



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Uma metodologia de otimização de roteamento com OSPF para tráfego de vídeo autossimilar

Henrique Domingues Garcia

Dissertação apresentada como requisito parcial
para conclusão do Mestrado em Informática

Orientadora

Prof.^a Dr.^a Priscila América Solis Mendez Barreto

Brasília
2015

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Mestrado em Informática

Coordenadora: Prof.^a Dr.^a Alba Cristina Magalhães Alves de Melo

Banca examinadora composta por:

Prof.^a Dr.^a Priscila América Solis Mendez Barreto (Orientadora) — CIC/UnB

Prof. Dr. Paulo H. P. de Carvalho — ENE/UnB

Prof. Dr. Mylène Christine Queiroz de Farias — ENE/UnB

CIP — Catalogação Internacional na Publicação

Garcia, Henrique Domingues.

Uma metodologia de otimização de roteamento com OSPF para tráfego de vídeo autossimilar / Henrique Domingues Garcia. Brasília : UnB, 2015.

93 p. : il. ; 29,5 cm.

Dissertação (Mestrado) — Universidade de Brasília, Brasília, 2015.

1. FTV, 2. Vídeo, 3. Roteamento, 4. autossimilaridade, 5. Engenharia de tráfego, 6. Otimização

CDU 004.7

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília-DF — Brasil

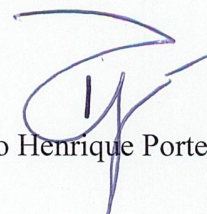
HENRIQUE DOMINGUES GARCIA

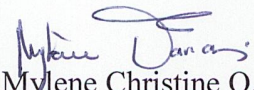
Uma metodologia de otimização de roteamento com OSPF para tráfego de vídeo autossimilar

Dissertação aprovada como requisito parcial para obtenção do grau de Mestre no Curso de Pós-graduação em Informática da Universidade de Brasília, pela Comissão formada pelos professores:

Orientador:


Prof. Dr. Priscila America Solis M. Barreto
(CIC/UnB)


Prof. Dr. Paulo Henrique Portela de Carvalho
(ENE/UnB)


Prof. Dr. Mylene Christine Q. de Farias
(ENE/UnB)

Vista e permitida a impressão.
Brasília, 04 de setembro de 2015.

Prof.^a Dr.^a Alba Cristina Magalhães Alves de Melo
Programa de Pós-Graduação em Informática
Departamento de Ciência da Computação
Universidade de Brasília

Dedicatória

Dedico esse trabalho a todos que me ajudaram em Brasília e que de alguma forma contribuíram para que eu suportasse morar nesse quadradinho, especialmente minha mãe pela fé e minha irmã Elisa pelo apoio.

Agradecimentos

Gostaria de agradecer a todos meus amigos de Goiânia que me apoiaram nesta empreitada senhor Rildo, Rubens, Tedy, Etheila, Laura, Naiara, Elaine, Mitcheã, Andreia, Thaís, Fernanda, Lidiane, tia Cristina, o tio Luigi, Isaac e Isabel, Job, Carol, meu orientador de graduação João Batista e os de Brasília, Luana obrigado pela paciência, Thiago "lagoa", meus companheiros de orientação Kerlla, Paula, Marcelo e Luiz. além de Nilson, Gustavo, Jeremias, Amanda, Daniel, Ariane, Ana, Leia, Wosley, Saad, Jonathan, Vera, Lucimeire e Juliana, Juan, Gabriel, Waldemir, Tina, Eduardo, Heraldo, Cláudio e Fernanda. Também não posso deixar de agradecer os professores André Drummond, Marcos Caetano, Flávio Vidal e minha orientadora que me ensinaram muito aqui. Provavelmente há mais nomes que me fogem a memória e que de uma forma ou de outra contribuíram com o desenvolvimento deste trabalho. Agradeço a todos.

Resumo

Este trabalho apresenta uma metodologia de engenharia de tráfego que tem como objetivo otimizar o custo do enlace de protocolos tradicionais de roteamento tais como o OSPF (*Open Shortest Path First*), a partir da otimização de métricas baseadas na utilização e na característica do tráfego do enlace. Foi utilizado um modelo de programação linear extraído da curva do atraso máximo do enlace calculado com o FEP (*Fractal Envelope Process*). Os resultados mostram que para o tráfego de vídeo que possui um alto grau de autossimilaridade, é necessário preservar a utilização média do enlace abaixo de 25% para obter valores de atrasos compatíveis com os requerimentos de QoS (Quality of Service) da aplicação.

Palavras-chave: FTV, Vídeo, Roteamento, autossimilaridade, Engenharia de tráfego, Otimização

Abstract

This paper presents a traffic engineering methodology that aims to optimize the link cost of traditional routing protocols such as OSPF (Open Shortest Path First) using link metrics based on the link load and traffic characteristics. We propose a linear programming model based on the maximum delay curve of the link calculated by the FEP (*Fractal Envelope Process*). The results show that for video traffic that has a high degree of self-similarity, the average link utilization must be below 25% to achieve delay values compatible with the QoS requirements (Quality of Service) of the application.

Keywords: FTV, Video, Routing, Self-Similarity, Traffic Engineering, Optimization

Sumário

1	Introdução	1
2	Roteamento e Engenharia de Tráfego	4
2.1	Introdução	4
2.2	Modelando a rede com grafos	7
2.3	Protocolos de roteamento	10
2.3.1	<i>Broadcast e flooding</i>	10
2.3.2	OSPF - Open Shortest Path First	11
2.4	Engenharia de tráfego e otimização	13
2.4.1	O problema de otimização de roteamento	16
2.5	Resumo do capítulo	18
3	Caracterização do Tráfego de Vídeo Digital	19
3.1	O Vídeo digital	19
3.1.1	Compressão de vídeo	21
3.2	Transmissão de vídeo na Internet	21
3.2.1	Distribuição de vídeo	23
3.2.2	Arquiteturas de transmissão de vídeo	24
3.3	Qualidade de serviço para vídeo na Internet	27
3.4	FTV	31
3.5	Tipos de tráfego	33
3.5.1	O Tráfego de Poisson	36
3.5.2	O Tráfego Autossimilar	36
3.6	Caracterização do tráfego de vídeo	40
3.7	Caracterização do tráfego FTV	44
3.8	Resumo do capítulo	46
4	Metodologia de Otimização de Roteamento e Resultados Experimentais	47
4.1	Proposta	47
4.1.1	Objetivos gerais	47

4.1.2	Objetivos específicos	47
4.2	Modelo de engenharia de trafego	48
4.3	Cenário de simulação	53
4.3.1	Topologia da rede	53
4.3.2	Métricas e parâmetros	54
4.3.3	Descrição da composição do tráfego nos cenários de simulação . . .	56
4.4	Resultados e discussão	60
4.4.1	Cenário 1	60
4.4.2	Cenário 2	64
4.4.3	Análise geral dos resultados	68
5	Conclusões e Trabalhos Futuros	72
	Referências	74

Lista de Figuras

2.1	Interações das camadas do modelo TCP/IP no roteamento	5
2.2	Dois tipos de comutação: a) Comutação por circuitos virtuais, b) Comutação por pacotes.	6
2.3	Estrutura da interconexão de redes.	7
2.4	(a) Um exemplo de rede de comunicações. (b) A representação na forma de grafo de (a).	8
2.5	Topologias em grafos	9
2.6	Algoritmo de Dijkstra	9
2.7	Exemplo do funcionamento do Dijkstra	10
2.8	Exemplo de <i>flooding</i> utilizando números de sequência	11
2.9	Um AS dividido em áreas com seus vários tipos de roteadores.	13
2.10	Exemplo de roteamento com peso igual ao inverso da capacidade	14
2.11	Exemplo de roteamento com todos os pesos igual a 1	15
2.12	Framework de engenharia de tráfego com OSPF [18]	16
3.1	Interpolação de quadros [35]	23
3.2	Exemplo de distribuição de vídeo ao vivo	24
3.3	Aplicação Cliente Servidor	25
3.4	Aplicação P2P ao vivo	26
3.5	Aplicação FTV	31
3.6	Arquitetura do FTV [55]	32
3.7	Exemplo de dispositivos usados com FTV [59], [60], [22]. [43].	33
3.8	Tráfego VBR	34
3.9	Tráfego em Rajadas	35
3.10	Exemplo de série temporal com LRD e parâmetro $H = 0,8$	38
3.11	Invariância a escala do filme Tropas estelares [44]	41
3.12	Atraso máximo da fila em função da utilização para Processo envelope	43
3.13	Crescimento do atraso até 99% de utilização	43

4.1	Crescimento do custo em função da utilização da fila M/M/1 para o modelo de Fortz	48
4.2	Comparação do tamanho da fila para vários modelos.	50
4.3	Crescimento do custos do enlace em função da utilização para o FEP	51
4.4	Comparação do crescimento do custo do enlace para o LPM-FEP com vários valores de H e o LPM-Fortz (azul pontilhado).	52
4.5	Mapa da Rede Nacional de Pesquisa.	53
4.6	Topologia da Rede Ipê utilizada para simulação	54
4.7	Fluxograma utilizado para a simulação	57
4.8	Arquitetura FTV utilizada neste trabalho proposta em [49]	59
4.9	Atraso médio da rede com 5% de tráfego de fundo	60
4.10	Utilização média da rede com 5% de tráfego de fundo	61
4.11	Atraso médio da rede com 10% de tráfego de fundo	62
4.12	Utilização média da rede com 10% de tráfego de fundo	62
4.13	Atraso médio da rede com 15% de tráfego de fundo	63
4.14	Utilização média da rede com 15% de tráfego de fundo	63
4.15	Atraso médio da rede com 20% de tráfego de fundo	64
4.16	Utilização média da rede com 20% de tráfego de fundo	65
4.17	Atraso médio da rede com 5% de tráfego de fundo	66
4.18	Utilização média da rede com 5% de tráfego de fundo	66
4.19	Atraso médio da rede com 10% de tráfego de fundo	67
4.20	Utilização média da rede com 10% de tráfego de fundo	67
4.21	Atraso médio da rede com 15% de tráfego de fundo	68
4.22	Utilização média da rede com 15% de tráfego de fundo	69
4.23	Atraso médio da rede com 20% de tráfego de fundo	69
4.24	Utilização média da rede com 20% de tráfego de fundo	70

