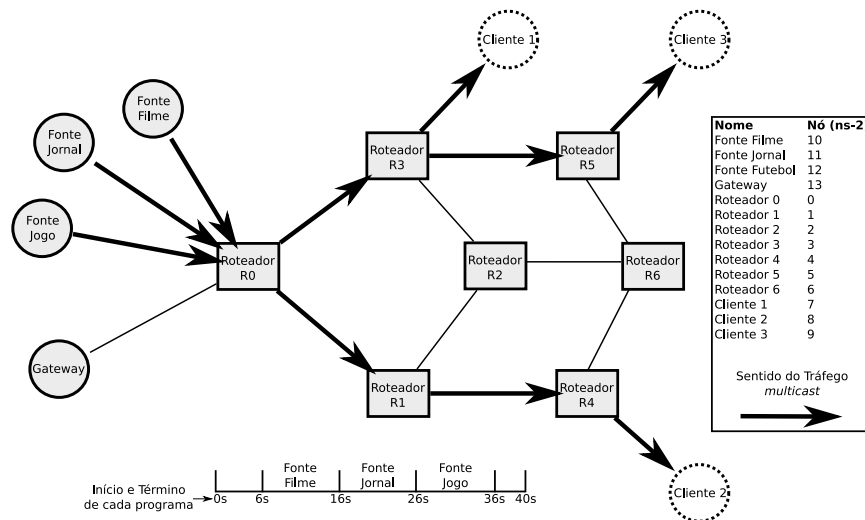


Figura 4.13: Topologia com a linha do tempo de início e fim de cada programa.

Na Figura 4.14, o resultado da simulação com PIM-DM difere ligeiramente das Figuras 4.15 e 4.16, para as simulações de PIM-SM e PIM-SSM, respectivamente. Durante a simulação de *zapping* com o PIM-DM, o que também foi observado nas simulação anteriores, o cliente 2 recebeu, durante os períodos de 0 à 6 e de 36 à 40 segundos, o anúncio contendo o trecho inicial da transferência de todos os programas. No período entre 6 e 16, durante a transmissão do filme, o cliente 2 permaneceu recebendo o anúncio seguido do início da transferência dos programas jornal e futebol. Esse comportamento também ocorreu na transmissão dos outros programas. Durante a transmissão do jornal, o cliente 2 recebeu o anúncio dos programas filme e futebol. E finalmente, durante a transmissão do programa futebol, o cliente 2 recebeu os anúncios dos programas filme e jornal. Em resumo, o cliente 2 só não recebia o anúncio do programa ao qual estava inscrito. Esse comportamento do PIM-DM gerou uma espécie de “ruído” na Figura 4.14. Este ruído é inerente aos insistentes anúncios que, no caso do PIM-DM, contêm parte do tráfego a ser transmitido naquele momento.

No resultado da simulação de *zapping* com o PIM-DM, Figura 4.14, o cliente 2 recebeu o fluxo do programa filme no momento 6,4, para uma solicitação realizada no momento 6. Os tempos de chegada dos programas jornal e futebol foram 16,6 e 26,4 para solicitações realizadas nos momentos 16 e 26, respectivamente.

Na Figura 4.15, proveniente das simulação utilizando PIM-SM, o cliente recebeu o fluxo de dados

referente ao filme no momento 6,2, sendo que sua requisição foi feita no momento 6. Para o jornal, o fluxo foi recebido no momento 16,2 sendo que solicitação ocorreu no momento 16, e para o jogo o fluxo foi recebido no momento 26,3 para uma solicitação feita no momento 26.

Para a simulação com PIM-SSM, Figura 4.16, o recebimento do fluxo do programa filme aconteceu no momento 6,2 para a solicitação feita no momento 6. O jornal foi recebido no momento 16,2 com a solicitação feita no momento 16 e o fluxo do jogo chegou ao cliente 2 no momento 26,2 para uma solicitação feita no momento 26.

Cada uma das simulações de *zapping* foi repetida seis vezes e calculadas a média, o desvio padrão e o intervalo de confiança para as medições obtidas. Nessas simulações foi observado o tempo de chegada do primeiro pacote de um programa após o envio da requisição *join* IGMP. O resultado das seis simulações para cada um dos programas e protocolos *multicast* é exibido na Tabela 4.2.

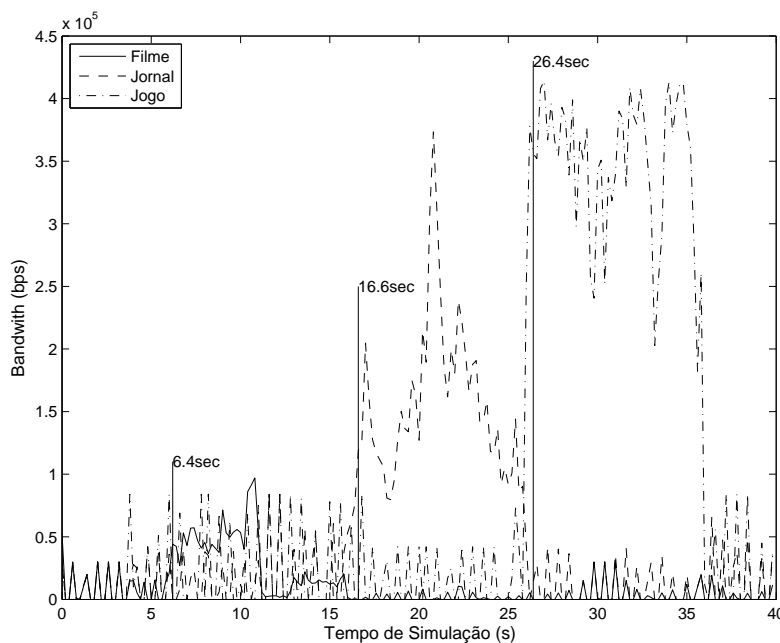


Figura 4.14: Tempo de *zapping* na simulação com *multicast* PIM-DM.