Lista de Tabelas

3.1	Padrões de TV digital [35]	20
3.2	Comparação dos requisitos de QoS de diversas aplicações [54]	28
3.3	Parâmetro H para diferentes quantizações e quantidades diferente de amostras [44]	42
3.4	Características do vídeo	45
4.1	Características dos Enlaces usados na simulação	55

Lista de Siglas

3D-HEVC - 3D	<i>High Efficiency Video Coding</i>
3DV	<i>3D Video</i>
AF	<i>Assured Forwarding</i>
AS	<i>Autonomus System</i>
ATM	<i>Asynchronous Transfer Mode</i>
ATSC	<i>Advanced Television System Committee</i>
BGP	<i>Border Gateway Protocol</i>
CBR	<i>Constant Bit Rate</i>
CRT	<i>Cathode Ray Tube</i>
CV	<i>Circuitos Virtuais</i>
DCT	<i>Discrete Cosine Transform</i>
DE	<i>Default</i>
DiffServ	<i>Differentiated Service</i>
DSCP	<i>Diferentiated Services Code Point</i>
ECMP	<i>Equal Cost MultiPath</i>
EDTV	<i>Enhanced Definition TV</i>
EF	<i>Expedited Forwarding</i>
fBm	<i>Movimento Browniano Fracionário</i>
fGn	<i>Ruído Gaussiano Fracionário</i>
FIFO	<i>First-in-First-Out</i>

FPS	<i>Frames per Seconds</i>
FTP	<i>File Transfer Protocol</i>
FTV	<i>Free Viewpoint Television</i>
HDTV	<i>High Definition TV</i>
IETF	<i>Internet Engeneering Task Force</i>
IGP	<i>Interior Gateway Protocol</i>
IntServ	<i>Integrated Service</i>
IP	<i>Internet Protocol</i>
IPv4	<i>Internet Protocol version 4</i>
IPv6	<i>Internet Protocol version 6</i>
IS-IS	<i>Intermediate System-Intermediate System</i>
ISO	<i>International Standards Organization</i>
ISP	<i>Internet Service Provider</i>
KLT	<i>Kerhunen-Loève Transform</i>
LRD	<i>Long Range Dependence</i>
MPEG	<i>Moving Picture Experts Group</i>
MPLS	<i>MultiProtocol Label Switching</i>
MSE	<i>Mean Square Error</i>
MST	<i>Minimum Spanning Tree</i>
MV-HEVC	<i>Multiview Video – High Efficiency Video Coding</i>
MVC	<i>Multiview Video Code</i>
NTSC	<i>National Television System Committee</i>
OSI	<i>Open System Interconnection</i>
OSPF	<i>Open Shortest Path First</i>
P2P	<i>Peer-to-Peer</i>
PAL	<i>Phase Alternating Line</i>
PHB	<i>Per-Hop Behavior</i>

PSNR	<i>Peak-Sinal-To-Noise Ratio</i>
QoE	<i>Quality of Experience</i>
QoS	<i>Quality of Service</i>
RGB	<i>Red-Green-Blue</i>
RIP	<i>Routing Information Protocol</i>
RSVP	<i>Resource reSerVation Protocol</i>
RTP	<i>Real Time Protocol</i>
SDTV	<i>Standard Definition TV</i>
SP	<i>Shortest Path</i>
TCP	<i>Transport Control Protocol</i>
TCP/IP	<i>Transport Control Protocol/Internet Protocol</i>
TOTEM	<i>TOolbox for Traffic Engineering Methods</i>
UHDTV	<i>Ultra High Definition TV</i>
USP	<i>Unique Shortest Path</i>
VBR	<i>Variable Bit Rate</i>
VNI	<i>Visual Networking Internet</i>
VTP	<i>Variance-Time-Plot</i>

Capítulo 1

Introdução

Até meados da década de 90 acreditava-se que o tráfego multimídia se comportava como o tráfego telefônico e as redes de pacotes eram projetadas com base no modelo de Poisson e teoria de filas. Porém, a Internet é uma rede de pacotes de tamanho variável e a taxa de processamento dos pacotes é variável. Este comportamento acaba gerando micro rajadas que em grandes escalas criam um efeito chamado de autossimilaridade [47]. Essa autossimilaridade é responsável pelo tráfego apresentar picos de atividade com maior duração do que o previsto pelo modelo de Poisson, podendo gerar perda de pacotes e longos atrasos.

Uma das soluções mais simples para garantir a qualidade da transmissão em uma rede é oferecer recursos excedentes, tais como largura de banda e memória, de tal forma que o tráfego não encontrará resistência para ser transmitido. Porém esta técnica gera desperdício de recursos além de ser muito cara e altamente empírica. O dimensionamento eficiente dos enlaces parte da correta caracterização do tráfego de rede dentro de uma metodologia de engenharia de tráfego. Entretanto, o tráfego autossimilar, que é maioria nas redes de comutação de pacotes, apresenta uma complexidade matemática que somente permite especificar valores máximos para atrasos e tamanho da fila de uma fração do tráfego [16] e [39], mas ainda não existe um consenso científico sobre um modelo exato para o dimensionamento dos enlaces.

A Internet nos últimos anos tem integrado diversas aplicações em que o vídeo é uma das que apresentam maior crescimento. De acordo com o relatório Interconexão Visual da Internet (VNI - *Visual Networking Internet*) da Cisco [58], estima-se que até 2018 o tráfego na Internet ultrapassará 1 zettabyte por ano, dos quais aproximadamente 90% serão de vídeo.

Vários estudos e pesquisas confirmam que o tráfego de vídeo é autossimilar [20] [14] [47] [36], o que representa um desafio considerável em termos de alcançar métricas de QoS em uma rede como a Internet projetada para o serviço do melhor esforço.

Atualmente as pesquisas para adequar a pilha Protocolo de Controle de Transporte/-Protocolo de Internet (TCP/IP - *Transport Control Protocol/Internet Protocol*) às futuras necessidades de aplicações que precisam de QoS se baseiam em dois paradigmas para a camada de rede e transporte: adequar os protocolos atuais através de pesquisas evolucionárias que tentam adaptar os protocolos que já existentes da forma mais escalável possível ou projetar uma nova arquitetura de Internet a partir do zero, ou *clean slate* [45].

Vários trabalhos anteriores [24] têm abordado o roteamento como uma forma de alcançar níveis de Qualidade de Serviço (QoS - *Quality of Service*) previsíveis na rede, sem o engessamento dos modelos tradicionais e altamente limitados, como o as arquiteturas *IntServ* e *DiffServ*. Já foi determinado que a arquitetura *IntServ* não possui escalabilidade suficiente para atender milhares de clientes e a arquitetura *DiffServ* não consegue oferecer qualidade de forma eficiente, além de que exige mudanças complexas nos nós da rede e fortes limitações entre domínios diferentes [54].

A utilização de protocolos tradicionais, tal como o protocolo Aberto de Primeiro Menor Caminho (OSPF - *Open Shortest Path First*) que é nativo da camada de rede do TCP/IP, permitem que a partir de uma representações simples e concisa, sejam alterados os custos dos enlaces com base no tipo de tráfego e o dinamismo da carga no enlace. Esta é uma alternativa interessante por possuir maior escalabilidade e compatibilidade com futuras evoluções da pilha TCP/IP, como a Versão 6 do Protocolo de Internet (IPv6 - *Internet Protocol version 6*), além de possuir estabilidade e baixa complexidade por não necessitar nenhuma alteração radical no modelo de camadas e facilidade na administração com redução da intervenção humana para gerenciamento da rede [18].

Diversos autores já trabalharam a otimização do tráfego considerando protocolos tradicionais de roteamento [18], [6] [32] [2] [10], porém, nenhum considerava a autossimilaridade como parâmetro para alteração do peso do enlace. Altin [2] descreve as principais pesquisas de engenharia de tráfego com protocolos de roteamento de menor caminho dos últimos 13 anos e percebe-se que a otimização de roteamento geralmente é um problema do tipo NP-completo e extremamente custosa. Neste mesmo cenário, uma grande gama de pesquisadores desenvolveram heurísticas para otimizar a busca de um valor ótimo em tempo assintoticamente polinomial. Apesar de alguns conseguirem, estas heurísticas apresentam apenas um valor sub ótimo.

Fortz [18] apresenta uma metodologia que foi utilizada como base neste trabalho. Ele desenvolve um conjunto de equações lineares que relaciona o crescimento do custo com o crescimento das filas do modelo M/M/1. A partir dessa função de custo é elaborado um modelo de programação linear baseado no problema geral de roteamento pelo menor caminho. Esta aproximação permite extrair os custos de forma que obedecem as restrições do modelo e reflita os parâmetros de qualidade exigidos em tempo polinomial.

Posteriormente, Leduc et al. [32] realizaram a implementação do modelo de Fortz em uma plataforma aberta com diversas heurísticas e algoritmos tanto para Multi-Protocolo de Roteamento de Rótulo (MPLS - *MultiProtocol Label Switching*), quanto para protocolos de menor caminho. Este simulador, chamado de Caixa de Ferramenta para Métodos de Engenharia de Tráfego (TOTEM - *TOolbox for Traffic Engineering Methods*) possui código aberto e está disponível no site do projeto em [32].

Balon em [6] realiza a comparação das funções objetivos de nove modelos de engenharia de tráfego, entre eles o modelo de Fortz e a recomendação da CISCO que sugere a métrica do inverso da capacidade do enlace, entre outros. Alguns modelos como o da CISCO, que considera o inverso da capacidade, não conseguem distribuir o tráfego corretamente gerando enlaces subutilizados enquanto concentram o tráfego em outros enlaces, gerando congestionamento. Neste trabalho o modelo de Fortz apresenta um dos melhores resultados, gerenciando o tráfego de forma muito próxima a ótima.

Com base nas pesquisas anteriores, o principal objetivo deste trabalho é definir uma metodologia de otimização de roteamento para melhora do desempenho de aplicações de vídeo, com base na adaptação dos protocolos tradicionais de melhor caminho para uma maior escalabilidade e generalização de aplicações na rede. Para isto foi realizado o estudo de diversos fluxos de vídeo de alta resolução e proposto um problema de otimização linear que permita a otimização dos custos dos enlaces em protocolos tradicionais de roteamento ao considerar o índice de autossimilaridade e utilização média do enlace para descoberta de rotas. O modelo definido será validado em um ambiente de simulação baseado no TOTEM, uma vez que foi utilizado em diversos trabalhos relacionados [32] [2].

Este trabalho está estruturado da forma seguinte. O capítulo 2 descreve o roteamento, os principais protocolos e as técnicas de engenharia de tráfego utilizadas. O capítulo 3 se concentra no tráfego de vídeo e suas características. Ele descreve as aplicações consideradas, suas necessidades e as técnicas utilizadas para estimação da autossimilaridade. O capítulo 4 apresenta a proposta deste trabalho assim como os resultados experimentais e a sua análise. Finalmente, no capítulo 5 são apresentadas as conclusões e trabalhos futuros desta pesquisa.

Capítulo 2

Roteamento e Engenharia de Tráfego

Este capítulo mostra os principais conceitos de roteamento e sua relação com a entrega de pacotes. Após a introdução de conceitos gerais, para explicar as abstrações utilizadas pelos protocolos, é feita uma breve apresentação da teoria de grafos na seção 2.2. Em seguida é feita a descrição do protocolo de roteamento utilizado neste trabalho, o OSPF, na seção 2.3. Por fim, a seção 2.4 é discutido o conceito de engenharia de tráfego e o problema de descoberta de rotas dentro de um domínio.

2.1 Introdução

O roteamento é a tarefa de guiar a informação através de uma rede de comunicação de sua origem a seu destino [8]. A camada de rede do modelo TCP/IP é responsável pela descoberta de rotas fim-a-fim através de um protocolo de roteamento [53]. Os roteadores da rede utilizam os protocolos de roteamento para descobrir os melhores caminhos. Posteriormente, a interface adequada para a rota de cada destino é armazenada em uma estrutura chamada tabela de roteamento.

Para realizar a comutação de dois pontos, ao receber o pacote, os roteadores removem os cabeçalhos da interface de rede, lê o endereço do Protocolo de Internet (IP - *Internet Protocol*) de destino do cabeçalho IP, procura em sua tabela de roteamento a interface que possivelmente está conectada àquele destino, deduz seu TTL (*Time To Live*), recalcula a soma de verificação e então o encaminha para a interface de rede associada, como mostra a Figura 2.1.

Existem dois tipos de comutação digital: a comutação por Circuitos Virtuais (CV), mostrada na Figura 2.2a onde o algoritmo de roteamento é utilizado para a escolha da rota do circuito virtual. Dessa forma a decisão de roteamento é realizada apenas durante a conexão do CV e todos os pacotes pertencentes a esta conexão seguem o mesmo caminho até o destino [8]. E a comutação de pacotes onde a decisão de roteamento é realizada

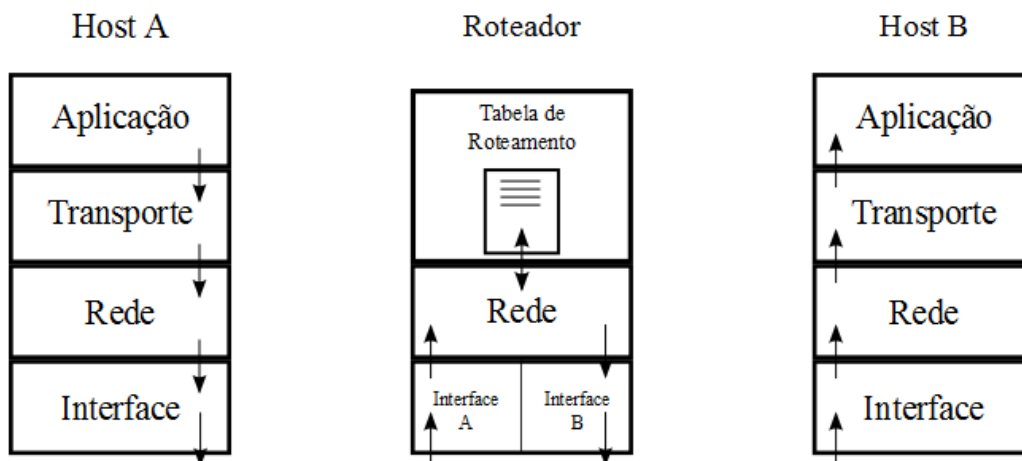


Figura 2.1: Interações das camadas do modelo TCP/IP no roteamento

individualmente para cada pacote em cada nó. Assim dois pacotes do mesmo par origem-destino (s, t) podem seguir caminhos diferentes, como mostra a Figura 2.2b.

A escolha do tipo de comutação depende do administrador da rede. Um conjunto de comutadores, também chamados de roteadores, sob a mesma administração técnica é chamada de Sistema Autônomo (*AS - Autonomus System*). A Internet é uma rede de milhares de AS interconectados entre si. Cada AS possui seu próprio sistema de gerenciamento, suas próprias políticas e seu próprio protocolo de roteamento. Contudo, o principal problema do roteamento é conseguir encontrar o melhor caminho para um conjunto de pacotes de sua origem ao seu destino baseado apenas em um pequeno conjunto de informações.

Os protocolos de roteamento utilizados dentro de um AS são chamados de Protocolos de Roteamento Intradomínio (IGP - *Interior Gateway Protocol*), como por exemplo o OSPF, o Protocolo de Roteamento de Informação (RIP - *Routing Information Protocol*) e IS-IS (Intermediate System - Intermediate System).

Os protocolos IGP podem ser classificados de acordo com a forma de encaminhamento: quando o alcance é apenas um destino o algoritmo é chamado *Unicast*. Se o destino for todos os nós da rede é chamado *Broadcast*, caso seja um grupo de nós é chamado *Multicast* e caso o destino seja qualquer nó de um grupo é chamado de *Anycast*. Por outro lado, os protocolos utilizados para interligar os AS são chamados Protocolos de Roteamento Interdomínio (EGP - *Exterior Gateway Protocol*) [17] [54]. A Internet possui apenas um único protocolo EGP chamado Protocolo de *Gateway* de Borda (BGP - *Border Gateway Protocol*). A Figura 2.3 exemplifica a interação dos sistemas autônomos. Cada AS tem autonomia para rodar qualquer protocolo IGP, mas todos são interligados pelo BGP.

Para ser considerado um bom algoritmo de roteamento ele deve possuir: exatidão,

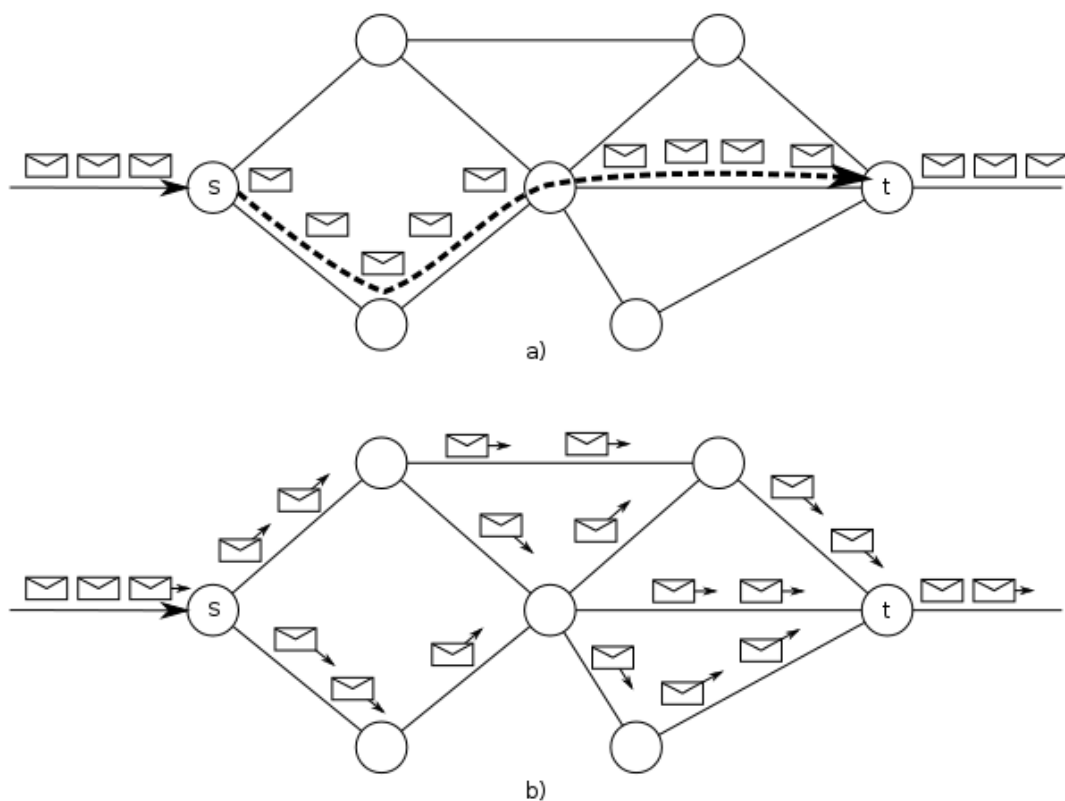


Figura 2.2: Dois tipos de comutação: a) Comutação por circuitos virtuais, b) Comutação por pacotes.

simplicidade, robustez, estabilidade, equidade e eficiência. O algoritmo deve ser capaz de tolerar todo tipo de falha de *hardware*, software e alterações de topologias sem que todo o sistema seja interrompido. Além disto, o algoritmo precisa encontrar um conjunto de rotas viáveis dentro de um tempo finito sem que essas rotas monopolizem os recursos da rede [54]. Assim, para o roteamento ser eficiente espera-se que o caminho sugerido otimize alguma métrica. Espera-se que o roteamento reduza o atraso e maximize a vazão. Para isso, a cada interface do roteador é atribuído um custo baseado em diversas métricas, como a latência, numero de saltos ou capacidade do canal.