Durante a medição dos tempos entre o ingresso no grupo de transmissão de um programa e a chegada da primeira rajada de tráfego correspondente deste (*zapping*), o protocolo *multicast* PIM-DM foi o que apresentou pior desempenho, de acordo com a Figura 4.14. Como visto na Figura 4.8, a carga que o protocolo *multicast* PIM-DM impõe na rede de distribuição é maior que a imposta pelos

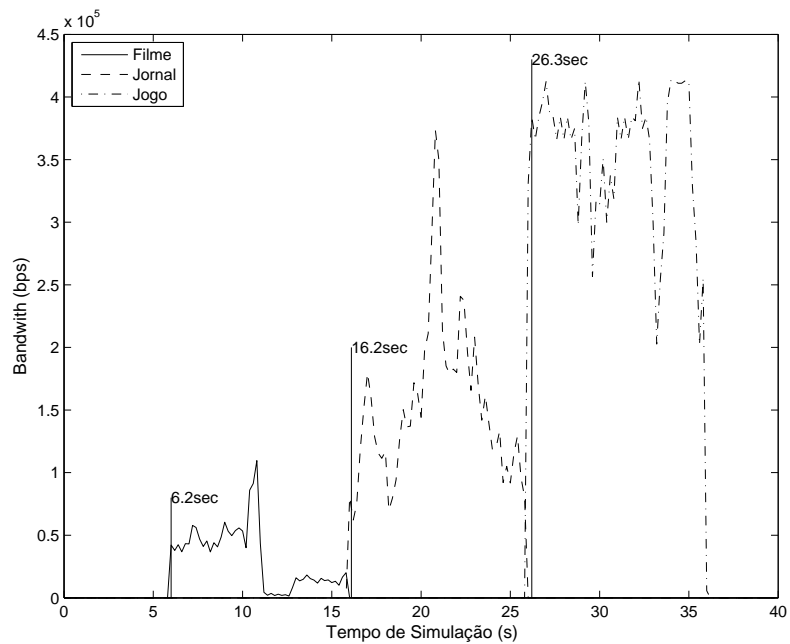


Figura 4.15: Tempo de *zapping* na simulação com *multicast* PIM-SM.

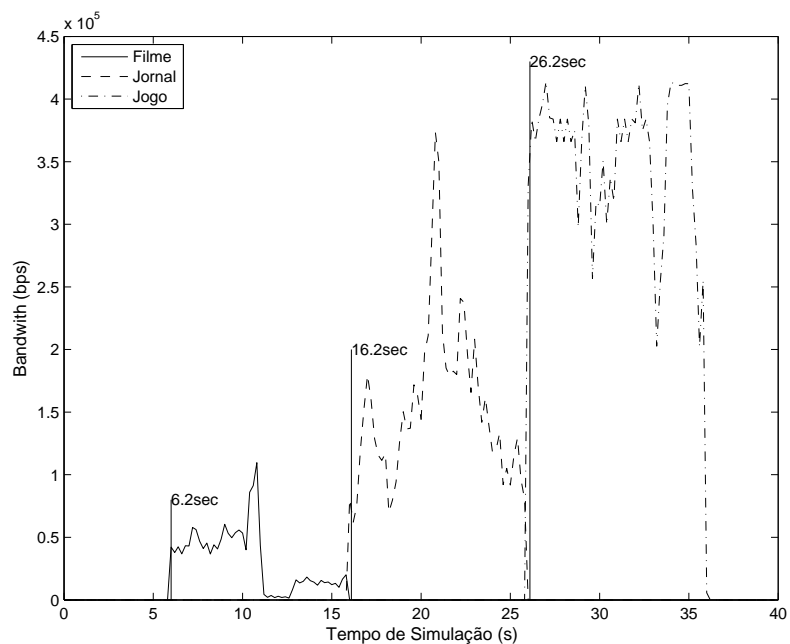


Figura 4.16: Tempo de *zapping* na simulação com *multicast* PIM-SSM.

protocolos PIM-SM e PIM-SSM, devido ao seu modo de anúncio. No anúncio do PIM-DM, por este protocolo assumir que todos estão interessados em seu conteúdo, é enviado juntamente com o anúncio o início da transferência. Por este motivo, uma possível explicação para o tempo de *zapping* no PIM-

DM ser maior que os demais protocolos é o fato de este protocolo congestionar mais facilmente a rede, aumentando assim o atraso.

Os tempos de *zapping* para os protocolos PIM-SM e PIM-SSM, Figuras 4.15 e 4.16 respectivamente, foram próximos um do outro (Tabela 4.1). O PIM-SM e PIM-SSM utilizam um método de anúncio semelhante e assumem que todos os nós da rede não estão interessados em um grupo de transmissão. A diferença entre estes protocolos, como visto no Capítulo 2, é apenas na troca do RP existente no PIM-SM pela transferência direta a partir da fonte, como a realizada no PIM-SSM. A diferença entre PIM-SM e PIM-SSM não é suficiente para alterar o perfil de tráfego, como apresentado na Figura 4.8. Por este motivo, não foi observada qualquer discrepância nos resultados de tempo de *zapping* entre PIM-SM e PIM-SSM. Já no PIM-DM o tempo de *zapping* foi relativamente maior.

Protocolo	1º zapping (s)	2º zapping (s)	3º zapping (s)
PIM-DM	0,4	0,6	0,4
PIM-SM	0,2	0,2	0,3
PIM-SSM	0,2	0,2	0,2

Tabela 4.1: Tempo de *zapping* para os protocolos *multicast*.