Para descobrir a melhor rota, os roteadores executam um protocolo de roteamento IGP. Nos protocolos IGP tradicionais, os protocolos trocam mensagens de forma que consigam construir a topologia completa da rede com todos os custos e então executam um algoritmo para descoberta do menor caminho como *Bellman-Ford* ou *Dijkstra*. Estes algoritmos permitem a construção de uma estrutura chamada tabela de roteamento que representa a interface que o roteador deve encaminhar o pacote baseado em seu endereço de destino [2].

Outro ponto importante é que os métodos de roteamento podem envolver vários problemas de teoria dos grafos, como o problema de encontrar o menor caminho e arvores de

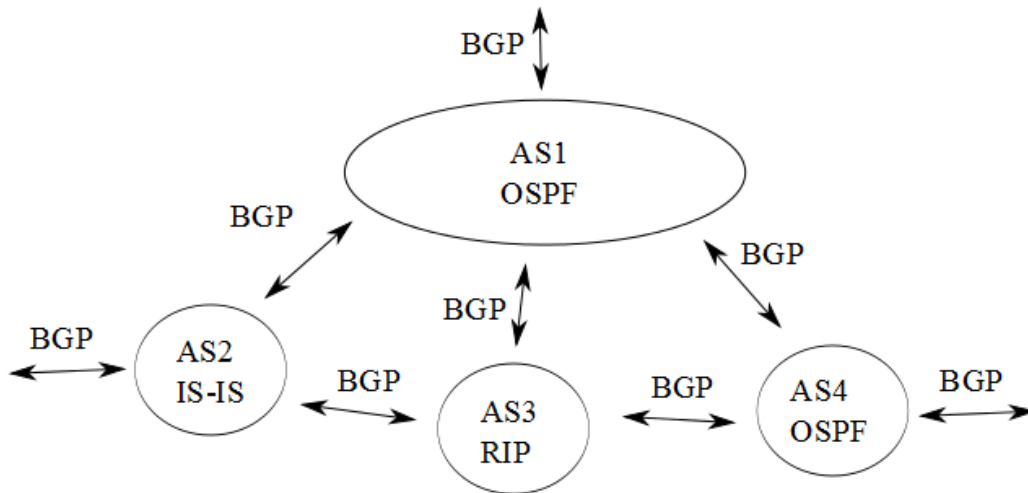


Figura 2.3: Estrutura da interconexão de redes.

custo mínimo, assuntos que serão discutidos nas seções seguintes.

A maior motivação para a otimização de rede vem da perspectiva de um futuro onde milhares de usuários utilizam aplicações de tempo real em dispositivos móveis e a administração da rede precisa encontrar uma solução economicamente viável para encontrar as melhores formas de transportar estas informações.

2.2 Modelando a rede com grafos

Muitos problemas de redes podem ser modelados com grafos ponderados [54]. A Figura 2.4 mostra a representação de uma rede (a) e sua forma gráfica (b). Assim, em um grafo bidirecional ponderado $G(V, E, \Omega)$ os vértices $v \in V$ são abstrações de roteadores, *hosts* e barramento. As aresta $e \in E$ são a representação dos enlaces dedicados que conectam cada nó da rede enquanto que os custos de se transmitir neste enlace são representados pelos peso $\omega \in \Omega$ em cada vértice. A Figura 2.5 mostra diversas topologias ponto-a-ponto sendo representadas na forma de grafos [17].

Como os protocolos geralmente trabalham com algum algoritmo de menor caminho, cada vértice possui um custo ω associado a cada interface representando o custo de se transmitir por determinada interface. Assim, uma função peso $\omega : E \rightarrow \mathbb{N}$ relaciona o custo de transmissão a uma métrica que se deseja otimizar [54].

Com base nestas definições algumas aplicações, como o *multicast* precisam criar Árvores Geradoras Mínimas (MST - *Minimum Spanning Tree*) para uma transmissão eficiente. Assim, uma árvore T construída a partir de um grafo G possui a menor custo igual a soma

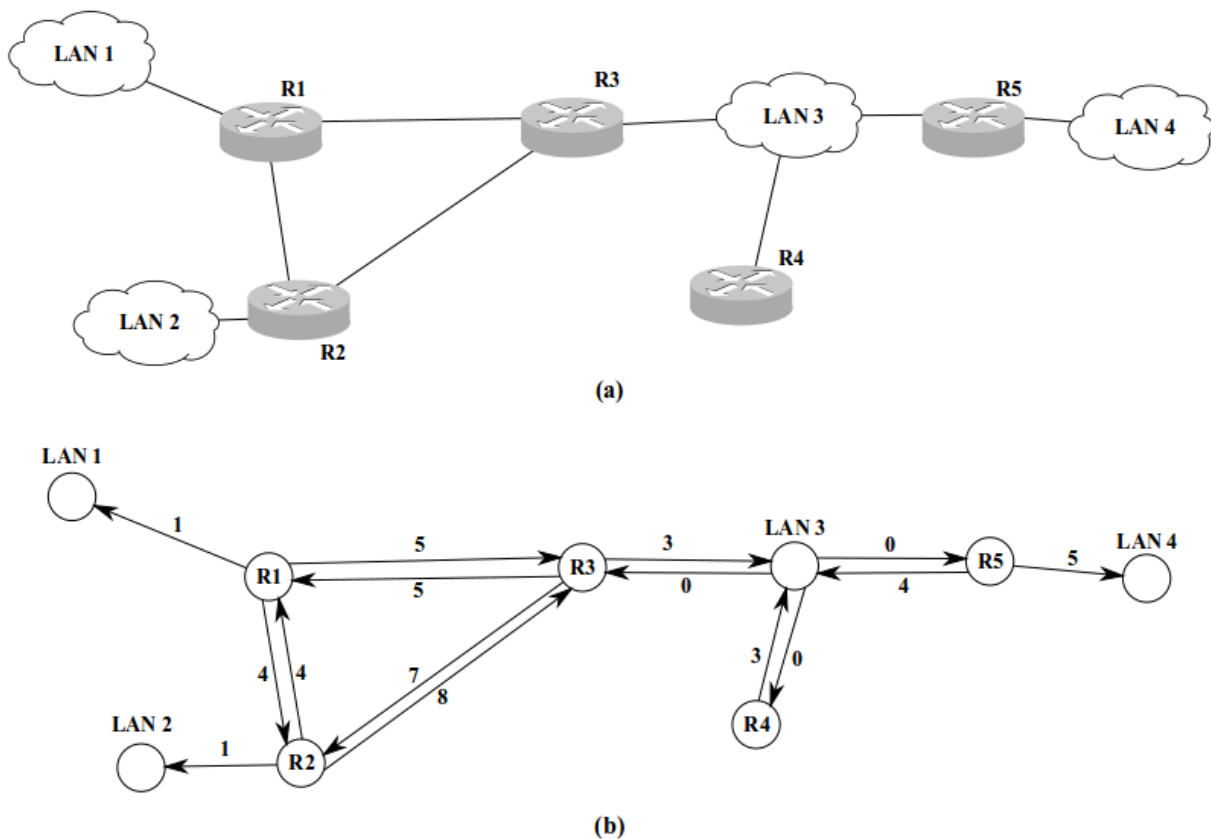


Figura 2.4: (a) Um exemplo de rede de comunicações. (b) A representação na forma de grafo de (a).

dos pesos de cada aresta na árvore. Algoritmos como o algoritmo de *Kruskal* e o algoritmo de *Prim* encontram uma MST em tempo polinomial de $O(E \log V)$ [12].

Outro problema fundamental de rede é encontrar o melhor caminho para entregar um pacote a um único destino. Os algoritmos para este problema são chamados de Algoritmos de Menor Caminho (SP - *Shortest Path*).

Dado um grafo $G(E, V, \Omega)$ e uma origem $s \in V$, os algoritmos de menor caminho encontram o menor caminho para todos os destinos da rede. A saída do algoritmo é um conjunto de enlaces $R \subset V$ que conecta s a $a, \forall a \in V, a \neq s$ e que minimiza a função $\phi(R) = \sum_{(a,b) \in R} w(a,b)$. Ou seja, escolher o caminho da origem s ao destino a cujo a soma dos pesos de todos os enlaces neste caminho seja a mínima possível.

Os dois algoritmos mais usados para encontrar o menor caminho são o algoritmo de *Bellman-Ford* e algoritmo de *Dijkstra*. O primeiro possui complexidade igual a $O(VE)$ e é utilizado no protocolo RIP. O segundo, mostrado na Figura 2.6, apresenta complexidade de $O(E \log V)$ e é utilizado no protocolo OSPF [54] [31].

A Figura 2.7 mostra um exemplo da execução do algoritmo de Dijkstra. O algoritmo

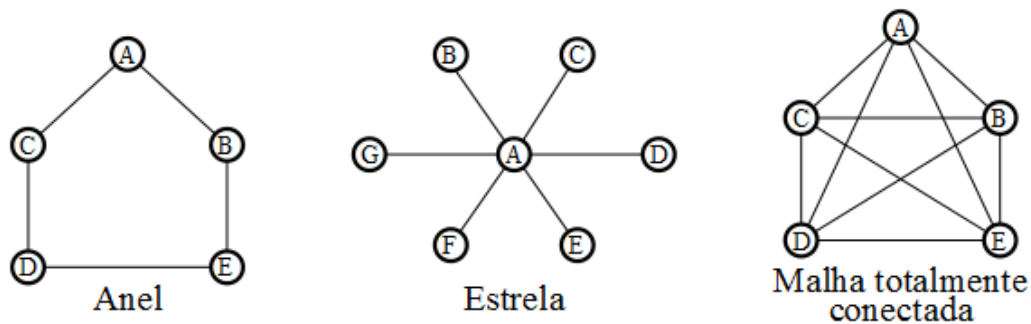


Figura 2.5: Topologias em grafos

```

DIJKSTRA (G, w, S)
1 FOR EACH VERTEX  $v \in V[G]$  DO
2    $d[v] \leftarrow \infty$ 
3    $\pi[v] \leftarrow NIL$ 
4  $d[s] \leftarrow 0$ 
5  $S \leftarrow \emptyset$ 
6  $Q \leftarrow V[G]$ 
7 WHILE  $Q \neq \emptyset$  DO
8    $u \leftarrow \text{EXTRACT-MIN}(Q)$ 
9    $S \leftarrow S \cup u$ 
10  FOR EACH VERTEX  $v \in Adj[u]$  DO
11    IF  $d[v] > d[u] + w(u, v)$  THEN
12       $d[v] \leftarrow d[u] + w(u, v)$ 
13       $\pi[v] \leftarrow u$ 

```

Figura 2.6: Algoritmo de Dijkstra

utiliza uma estratégia gulosa com busca em profundidade, escolhendo a cada iteração a primeira menor rota. Partindo do nó a , o algoritmo descobre o menor caminho para todos os nós em tempo $O(E + V \log V)$. Neste exemplo, partindo do nó a o algoritmo coloca todos os seus vizinhos em sua MST, pois até o momento estes são os menores caminhos partindo de a . Após isto, ele compara seus vizinhos e escolhe o de menor custo total (nó f) para ser o próximo nó a ser iterado. O nó f compara o menor custo de todos os seus vizinhos com o menor custo total do nó a até os nós que foram comparados. No terceiro passo, os vizinhos do nó são os nós b, d e e , porém o custo de $a - e$ é menor do que o custo de $a - f - e$. Por isto esta maior rota não é incluída na árvore. Desta forma, nó a a nós são analisadas e comparadas para encontrar uma árvore que representa todos os menores caminhos partindo de uma origem a .

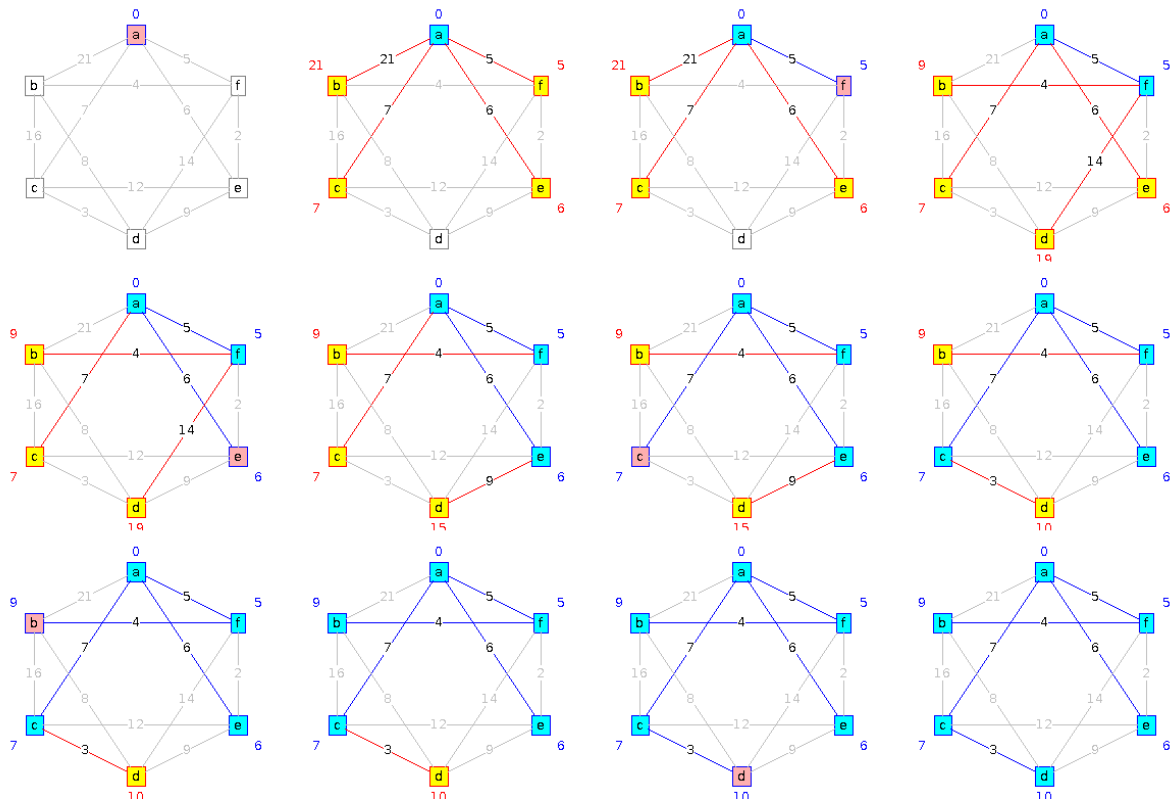


Figura 2.7: Exemplo do funcionamento do Dijkstra

2.3 Protocolos de roteamento

Diversos tipos de roteamento *unicast* foram desenvolvidos no passado. O que os difere é o modo como armazenam a informação de topologia da rede [17]. Os Protocolos de estado de enlace armazenam a topologia na forma de grafo e protocolos de vetor de distância armazenam a topologia do menor caminho em um vetor. Outra classe de protocolos chamada de biologicamente inspirados, tentam simular o comportamento de enxames de formigas, abelhas ou outros agentes biológicos para a descoberta e manutenção de rotas [4]. Este trabalho considera apenas os protocolos de estado de enlace.

2.3.1 *Broadcast e flooding*

Todos os protocolos IGP precisam se comunicar com os vizinhos para manter atualizada as informações sobre a topologia da rede. A principal forma de propagar esta informação é através de uma técnica chamada inundação (*flooding*).

Quando o nó deseja inundar a rede, o mesmo pacote é transmitido por todas as interfaces do dispositivo e todos os nós que receberem esta mensagem devem a encaminhar

por todas as suas interfaces. Esta forma simples de roteamento permite que todos os nós da rede recebam as informações transmitidas.

A Figura 2.8 apresenta um exemplo de *flooding* utilizando números de sequência. Ao receber um pacote de *flooding* o nó incrementa o número de sequência do pacote antes de encaminhá-lo. Quando um nó recebe um pacote com um número de sequência igual ou menor ao já recebido ele apenas o descarta. Esta técnica é utilizada para evitar que os roteadores transmitam pacotes para endereços que já o receberam.

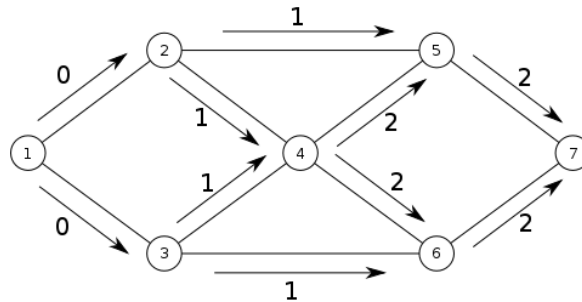


Figura 2.8: Exemplo de *flooding* utilizando números de sequência

O *flooding* é considerado uma forma de roteamento bastante robusta pois a entrega do pacote não depende do conhecimento da topologia, assim, se o destino estiver na rede a chance de um pacote não ser entregue será bastante baixa. Porém, apesar da sua grande confiabilidade, esta técnica pode gerar uma grande quantidade de mensagens redundantes causando desperdício de largura de banda. Além disso, podem ocorrer ciclos infinitos se não for utilizada algum mecanismo para limitar o numero de encaminhamentos.

2.3.2 OSPF - Open Shortest Path First

Os primeiros protocolos de roteamento utilizavam um algoritmo de vetor de distância baseado no algoritmo distribuído de *Bellman-Ford* [25], mas possuíam problemas com escalabilidade e lenta convergência [54]. Devido a isto a Força Tarefa de Engenharia da Internet (IETF - *Internet Engeneering Task Force*) propôs um protocolo de estado de enlace chamado OSPF [37]. Posteriormente um protocolo muito semelhante chamado Sistema Intermediário - Sistema Intermediário (IS-IS *Intermediate System - Intermediate System*) foi padronizado e a Organização Internacional de Padronização (ISO - *International Standards Organization*) passa a utilizá-lo no Sistema de Interconexão Aberto (OSI - *Open System Interconnection*) [54]. Atualmente o IS-IS é mais utilizado em Provedores de Serviço de Internet (ISPs - *Internet Service Provider*) enquanto o OSPF é utilizado em redes empresariais [54].