Para classificar os resultados de tempos de *zapping* obtidos nesta simulação, foi utilizado o índice MOS (*Mean Opinion Score*) de qualidade subjetiva. O índice MOS é um padrão numérico usado para medir a qualidade da voz/vídeo após a compressão e/ou transmissão. O trabalho proposto por Kooij et al. (2006), baseado na norma ITU-T (2006), utilizado também em Jennehag and Pettersson (2008); Kim et al. (2008), apresenta um método para relacionar o tempo de *zapping* com o índice MOS de qualidade subjetiva, descrito na Tabela 4.4. Relacionando a coluna MOS da Tabela 4.3, resultado do trabalho proposto em Kooij et al. (2006), com a Tabela 4.4, é possível obter uma classificação para os tempos de *zapping* obtidos com os protocolos simulados.

De acordo com a Tabela 4.1, que apresenta todos os resultados de tempos de *zapping* obtidos durante a simulação, as classificações variaram de regular, no PIM-DM, para bom, no PIM-SM e PIM-SSM. O protocolo PIM-DM, que obteve uma média no tempo de *zapping* de 0,46 segundos, obteve com esse tempo um índice MOS de 3,5, de acordo com a Tabela 4.3. Com o índice MOS de 3,5 o PIM-DM foi classificado como “regular” de acordo com o Tabela 4.4. Seguindo o mesmo

	PIM-DM (s)	PIM-SM (s)	PIM-SSM (s)
Simulações Filme*	0,112330	0,019498	0,018354
	0,058354	0,024265	0,018354
	0,032966	0,019901	0,018354
	0,031945	0,022280	0,021351
	0,031982	0,020826	0,018949
	0,100322	0,031327	0,020170
Média	0,061317	0,023016	0,019255
Desvio Padrão	0,029735	0,004561	0,001302
Intervalo de Confiança	0,023793	0,003649	0,001042
Simulações Jornal*	0,042225	0,019743	0,018798
	0,033046	0,018362	0,018354
	0,033488	0,019751	0,018354
	0,031671	0,018362	0,018354
	0,098354	0,018362	0,018354
	0,032850	0,019943	0,018354
Média	0,045272	0,019087	0,018428
Desvio Padrão	0,026285	0,000798	0,000181
Intervalo de Confiança	0,021032	0,000638	0,000145
Simulações Jogo*	0,034028	0,018362	0,018354
	0,058354	0,058362	0,018354
	0,048748	0,075248	0,021561
	0,160314	0,066219	0,021706
	0,158798	0,018362	0,067432
	0,040949	0,018362	0,058970
Média	0,083532	0,042486	0,034396
Desvio Padrão	0,059445	0,026961	0,022520
Intervalo de Confiança	0,047565	0,021573	0,018019

* Valores obtidos em cada uma das seis simulações realizadas.

Tabela 4.2: Valores obtidos para o instante de tempo de chegada do primeiro pacote referente a um programa entregue após a requisição *join* IGMP.

procedimento realizado no protocolo PIM-DM para classificar os protocolos PIM-SM e PIM-SSM, estes obtiveram um índice MOS de 4, 6 e foram classificados como “bom”, de acordo com o Tabela 4.4.

Tempo de zapping (s)	MOS	Desvio padrão da escala MOS
0	4,9	0,30
0,1	4,9	0,36
0,2	4,6	0,50
0,2	4,5	0,51
0,5	3,5	0,60
0,5	3,3	0,48
1	2,3	0,66
2	1,6	0,92
2	2,0	0,71
5	1,1	0,44

Tabela 4.3: Pontuação na escala MOS segundo tempo de zapping.

MOS	Avaliação
5	Excelente
4	Bom
3	Regular
2	Ruim
1	Péssimo

Tabela 4.4: Avaliação da escala MOS segundo pontuação obtida pelo tempo de zapping.

Acredita-se que um tempo de zapping entre 0 e 0,1 segundos (“excelente” de acordo com a classificação MOS) proporciona ao cliente uma experiência de instantaneidade na troca de programas. Sem dúvidas esta experiência é a melhor que um cliente pode ter. Esse mesmo cliente perceberia uma ligeira lentidão, na classificação “bom” segundo índice MOS, devido o tempo de zapping 0,2 segundos. Acredita-se que nesta classificação (“bom”) o cliente não tenha uma percepção de demora, mas também não tem a troca instantânea de canais como na classificação “excelente”. A classificação “regular” (0,5 segundos) tem um tempo considerável na troca de programas se comparado com as experiências obtidas nas classificações “excelente” e “bom”. A partir da classificação “regular” o cliente certamente perceberá o sistema lento e obviamente a experiência do cliente tende a piorar caso haja acréscimos no tempo de zapping.

4.4 Protocolos P2P

A simulação dos protocolos P2P foi considerada desnecessária, pois foi constatado que o desempenho destes não se iguala ao de um protocolo *multicast*, tal como discutido no Capítulo 3. Adaptando o raciocínio exposto em Cha et al. (2008) à topologia específica empregada nas simulações (Figura 3.6), demonstra-se que os protocolos *unicast* e P2P têm custos de transmissão semelhantes. Considerando que a rede tenha custo 1 para cada enlace, e que um programa deve ser transmitido da fonte, nó 10, para todos os clientes, nós 7, 8 e 9, é possível chegar às seguintes conclusões:

- o protocolo *unicast*, por criar um canal único para cada transmissão entre fonte e cliente, ocupa o somatório de 11 enlaces na realização da transmissão, a saber:
 - na transmissão da fonte para o nó 7, são ocupados 3 enlaces: 10-0, 0-3 e 3-7;
 - da fonte para o nó 8, são ocupados mais 4 enlaces: 10-0, 0-1, 1-4 e 4-8;
 - e da fonte para o nó 9, são ocupados 4 enlaces: 10-0, 0-3, 3-5 e 5-7.
- no protocolo P2P, há o compartilhamento do conteúdo recebido por um cliente entre os demais através de fluxos *unicast*. Nesta topologia o nó 7 recebe o conteúdo vindo da fonte e compartilha com os nós 9 e 8. Este protocolo também produz custo 11:
 - na transmissão da fonte para o nó 8 são ocupados 3 enlaces: 10-0, 0-3 e 3-7;
 - no compartilhamento do conteúdo recebido pelo nó 8 com o nó 9 são ocupados 3 enlaces: 7-3, 3-5 e 5-9;
 - no compartilhamento do conteúdo recebido pelo nó 9 com o nó 8 são ocupados mais 4 enlaces: 9-5, 5-6, 6-4 e 4-8.
- o protocolo *multicast*, por ser baseado em grupos de transmissão, não requer fluxos únicos entre fonte e cliente, tão pouco o compartilhamento de conteúdo entre eles. Neste protocolo multi-destinatário, o fluxo passa apenas uma vez por cada enlace da rede e na topologia especificada há a ocupação de apenas 9 enlaces:

- na transmissão da fonte para o nó 7 são inscritos no grupo *multicast* os roteadores 0 e 3, são ocupados então 3 enlaces: 10-0, 0-3 e 3-7;
- para o nó 9 receber o fluxo da fonte, foi necessário inscrever também o roteador 5 no grupo de transmissão, ocupando então mais 2 enlaces: 3-5 e 5-9;
- para o nó 8 receber o fluxo, foi necessário inscrever os roteadores 1 e 4, ocupando assim mais 3 enlaces: 0-1, 1-4 e 4-8.

De acordo com o relatado anteriormente, o *unicast* sobrecarrega a fonte do tráfego e não otimiza a transmissão na rede de distribuição. No protocolo P2P, por haver compartilhamento de conteúdo entre clientes, a carga é retirada da fonte e repassada sem otimização alguma para a rede de distribuição. Na topologia utilizada, ambos os protocolos sobrecarregam a rede. Em comparação com o protocolo *multicast*, é possível concluir que os protocolos *unicast* e P2P não atendem às necessidades deste trabalho, o qual busca eleger o mecanismos que menos impõe carga sobre a rede de distribuição.

Um possível problema que também pode ocorrer na utilização do protocolo P2P, em decorrência deste compartilhar o conteúdo entre os clientes da rede, é a variação do atraso percebida pelos clientes na reprodução do programa. No simples exemplo de compartilhamento de conteúdo entre clientes P2P citado anteriormente, o cliente 8 possivelmente reproduzirá o programa com atraso em relação ao cliente 7. Essa suposição se apoia no fato do cliente 8 receber o tráfego através do cliente 9, que por sua vez o recebeu através do cliente 7 já com um possível incremento no atraso.

O exemplo exposto deixa claro e melhor desempenho de protocolos P2P se comparado ao protocolo *unicast*. No entanto, também fica claro que o método de compartilhamento de dados utilizando fluxos *unicast* entre *peers* de uma rede P2P não otimiza a distribuição como no *multicast*.

4.5 Considerações finais

Este capítulo apresentou os resultados de simulações realizadas envolvendo os protocolos *unicast* e *multicast* PIM-DM, PIM-SM e PIM-SSM. As simulações realizadas buscaram inicialmente avaliar

a carga que cada um desses protocolos impõem em uma rede de distribuição durante a transmissão de programas como filme, jornal e jogo de futebol.

De acordo com o que foi apresentado no decorrer deste capítulo, o protocolo *unicast*, como esperado, obteve um resultado inferior em relação aos protocolos *multicast* durante a avaliação de carga imposta na rede. As simulações de carga subsequentes utilizaram apenas os protocolos *multicast* PIM-DM, PIM-SM e PIM-SSM. Na avaliação dos resultados de carga realizada para estes três protocolos *multicast*, o PIM-DM obteve o pior resultado seguido do PIM-SM. O protocolo PIM-SSM foi o que obteve o melhor desempenho durante a transmissão dos programas, como visto nas Figuras 4.6, 4.7 e 4.8.

Na avaliação do desempenho do tempo entre mudanças de programa (*zapping*), utilizando os protocolos *multicast* PIM-DM, PIM-SM e PIM-SSM, obtiveram os melhores resultados os protocolos PIM-SM e PIM-SSM. As Figuras 4.14, 4.15 e 4.16, apresentaram os tempos de *zapping* para os protocolos *multicast* PIM-DM, PIM-SM e PIM-SSM, respectivamente. Relacionando estes valores de tempo com as Tabelas 4.3 e 4.4, obteve-se as classificações de regular para PIM-DM e de bom para PIM-SM e PIM-SSM.

Os resultados obtidos nesta dissertação estão em concordância com os trabalhos realizados por Komosny and Novotny (2007) e Cisco Systems Inc. (2007). Ambos utilizam o protocolo *multicast* PIM-SSM para otimizar a distribuição de tráfego IPTV, obtendo expressiva redução de carga se comparado à utilização de outros protocolos. O PIM-SSM também obteve resultados que indicam seu melhor desempenho frente aos demais protocolos utilizados nesta dissertação. Durante as simulações o PIM-SSM obteve ainda tempos reduzidos de *zapping* (máximo de aproximadamente 600 e mínimo de aproximadamente 200 ms) se comparado com o que foi visto em Uzunalioglu (2009) (aproximadamente 830 ms).

Capítulo 5

Conclusões

O objetivo do trabalho consistiu em eleger, através de simulações, quais protocolos dentre os abordados nesta dissertação sejam capazes de fornecer uma base para a implementação de um sistema IPTV numa rede corporativa. Tendo em vista as redes IP atuais e a largura de banda demandada pelo conteúdo de TV em um sistema IPTV, avaliou-se principalmente a carga que cada mecanismo impõe sobre a rede de distribuição. O sistema IPTV proposto deverá então atender a todos os clientes de uma rede corporativa, otimizando a transmissão e evitando sobrecargas.