O OSPF é um protocolo de roteamento intradomínio baseado no algoritmo de roteamento por estado de enlace. Ele opera abstraindo todos os elementos da rede em um grafo direcionado ponderado, onde o peso de cada enlace pode ser baseado em diferentes características da rede, como atraso ou capacidade do enlace. Os pesos das duas direções de um enlace também podem ser diferentes.

Recuperando a topologia da rede como um grafo, o OSPF utiliza um algoritmo de menor caminho como o de *Dijkstra* para calcular o caminho mais curto. Ainda assim, podem existir mais de um menor caminho para os destinos. Neste caso o roteador distribui os pacotes com aquele destino igualmente por todos os caminhos. Este balanceamento do tráfego é chamado de Múltiplos caminhos de Custo Igual (ECMP - *Equal Cost Multi-Path*) [38].

Conseqüentemente, para lidar com o grande crescimento das redes, o OSPF divide a rede em áreas numeradas. Isto ocorre porque, dependendo do tamanho da rede, o roteador não precisa conhecer todos os destinos da rede. Para isto, o OSPF foi projetado para trabalhar com hierarquias de áreas. Cada área é uma rede ou conjunto de redes. Os roteadores conhecem apenas a topologia da área a qual pertencem, todo o resto fica oculto.

Cada sistema autônomo possui uma área 0 chamada de *backbone* e todas as outras áreas estão conectadas a esta área. Dentro de uma área, cada roteador possui o mesmo banco de dados de estado do enlace e utiliza o mesmo algoritmo de menor caminho que calcula os menores caminhos para os nós daquela área e para o roteador do *backbone* que se conecta a esta área.

O OSPF define quatro classes de roteadores em uma áreas:

- **Roteadores Internos:** São todos os roteadores que estão dentro de uma mesma área.
- **Roteadores de Borda de área:** São os roteadores de uma área que se conectam ao *backbone*.
- **Roteadores do *Backbone*:** São todos os roteadores dentro da área 0.
- **Roteadores de Borda do AS:** São os roteadores que ligam o AS a outro AS.

Um roteador pode pertencer a mais de um tipo. Por exemplo, um roteador interno também pode ser um roteador de borda de área. A Figura 2.9 apresenta um AS dividido em diversas áreas com seus quatro tipos de roteadores.

Quando um roteador em uma área é iniciado, ele envia uma mensagem do tipo *HELLO* por todas as suas interfaces. Com base na resposta o roteador sabe quem são seus vizinhos.

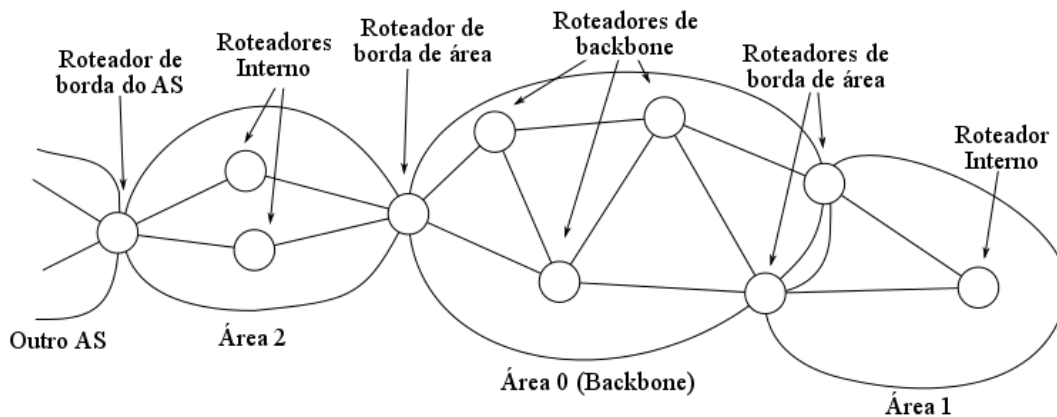


Figura 2.9: Um AS dividido em áreas com seus vários tipos de roteadores.

Dentro de uma área o OSPF escolhe um roteador para ser seu roteador designado. Este roteador é considerado adjacente a todos os outros da sua rede. Periodicamente, os roteadores trocam mensagem do tipo *LINK STATE UPDATE* com seu roteador adjacente de forma a manter suas informações de estado e custo atualizadas no banco de dados de sua área. Nestas mensagens, um número de sequência e confirmações são utilizados. Com base nestas informações, cada roteador constrói um grafo para sua área e calcula o menor caminho através do algoritmo de *Dijkstra*.

Em 2003 foi proposto uma extensão para o OSPF para implementação de engenharia de tráfego [29]. Estas extensões permitem incorporar a carga do tráfego nas informações de estado do enlace e decisões da escolha da rota. Mas estas extensões exigem modificações dos roteadores para que aceitem os novos tipos de mensagem.

Outra adaptação proposta foi feita por Fortz em [18], onde apenas ajustando os pesos dos enlaces de forma dinâmica e utilizando protocolos tradicionais é possível construir uma rede robusta o suficiente para as flutuações de tráfego e falhas.

Enquanto isto outras pesquisas abordam novas arquiteturas afim de otimizar ao máximo a utilização das redes, acreditando que esta arquitetura de roteamento baseadas em pesos estáticos não suportará o tráfego da Internet do futuro [24] [5].

2.4 Engenharia de tráfego e otimização

Geralmente os protocolos tradicionais (OSPF e IS-IS) escolhem o melhor caminho baseado em pesos estáticos configurado pelo operador. Encontrar os mecanismos mais eficientes de moldar o tráfego às condições da rede de forma que algumas métricas como utilização e latência mantenham-se em intervalos aceitáveis, é chamado de engenharia de tráfego.

Na engenharia de tráfego o termo fluxo é utilizado para designar a sequência de transmissão de todos os pacotes que possuam a mesma origem e o mesmo destino. No campo de pesquisa operacional, alguns autores [2] [40] identificam os fluxos como *commodities*, ou seja, toda a demanda de envio e recepção.

De acordo com [2], um dos padrões para os pesos dos enlaces estabelecido pela CISCO *Networks* é definido como $\phi = 1/c_a, \forall a \in A$, sendo A o conjunto dos enlaces e c a capacidade do enlace. Isto deixa os enlaces de menor capacidade com o custo maior, resultando em enlaces menos sobrecarregados e reduzindo o atraso.

O exemplo da Figura 2.10 mostra um exemplo de escolha de pesos para o roteamento. A escolha do peso da rede foram feita utilizando o inverso da capacidade do enlace e depois multiplicados por 100 para transformá-lo em números inteiros. Os números entre parênteses possuem a seguinte notação (c_a, ω_a, l_a) , onde c_a representa a capacidade do enlace $a \in A$, sendo A o conjunto de roteadores na rede, ω_a o peso do enlace s e l_a a carga sobre o enlace a . Caso a

Se considerar-se a métrica principal igual a utilização máxima da rede $U^{max} = \max_{a \in A} \frac{l_a}{c_a}$, neste exemplo todo o tráfego percorrerá o caminho com a maior capacidade $c_a = 5$ gerando uma utilização de $U^{max} = 80\%$ enquanto os outros enlaces de menor capacidade ficam inutilizados.

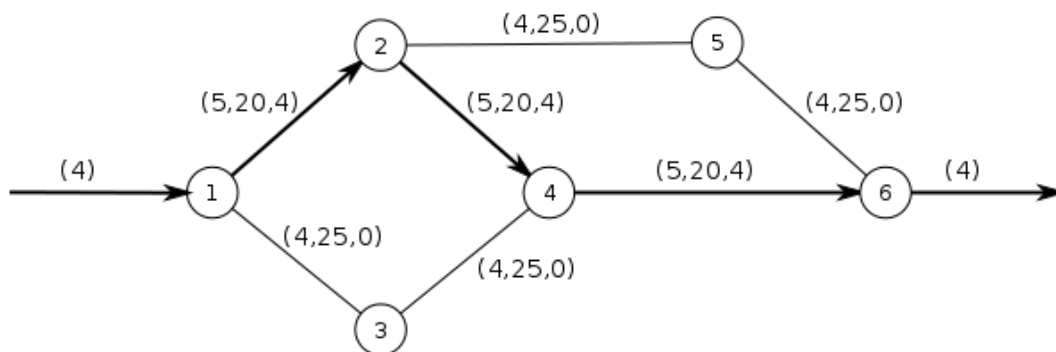


Figura 2.10: Exemplo de roteamento com peso igual ao inverso da capacidade

Já a Figura 2.11 mostra um exemplo onde foi reduzida e os pesos dos enlaces da mesma rede foram escolhidos todos iguais a 1. Neste caso há mais de um menor caminho e o tráfego é dividido arbitrariamente a cada bifurcação. A forma como o tráfego será dividido depende do protocolo utilizado. Neste exemplo a demanda de tráfego é definida com quatro unidades de tráfego. O nó 1 encaminha 2 unidades para o nó 2 e 2 unidades para o nó 3. O nó 2 divide o fluxo recebido enviando 1 unidade para o nó 5 e 1 unidade para o nó 4. O nó 4 recebe 2 unidade de tráfego do nó 3 e mais 1 unidade do nó 2, encaminhando 3 unidades para o nó 6. Por fim, todos os fluxos se encontram no nó 6 que encaminha 4 unidades, sendo 3 vindas do nó 4 e 1 vinda do nó 5. Desta forma,

esta configuração não deixa nenhum enlace ocioso e alcança uma utilização máxima de $U^{max} = 60\%$. O resultado é chamado de ótimo quando não existe nenhum outro resultado melhor do que o apresentado. Este exemplo mostra como o resultado ótimo pode reduzir consideravelmente uma métrica dos enlaces (a utilização máxima, neste caso) sem alterar o hardware da rede.

O protocolo OSPF utiliza-se de um campo de 16 bits para armazenar o peso de cada enlace [38]. Além disto, prevê a situação quando há vários menores caminhos, chamado de ECMP. Este caso exige uma regra de divisão de fluxo e o problema de escolher a proporção em que o tráfego deve ser dividido é um problema NP-completo [40].

O RFC 2328, que padroniza o OSPF, não define uma regra para se aplicar no caso de múltiplos caminhos, mas prevê o seu uso e menciona a divisão do tráfego arbitrariamente quando há mais de um menor caminho. O OSPF não é obrigado a armazenar todos os menores caminhos. Existem aplicação que não podem ter o fluxo dividido, este problema é chamado de único menor caminho (USP - *Unique Shortest Path*).

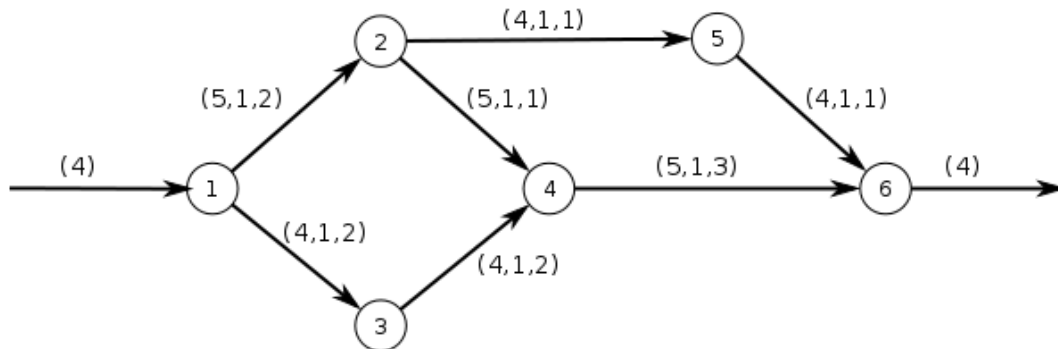


Figura 2.11: Exemplo de roteamento com todos os pesos igual a 1

O *framework* da engenharia de tráfego com OSPF é mostrado na Figura 2.12. Com base em medições da rede, são extraídas informações sobre a topologia e desempenho da rede que são utilizadas para gerar um modelo matemático do roteamento da rede. Em seguida é aplicado alguma técnica de otimização ou heurística neste modelo, afim de se obter um conjunto de custos que satisfazem as restrições do modelo e que são aplicados para otimizar alguma métrica da rede [18].

Os dados para as medições geralmente são obtidos acessando as Bases de Gerenciamento de Informações (MIB - *Management Information Base*) dos roteadores da rede. As métricas de desempenho dependem das aplicações. Assim, para não gerar complexidade desnecessária, a modelagem matemática deve ser uma abstração simplificada com apenas as principais características do sistema em estudo [40].

Algumas tecnologias permitem maior flexibilidade na divisão dos caminhos com a criação de rotas explícitas gerando maior controle sobre o fluxo do que um sistema apenas

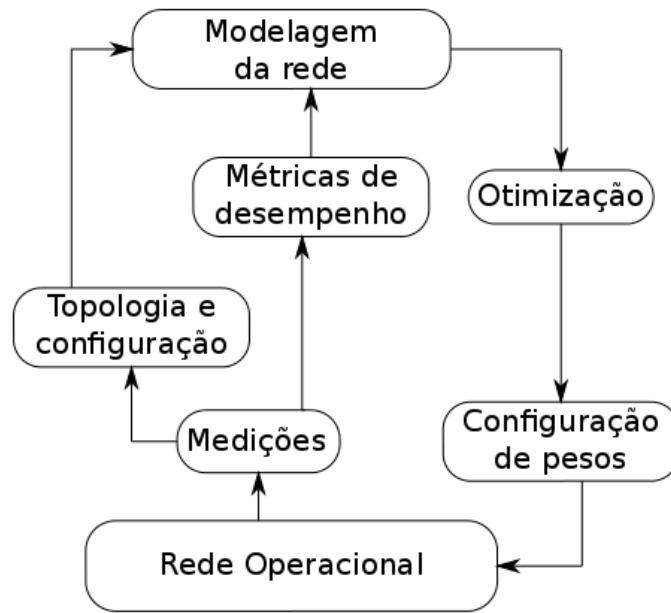


Figura 2.12: Framework de engenharia de tráfego com OSPF [18]

de pesos por enlace. Tais protocolos requerem um hardware específico ou reconfiguração da rede aumentando a complexidade e o custo do procedimento.

Porém, comparado com o sistema de roteamento ótimo, Fortz [18] mostra que é possível conseguir um desempenho apenas 3% inferior escolhendo os pesos de forma correta permitindo até 50 % mais utilização da rede do que o modelo proposto pela Cisco.

Além de maximizar a utilização total da rede, Fortz também analisou a robustez a flutuações do tráfego e falhas, apresentando ótimos resultados. Outro ponto importante foi o desenvolvimento da heurística de busca local que permite que os pesos sejam calculados de forma mais rápida.

2.4.1 O problema de otimização de roteamento

O problema básico do roteamento deve considerar a capacidade da rede e garantir que as demandas dos transmissores sejam satisfeitas, ou seja, entregar todos os pacotes que são injetados na rede com uma restrição específica de QoS [51].

Seja N o conjunto dos nós da rede e f_{ij}^{st} o fluxo de origem em s e destino em t que passa pelo enlace (i, j) . Considere também δ_i^{st} o fluxo de origem s e destino t que chega ao nó i e $-\delta_i^{st}$ o fluxo que sai de i . O problema do roteamento, também chamado de Problema de Fluxo de Rede Multi Produto (MCNFP *Multi-Commodity Network Flow Problem*), considera escolher caminhos pela rede de forma que toda sua demanda alcance seu destino [8].

O objetivo de um programa linear do MCNFP apresentado a seguir, é minimizar a função custo ϕ , definida pela equação 2.1. A restrição 2.2 garante que todo o fluxo que chegar a um nó deve sair dele, a menos que o nó seja o destino do fluxo representado pela matriz de demanda por $\delta(st)$ como destino do fluxo ou por $-\delta(st)$ como origem do fluxo enquanto a restrição 2.3 define a carga l como sendo a soma de todos os fluxos em um enlace e a equação 2.4 garante que a carga não ultrapasse a capacidade c do enlace (i, j) . Por fim, a restrição 2.5 representa a função linear de custo em partes, que define o comportamento do custo em função da carga e capacidade do enlace para em um conjunto de intervalos de linearidade Z composto de um ou vários segmentos interligados, onde α_z e β_z representam os coeficientes de cada segmento.

$$\min \quad \Phi = \sum_{(i,j) \in N} \phi_{(i,j)} \quad (2.1)$$

$$\sum_{j \in N} f_{(i,j)}^{(s,t)} - \sum_{j \in N} f_{(j,i)}^{(s,t)} = \begin{cases} \delta(st) & \text{se } i = s \\ -\delta(st) & \text{se } i = t \\ 0 & \text{caso contrario} \end{cases} \quad i, j, s, t \in N \quad (2.2)$$

$$l(i, j) = \sum_{(s,t) \in N} f_{(i,j)}^{(s,t)} \quad (2.3)$$

$$l_{ij} \leq c_{ij} \quad , \quad (i, j) \in N \quad (2.4)$$

$$\phi \geq \alpha_z l_{ij} - \beta_z c_{ij} \quad (i, j) \in A, \quad z \in Z \quad (2.5)$$

Considerando que o problema de otimização de pesos é NP-completo [18], uma alternativa é utilizar aproximações. Altin et al. em [2] cita várias técnicas e trabalhos com aproximações utilizadas para resolver problemas de otimização em fluxos de rede utilizando programação linear, linear inteira, mista e de ordens superiores. Entre as técnicas citadas estão a *Local Search*, *Colum Generation*, algoritmos genéticos, aproximação por *Simulated Annealing*, *Lagrangian Relaxation* entre outros.

Em [18] Fortz e Thorup consideram diversas vantagens em se realizar a otimização de protocolos tradicionais como o OSPF e IS-IS. Dentre elas pode-se citar a estabilidade do protocolo, o baixo *overhead* e o baixo custo de implementação. Nestas técnicas o ponto chave é a manipulação dos pesos dos enlaces de forma que os pacotes encontrem os melhores caminhos, minimizando, desta forma, a utilização máxima e o atraso total da rede. De acordo com o RFC 2328 [38] o protocolo OSPF possui 16 bits destinado a divulgação da métrica do enlace. Porém, problema de encontrar o menor numero inteiro para se utilizar como métrica é NP-difícil [9].

A decisão de escolher o custo do enlace não é um trabalho trivial, mas dependendo dos valores escolhidos o roteamento pode aumentar a capacidade máxima da rede. Fortz

et al. [18] propuseram uma função em partes para a determinação do peso como uma aproximação do tamanho de uma fila M/M/1 com chegadas de Poisson, apresentando uma boa distribuição da carga pela rede, como apresentado no trabalho de Balon et al. [6].

2.5 Resumo do capítulo

Este capítulo apresentou as principais técnicas utilizadas para realizar a descoberta de rotas em uma rede IP. Foram apresentados os conceitos de sistema autônomo e protocolos intradomínio. Também foi apresentado o protocolo de roteamento OSPF e a teoria de grafos utilizada para modelar uma rede.

Por fim, este capítulo apresenta a engenharia de tráfego como uma poderosa ferramenta para otimizar uma rede de computadores e sua relação com protocolos IGP na busca pelo menor caminho.