Foi desenvolvido um ambiente de simulação capaz de se aproximar do cenário real de um sistema IPTV em uma rede corporativa. Para desenvolver este ambiente foram utilizados:

- o simulador de redes de computadores de maior impacto acadêmico, ns-2;
- topologia e perfil de tráfego extraídos de uma rede de distribuição real;
- um módulo, adaptado ao simulador, que permitiu utilizar o protocolo *multicast* PIM-SSM;
- dados retirados de programas reais, gerando assim um tráfego VBR semelhante ao que é transmitido em sistemas IPTV;
- *framework* desenvolvido exclusivamente para esta dissertação utilizando as linguagens *BASH*,

C/C++, Perl e Matlab com a finalidade de executar automaticamente as simulações e analisar os dados (*traces*) extraídos do ns-2;

- os protocolos *unicast*, *multicast* PIM-DM, PIM-SM e PIM-SSM para a simulação de transmissão de conteúdo IPTV.

Entre os protocolos abordados nesta dissertação, o único protocolo que não foi submetido à simulações foi o P2P, devido a limitações no simulador. No entanto, análises teóricas mostraram que o desempenho na redução de carga deste protocolo não é equiparável ao de um protocolo *multicast*. No protocolo P2P há o compartilhamento de um programa entre clientes a fim de todos estes completarem suas transmissões através de múltiplos fluxos *unicast*. Em uma rede P2P, um cliente faz o *download* de partes de um programa existente em outros clientes, enquanto também faz o *upload* das partes que já possui deste programa para outros clientes que não a tenham. Essa dinâmica faz com que no protocolo P2P, um cliente não apenas receba um fluxo da rede de distribuição, mas que retorne para a rede parte daquilo que recebeu, a fim de contribuir com outros clientes. Este comportamento, utilizando múltiplos fluxos *unicast*, piorou o desempenho do P2P em relação a um protocolo *multicast*. No *multicast*, apenas uma cópia de cada programa transita entre os roteadores, atendendo a todos os clientes.

No protocolo P2P ainda há a possibilidade de surgir outro problema, inerente à dinâmica de compartilhamento de conteúdo entre clientes: o atraso variável. Um cliente completa sua transmissão utilizando partes do tráfego que outros clientes já receberam. Isso leva a uma possível a reprodução tardia de um programa em um cliente, em relação aos demais. Para um cliente reproduzir um *streaming* (YouTube por exemplo), é necessário que o tráfego chegue até ele e encha um *buffer*, utilizado para eliminar possíveis problemas durante a recepção dos pacotes. Para que a reprodução de um conteúdo não sofra diretamente com a variação do atraso na chegada de pacotes, ou na chegada fora de ordem destes, o tamanho do *buffer* é aumentado. Quanto maior o *buffer*, mais tempo levará para a rede enchê-lo, o que conseqüentemente refletirá em atrasos para o início da reprodução. A mudança de canais (*zapping*) também pode ser afetada. Neste protocolo é necessário a localização de *peers*, para dar início às transmissões que vão compartilhar um conteúdo entre clientes. Acredita-se que este processo de localização e solicitação de partes de um conteúdo existentes em outros clientes pode

gerar atrasos no tempo de *zapping*.

Outro problema observado no protocolo P2P é que este compartilha entre *peers* um conteúdo que é necessariamente quebrando em fragmentos. Cada *peer* então compartilha os fragmentos que possui deste conteúdo com os demais *peers* na rede virtual. Em um sistema IPTV o conteúdo é reproduzido durante o download (*streaming*). Isso dificulta a utilização de um sistema P2P pois os fragmentos que um *peer* recebe podem não seguir a ordem correta o que implicará em falhas na reprodução do *streaming*.

A perda de desempenho do protocolo P2P em relação à carga imposta na rede de distribuição só não é comparada ao do protocolo *unicast*. Este protocolo inviabiliza totalmente a implementação de um sistema IPTV. Enquanto no P2P existe o compartilhamento de um fluxo entre clientes, o que faz migrar a carga imposta na fonte para a rede de distribuição, no *unicast* a carga incide sobre toda a rede, incluindo a fonte. Neste protocolo, múltiplos fluxos únicos são criados entre fonte e clientes a fim de realizar as transmissões dos programas anunciados no sistema IPTV. Este comportamento do *unicast* faz com que a carga imposta na fonte e rede de distribuição cresça proporcionalmente em relação ao número de clientes no sistema IPTV.

Nas avaliações realizadas sobre a carga imposta na rede de distribuição pelos diferentes protocolos *multicast*, foi possível perceber que entre estes houve variação no nível de carga. De acordo com os resultados obtidos no Capítulo 4, conclui-se que a diferença entre os métodos de difusão, PIM-DM, PIM-SM e PIM-SSM, acontece principalmente em momentos nos quais só existem os anúncios dos programas. Durante uma transferência, como visto no período de 10 a 30 segundos da Figura 4.9, os protocolos *multicast* mantiveram um padrão de transmissão semelhante.

A conclusão quanto ao método de difusão que menos impõe carga na rede de distribuição foi feita então com base nos anúncios de cada protocolo. O protocolo PIM-DM anuncia periodicamente para todos os nós da rede a existência de seus programas. Em cada anúncio, este protocolo envia um trecho atual da transmissão, pois assume que todos os nós da rede estão interessados no tráfego. No protocolo PIM-SM o anúncio não contém parte da transmissão como no PIM-DM. No PIM-SM um roteador é eleito para ser o RP; assim, a fonte transmite continuamente seu programa para o RP eleito,

que por sua vez atende aos receptores do grupo *multicast*.

No PIM-DM, quanto mais esparsos os ambientes, maior será a carga imposta por este protocolo em seus anúncios. A utilização do PIM-DM pode ser interessante em um ambiente onde a maioria dos clientes da rede de distribuição estão interessados no grupo de transmissão anunciado pela fonte. Em um ambiente destes, o número de anúncios emitidos pela fonte se reduz, devido à maioria dos clientes já estarem inscritos no grupo de transmissão. No entanto, num sistema IPTV existe uma variedade de grupos de transmissão proporcional ao número de programas existentes, somado a um ambiente esparsos. O PIM-DM causaria sobrecargas, anunciando todos os canais para a todos os clientes durante todo o tempo de atividade dos grupos.

O impacto de sobrecarga causado pelo PIM-SM é menor se comparado com o PIM-DM. O anúncio deste protocolo é sutil, mas ainda assim neste modelo existe a transmissão contínua do tráfego da fonte para o nó RP. Em um sistema IPTV, todos os programas seriam transmitidos da fonte para um roteador (RP eleito) situado estrategicamente na rede de distribuição. Os enlaces envolvidos certamente serão sobrecarregados durante todo o tempo de atividade do grupo de transmissão, mas este método de difusão tem um desempenho de redução de carga melhor se comparado com o PIM-DM.

O PIM-SM ainda limita a difusão, com a existência de apenas um ponto de transmissão na rede de distribuição. Neste protocolo, é desejável que o RP se situe próximo a “todos” os clientes, pois será a partir deste RP que o tráfego será transmitido para todos eles. Em caso de falhas, outro RP é eleito. No entanto, não existirá sempre um RP mais próximo para cada cliente (ou para uma região). Todos os clientes devem solicitar seu tráfego para o único RP existente na rede de distribuição.

Os métodos de difusão citados anteriormente, PIM-DM e PIM-SM, sobrecarregam a rede se comparados com o PIM-SSM. No PIM-SSM, os anúncios não são acompanhados de trechos da transmissão e também não há a necessidade de um RP na rede de distribuição. No PIM-SSM a transmissão parte da fonte diretamente para o cliente, após este ingressar no grupo de transmissão através de uma mensagem IGMP (*join*). Este protocolo permite ainda a existência de diversas fontes para o mesmo grupo de transmissão. Diferente do que acontece no PIM-SM, onde as mensagens *join* de todos os clientes são enviadas para o único RP na rede de distribuição, no PIM-SSM cada cliente enviará sua

mensagem *join* para a fonte mais próxima que anuncie o programa desejado.

No PIM-SSM existe a possibilidade das fontes serem posicionadas estrategicamente na rede de distribuição, de forma que o tráfego IPTV utilize menos os enlaces necessários para outros fluxos (tráfego para *Internet* por exemplo). Isso faz reduzir o congestionamento em muitos enlaces da rede de distribuição, pois cada tráfego utilizará caminhos distintos, reduzindo o atraso entre clientes, *gateway* e fontes.

De acordo com a metodologia proposta (Capítulo 3) e os resultados obtidos (Capítulo 4), a principal contribuição deste trabalho foi a de eleger o protocolo *multicast* PIM-SSM como o ideal para a transmissão de conteúdo de TV digital, IPTV, em uma rede corporativa. Isto é, o PIM-SSM se sobressaiu sobre os demais protocolos porque impôs menor carga na rede de distribuição durante a transmissão de programas no sistema IPTV simulado.

Em uma simulação que teve o intuito de medir o tempo entre as trocas de canais, o *zapping*, novamente o protocolo *multicast* PIM-SSM foi eleito por sobrecarregar menos a rede de distribuição. Quanto menos um protocolo sobrecarrega a rede, menor é o valor esperado do atraso. De acordo com os resultados obtidos nas simulações de *zapping* (Capítulo 4), este obteve 0.2 segundos, que pelo índice MOS de qualidade subjetiva é classificado como “bom”. Assim, o resultado obtido pelo PIM-SSM nos tempos de *zapping* reafirma este protocolo como o método sugerido por esta dissertação para implementar um sistema IPTV.

Com a utilização do protocolo *multicast* PIM-SSM em uma rede de distribuição, esta deve suportar em seus enlaces pouco mais do que largura de banda necessária para o fluxo de uma cópia de cada programa anunciado pelo serviço IPTV. Com a otimização proposta por este protocolo, apenas uma cópia de cada programa transitará nos enlaces que ligam fonte e clientes, seja qual for o número de clientes. Isso permite que serviços de IPTV sejam implementados em redes atuais, redes estas que devem ser dimensionadas de acordo com o número de canais anunciados. Entretanto, existem limitações quanto ao tipo de conteúdo e aos equipamentos utilizados na rede de distribuição (é necessário que todos os roteadores tenham suporte ao protocolo *multicast* a ser utilizado). O protocolo *multicast*, por ser multidestinatário, consegue distribuir um programa para todos os clientes utilizando apenas

uma cópia deste fluxo na rede de distribuição. No entanto, no modelo *multicast* o cliente é restrito a assistir a programação (“ao vivo”) que a fonte distribui para todos os clientes. No PIM-SSM o cliente não tem a opção de acesso individual à conteúdos de TV, como VoD.

No entanto, as implementações do protocolo PIM-SSM são raras. A empresa *Cisco Systems Inc.* detêm em poucos dos seus melhores roteadores uma implementação deste protocolo. Até onde pode-se pesquisar não há implementações deste protocolo em roteadores distribuído por outros fabricantes. Espera-se que num futuro breve o protocolo PIM-SSM seja suportado em mais roteadores devido à necessidade de protocolos *multicast* impulsionada pelo *IPTV*. O PIM-SSM será privilegiado por ter um desempenho melhor frente aos demais protocolos *multicast*, como apresentado nessa dissertação.

Uma forma de contornar a pouca existência ou a heterogeneidade nas implementações de um protocolo (i.g. PIM-SSM) é a virtualização dos roteadores. O sistema de um roteador pode ser virtualizado em diversas instâncias e acessado remotamente pelos ativos espalhados na rede de distribuição. Esta é uma saída para resolver problemas como a incompatibilidade entre implementações de um protocolo realizadas por diversas empresas. Neste caso, um roteador é adquirido e seu sistema operacional é virtualizado com o número de instâncias necessárias para atender aos ativos existentes na rede de distribuição. Este tipo de solução também traz economias na implementação de uma rede pelo uso reduzido de roteadores físicos. Devido a escassez de roteadores com uma implementação do protocolo PIM-SSM, os poucos existentes são extremamente caros. Utilizando virtualização o número de roteadores necessários reduz drasticamente.

5.1 **Trabalhos Futuros**

Protocolos como o P2P deixaram de fazer parte das simulações devido à limitações do simulador ns-2. A inclusão deste protocolo em simulações utilizando os diversos modelos de compartilhamento entre clientes, sem dúvida é uma contribuição importante. A comparação através de simulação ou implementação entre os protocolos *multicast* e P2P é um trabalho a ser desenvolvido.

Outros protocolos *multicast*, como exemplo do PIM-MSDP (*Multicast Source Discovery Proto-*

col), ainda não suportado no ns-2, uma variante do PIM-SM que possibilita a existência de múltiplos RPs na mesma árvore de distribuição (Kim et al., 2003), também são importantes de serem simulados e comparados aos resultados obtidos nesta dissertação.

O PIM-SSM se limita a tráfegos contínuos, como programas em uma TV convencional. Caso haja a necessidade de transmissão de conteúdo VoD em um sistema IPTV implementado com PIM-SSM, este protocolo não será capaz de atender às necessidades individuais de cada cliente. Se faz necessário então um modelo de serviço que otimize os dois tipos de tráfego, uma tecnologia híbrida, utilizando *multicast* PIM-SSM para programas contínuos e outra tecnologia (possivelmente P2P) para conteúdos VoD.

Referências Bibliográficas

- Adams, A., Nicholas, J., and Siadak, W. (2003). Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised). *RFC 3973*.
- Agarwal, S. (2007). A case study of large scale P2P video multicast. *IP Multimedia Subsystem Architecture and Applications, 2007 International Conference*, pages 1–5.
- Begen, A. C., Glazebrook, N., and Ver Steeg, W. (2009). A unified approach for repairing packet loss and accelerating channel changes in multicast iptv. *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, pages 1–6.
- Cain, B., Deering, S., Kouvelas, I., Fenner, B., and Thyagarajan, A. (2002). Internet Group Management Protocol, version 3. *RFC 3376*.
- Caja, J. (2006). Optimization of IPTV multicast traffic transport over next generation metro networks. *Telecommunications Network Strategy and Planning Symposium, 2006. NETWORKS 2006. 12th International*, pages 1–6.
- Camilo, T., Silva, J. S., and Boavida, F. (2004). Enabling ns-2 with SSM environments. *7ª Conferência Sobre Redes de Computadores CRC2004*.
- Cardozo, A. Q. (2002). Mecanismos para garantir qualidade de serviço de aplicações de video sob demanda. Master's thesis, UFRJ.
- Cha, M., Rodriguez, P., Moon, S., and Crowcroft, J. (2008). On next-generation telco-managed P2P TV architectures. In *ACM Internet Measurement Conference*.

- Cisco Systems Inc. (2007). IPTV Solutions for Wireline Carriers - IP Multicast Technical Overview. http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6552/prod_white_paper0900aecd804d5fe6_ns610_Networking_Solutions_White_Paper.html acessado em 03/2009.
- Costa, L. H. M. K. and Duarte, O. C. M. B. (2003). Roteamento Multicast na Internet. *Mini-Curso do XXI Simpósio Brasileiro de Redes de Computadores - SBRC'2003*, pp. 228-275, maio de 2003.
- Deering, S., Fenner, B., and Haberman, B. (1999). Multicast Listener Discovery (MLD) for IPv6. *RFC 2710*.
- Diot, C., Dabbous, W., and Crowcroft, J. (Apr 1997). Multipoint communication: a survey of protocols, functions, and mechanisms. *Selected Areas in Communications, IEEE Journal on*, 15(3):277–290.
- Estrin, D., Farinacci, D., Helmy, A., Thaler, D., Deering, S., Handley, M., Jacobson, V., Sharma, P., and Wei, L. (1998). Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification. *RFC 2362*.
- Fan, Z. (2007). Bandwidth allocation for MPEG-4 traffic in IEEE 802.11e wireless networks based on traffic prediction. *Future generation communication and networking (fgcn 2007)*, 1:191–196.
- Fitzek, F. and Reisslein, M. (2001). MPEG-4 and H.263 video traces for network performance evaluation. *Network, IEEE*, 15(6):40–54.
- Gallo, D., Miers, C., Coroama, V., Carvalho, T., Souza, V., and Karlsson, P. (2009). A multimedia delivery architecture for IPTV with P2P-Based time-shift support. *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, pages 1–2.
- Gill, P., Arlitt, M., Li, Z., and Mahanti, A. (2007). YouTube traffic characterization: a view from the edge. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 15–28, New York, NY, USA. ACM.
- Gnutella (2004). *The Gnutella Protocol Specification, v0.4*.
- Gonçalves, M. and Niles, K. (1998). IP Multicasting Concepts and Applications. *McGraw-Hill*.

- Hei, X., Liu, Y., and Ross, K. W. (2008). IPTV over P2P streaming networks: The mesh-pull approach. *IEEE COMMUNICATIONS MAGAZINE*, 46(2):86–92.
- Holbrook, H. and Cheriton, D. (1999). IP Multicast Channels: EXPRESS Support for Large-scale Single-source Applications. *Proc. ACM SIGCOMM'99*.
- ISI (2009). *Network Simulator*, <http://www.isi.edu/nsnam/ns/> (Acessado em 02/2009).
- ITU-T (1991). ITU-T International Telecommunication Union - Telecommunication Standardization Sector. *Recommendation I.324: ISDN Network Architecture*.
- ITU-T (2006). International Telecommunication Union - Telecommunication Standardization Sector. *G.1030 Estimating end-to-end performance in IP networks for data applications*.
- Jennehag, U. and Pettersson, S. (2008). On synchronization frames for channel switching in a gop-based IPTV environment. *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pages 638–642.
- KaZaA (2008). *Official web site* (<http://www.kazaa.com>).
- Kim, D., Meyer, D., Kilmer, H., and Farinacci, D. (2003). Anycast Rendezvous Point (RP) mechanism using Protocol. *RFC 3446*.
- Kim, Y., Park, J. K., Choi, H. J., Lee, S., Park, H., Kim, J., Lee, Z., and Ko, K. (2008). Reducing IPTV channel zapping time based on viewer's surfing behavior and preference. *Broadband Multimedia Systems and Broadcasting, 2008 IEEE International Symposium on*, pages 1–6.
- Komosny, D. and Novotny, V. (2007). Tree structure for source-specific multicast with feedback aggregation. *ICN07 - The Sixth International Conference on Networking, Martinique*.
- Kooij, R., Ahmed, K., and Brunnström, K. (2006). Perceived quality of channel zapping. *Proceedings of the Fifth IASTED International Conference Communication System and Networks*, pages 155–158.

- Kukhmay, Y., Glasman, K., Peregudov, A., and Logunov, A. (2006). Video over IP networks: Subjective assessment of packet loss. *Consumer Electronics, 2006. ISCE '06. 2006 IEEE Tenth International Symposium on*, pages 1–6.
- Kurose, J. F. and Ross, K. W. (2006). *Redes de computadores e a Internet: uma abordagem top-down*. Pearson Addison Wesley, São Paulo, 3 edition.
- Lanfranchi, L. and Bing, B. (2008). Short-term MPEG-4 AVC bandwidth prediction for broadband cable networks. *Communication Networks and Services Research, Annual Conference on*, 0:25–29.
- Lee, C.-S. (2007). IPTV over next generation networks in ITU-T. *Broadband Convergence Networks, 2007. BcN '07. 2nd IEEE/IFIP International Workshop on*, pages 1–18.
- Luo, X., Jin, Y., Zeng, Q., Sun, W., Guo, W., and Hu, W. (2008). On the stability of multicast flow aggregation in IP over optical network for IPTV delivery. *CHINESE OPTICS LETTERS*, 6(8):553–557.
- Mushtaq, M. and Ahmed, T. (2009). IPTV service delivery over SP-Driven P2P networks - a framework for linear to non-linear service distribution. *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, pages 1–5.
- Ouali, A., Jaumard, B., and Hebuterne, G. (2008). Trade-offs in peer delay minimization for video streaming in P2P systems. *Cluster Computing and the Grid, 2008. CCGRID '08. 8th IEEE International Symposium on*, pages 615–620.
- Pereira, V., Monteiro, E., and Mendes, P. (2007). Evaluation of an overlay for source-specific multicast in asymmetric routing environments. *Global Telecommunications Conference, 2007. GLOBE-COM '07. IEEE*, pages 1987–1992.
- Peterson, L. and Davie, B. (2004). *Redes de computadores: uma abordagem de sistemas. 3 Ed. Rio de Janeiro: Campus, 2004*
- Rocha, J., Domingues, M., Callado, A., Souto, E., Silvestre, G., Kamienski, C., and Sadok, D. (2004). Peer-to-peer: Computação colaborativa na Internet, capítulo 1, pp. 3–46. *Minicurso SBRC 2004*.

- Sasaki, C., Tagami, A., Hasegawa, T., and Ano, S. (2008). Rapid channel zapping for IPTV broadcasting with additional multicast stream. *Communications, 2008. ICC '08. IEEE International Conference on*, pages 1760–1766.
- She, J. and Ho, P.-H. (2008). Cooperative coded video multicast for IPTV services under EPON-WiMAX integration. *IEEE COMMUNICATIONS MAGAZINE*, 46(8):104–110.
- She, J., Yu, X., Ho, P.-H., and Yang, E.-H. (2009). A Cross-Layer Design Framework for Robust IPTV Services Over IEEE 802.16 Networks. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, 27(2):235–245.
- Shihab, E. and Cai, L. (2007). IPTV Distribution Technologies in Broadband Home Networks. *Electrical and Computer Engineering, 2007. CCECE 2007. Canadian Conference on*, vol., no., pp.765-768, 22-26.
- Silva, J. S., Camilo, T., Pinto, P., Ruivo, R., Rodrigues, A., Gaudêncio, F., and Boavida, F. (2008). Multicast and IP multicast support in wireless sensor networks. *JNW*, 3(3):19–26.
- Silverston, T. and Fourmaux, O. (2006). P2P IPTV measurement: A comparison study. *CoRR*.
- Smith, D. (2007). IPTV bandwidth demand: Multicast and channel surfing. *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 2546–2550.
- Tanenbaum, A. S. (1997). *Redes de computadores*. Campus, Rio de Janeiro, 4 th. edition.
- Teh, A., Pudney, P., and Jayasuriya, A. (2007). Efficient admission control based on predicted traffic characteristics. *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, pages 1–5.
- Tian, Y., Wu, D., Sun, G., and Ng, K.-W. (2008). Improving stability for peer-to-peer multicast overlays by active measurements. *JOURNAL OF SYSTEMS ARCHITECTURE*, 54(1-2):305–323.
- Uzunalioglu, H. (2009). Channel change delay in IPTV systems. *Consumer Communications and Networking Conference, 2009. CCNC 2009. 6th IEEE*, pages 1–6.

- Venkataraman, V., Yoshida, K., and Francis, P. (2006). Chunkyspread: Heterogeneous unstructured tree-based peer-to-peer multicast. *Network Protocols, 2006. ICNP '06. Proceedings of the 2006 14th IEEE International Conference on*, pages 2–11.
- Wu, C., Li, B., and Zhao, S. (2008). Multi-channel live P2P streaming: Refocusing on servers. *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 1355–1363.
- Yarali, A. and Cherry, A. (2005). Internet Protocol Television (IPTV). *Proceedings of the TENCON 2005 - IEEE International Region 10 Conference.*, pages pp.1–6.
- Zeng, Y. and Strauss, T. (2008). Enhanced Video Streaming Network with Hybrid P2P Technology. *BELL LABS TECHNICAL JOURNAL*, 13(3):45–58.