No próximo capítulo será apresentado o vídeo digital, as formas de transmiti-lo, suas principais características, métricas de qualidade para este tipo de tráfego e uma nova aplicação chamada *Free Viewpoint Television* (FTV).

Capítulo 3

Caracterização do Tráfego de Vídeo Digital

Este capítulo apresenta a fundamentação teórica das tecnologias de vídeo digital. Nas Seções 3.1 e 3.2 é feita uma breve apresentação das principais técnicas de compressão que permitiram o uso do vídeo na Internet.

A qualidade de transmissão da mídia digital é apresentado na Seção 3.3 e em seguida é realizado a caracterização do tráfego autossimilar e da sua relação com o tráfego de vídeo nas Seções 3.5 e 3.6.

Este capítulo encerra-se com a caracterização do tráfego da aplicação *Free Viewpoint Television* (FTV) na Seção 3.4 que será utilizado para validar o a proposta principal deste trabalho.

3.1 O Vídeo digital

O vídeo digital ganhou mais atenção a partir do primeiro padrão do Grupo de Especialistas em Imagens com Movimento (MPEG - *Moving Picture Experts Group*) impulsionando o desenvolvimento de sistemas de transmissão e armazenamento de fluxo de vídeo digital. E, em 1995 a *Progressive Networks* que logo muda seu nome para *Real Networks* realiza a primeira difusão de um vídeo ao vivo pela Internet através de seu software *RealAudio Player* [57].

O vídeo digital pode ser representado como uma sequência de imagens digitais e estas imagens são representadas por uma matriz de *pixels*. Estes *pixels*, sendo os menores elementos da imagem digital, geralmente carregam informação sobre a cor de um ponto da imagem.

Nestas imagens digitais, as cores são codificadas utilizando uma certa quantidade de bits por pixel. Quanto mais bits por pixel, mais cores o pixel pode representar, porém

mais memória é necessária para armazenar a imagem. Devido as limitações humanas para enxergar variações pequenas de cores, raramente se codifica as intensidades de cores com mais do que 3 bytes de cores (2^{24} cores). Um exemplo de padrão que utiliza 3 bytes é o padrão RGB (*Red-Green-Blue*), onde as cores vermelho, verde e azul são codificados em 255 níveis diferentes. Outras características do vídeo são a taxa de quadros por segundo, proporção, numero de linhas e tipo apresentação do quadro são definidos pelo padrão utilizado.

A tabela 3.1 mostra os diferentes formatos de TV digital suportado pela ATSC (*Advanced Television System Committee*), órgão formado por várias empresas e pesquisadores responsáveis pelos padrões para difusão de HDTV (*High Definition TV*). Esta tabela apresenta o nome do padrão, a taxa de exibição de quadros em quadros por segundos (fps - *frame per second*), se a exibição do quadro é Entrelaçado (I) ou Progressivo (P), o número de linhas e a proporção da tela medida em colunas por linhas.

Tabela 3.1: Padrões de TV digital [35]

Padrão	Taxa de exibição de quadros	Linhas	Proporção
SDTV	60I 50I	480	4:3
EDTV	60P 60I 50P 50I 30P 25P 24P	480 ou 720	4:3 ou 16:9
HDTV	60P 60I 50P 50I 30P 25P 24P	720 ou 1080	16:9
UHDTV	120P	2160	16:9

O formato SDTV (*Standard Definition TV*) corresponde a um sinal de vídeo digital com qualidade semelhante ao padrão norte americano de televisão analógica NTSC (*National Television System Committee*) com proporção de quadro de 4:3 e 480 linhas. Este formato permite a exibição das imagens com quadros que alternam a exibição de suas linhas pares e ímpares a 60 fps ou 50 fps, técnica chamada de entrelaçamento, resultando em uma taxa efetiva de 30 fps e 25 fps respectivamente. Estas frequências dependem da frequência do sistema elétrico adotado no país: 50 Hz ou 60 Hz.

O formato EDTV (*Enhanced Definition TV*) surgiu para melhorar a qualidade do vídeo permitindo uma quadro mais largo e uma melhor exibição dos quadros. O desenvolvimento da exibição de quadros progressiva foi um grande avanço que permite o conteúdo do quadro ser gerado de uma vez reduzindo anomalias na formação da imagem. Atualmente é o padrão utilizado em DVDs e suportado pela TV Digital Brasileira [48].

O HDTV (*High Definition TV*) é um formato de vídeo superior ao EDTV que suporta uma proporção de 16:9 e quadros com 720 linhas. O quadro é progressivo ou progressivas. A codificação padrão é o H.264 e suporta 5.1 canais do sistema Dolby Surround.

O UHDTV (*Ultra High Definition TV*) é a última geração de TV digital para um conjunto de padrões para Resoluções mais altas que a HDTV, entre elas o formato 4K com 2160 linhas. Ele suporta quadros com razão 16:9 e 256:135, 12 bits de cores, formato

progressivo e taxa de quadros igual a 120 fps. Este novo formato de vídeo requer uma largura de banda de transmissão muito mais elevada do que os formatos atuais.

3.1.1 Compressão de vídeo

Aplicações de vídeo são bem conhecidas por possuírem uma taxa de transmissão extremamente alta. O Vídeo em formato bruto exige uma largura de banda tão alta que pode não ser suportado pelos padrões atuais de rede local, logo deve ser utilizada alguma técnica para reduzir a largura de banda necessária para sua transmissão. De acordo com o padrão ITU-R Rec.601 [27], o vídeo NTSC pode ser codificado sem compressão utilizando 16 bits de cores (65536 cores) com uma taxa de 30 fps e quadros com 525×858 pixels. Assim, é gerado um fluxo de $16 \times 30 \times 525 \times 858 \cong 216$ Mbps, sem considerar o som. Esta taxa é maior do que duas vezes a suportada pelo padrão de rede local *Fast Ethernet* [26].

A partir da necessidade de reduzir o armazenamento e a taxa de transmissão de vídeos e imagens nas redes, foram desenvolvidas diversas técnicas de compressão. Os algoritmos destas técnicas podem ser divididos em dois tipos: *lossless* (sem perdas) e *lossy* (com perdas) como descrito a seguir [35]:

Lossless: Os algoritmos do tipo *lossless* realizam a compressão e descompressão dos dados sem perda de informação durante o processo. Algoritmos como a Codificação de Huffman e o Algoritmo de Shannon-Fano utilizam estruturas de dados avançadas para reduzir os bits de cada caractere.

Lossy: Devido a limitação dos olhos e ouvidos humanos, informações como imagens, áudio e vídeo podem sofrer pequenas perdas sem que a compreensão da informação seja comprometido. Desta forma as codificações que geram perdas, também chamadas de *Lossy*, possuem uma taxa de compressão muito maior. Geralmente estes algoritmos utilizam uma transformada como a transformada discreta do Cosseno (DCT - *Discrete Cosine Transform*) para reduzir a autocorrelação da entrada, reduzindo a repetição de padrões. Porém, a partir de um certo elemento, o resultado da transformada pode ser muito próximo de zero. Estes valores possuem informações imperceptíveis aos olhos humanos podendo ser descartada. Além da DCT também pode ser usada outras transformadas a transformada de *Wavelet*.

3.2 Transmissão de vídeo na Internet

Somente a partir da década de 90 as novas aplicações de vídeo passaram a compor um novo tipo tráfego na Internet, exigindo mais recursos do que o oferecido pelo serviço de melhor esforço, como por exemplo, garantias de largura de banda e atrasos máximos.

Após o avanço das técnicas de compressão de vídeo foi possível transmiti-lo pela Internet, porém sob diversos requisitos de qualidade. Geralmente em aplicações com áudio comprimido, a largura de banda da telefonia e áudio não atingem mais do que 200 Kbps, porém, a transmissão de vídeo exige muito mais largura de banda. Além disto, quanto maior a qualidade do vídeo, maior a largura de banda necessária para sua transmissão [35]. De acordo com Tanenbaum é tolerável que o fluxo do vídeo ao vivo tenha atrasos de até 15 segundos [54] porém, variações do atraso que são induzidos pela comutação de pacotes devem ser suprimidas através de *buffers* no receptor. Caso o vídeo seja interativo, além do fluxo ser bidirecional, a tolerância a atrasos devem ser bem menores.

É consenso na literatura básica [31], [17], [53] que o serviço de melhor esforço (*best-effort*) da Internet não consegue oferecer sozinho qualidade na entrega dos pacotes de vídeo, uma vez que uma das características principais do projeto inicial da Internet era não oferecer serviços diferenciados para diferentes pacotes. Por isso o IP sozinho não oferece controle de erro, nem garantia de entrega, atraso e *jitter* mínimos ou garantias de larguras de banda, sendo estes serviços oferecidos por outros protocolos ou aplicações. Além disso, por se basear na comutação de pacotes de tamanhos variados, o pacote ao chegarem no roteador são enfileirados na memória e aguardam a disponibilidade do comutador para que possam ser enviados ao próximo destino gerando atrasos variados [17] [31]. É por este processo que a comutação de pacotes também é conhecida como *store-and-forward* e a variação estatística do atraso é chamada de *jitter*.

Neste sistema, a taxa de chegada de pacotes é caracterizada por um processo estocástico, o que significa que a chegada e processamento dos pacotes no comutador é aleatória. Sob certas condições, como quando a taxa de chegada é superior a taxa de encaminhamento do nó, a memória de sua fila pode ser insuficiente e o dispositivo começa a descartar os pacotes que chegam. Esta condição é chamada congestionamento [53].

Na transmissão de vídeo, para lidar com a perda de quadros o receptor precisa utilizar alguma técnica de interpolação. Interpolação são técnicas de processamento digital de vídeo para criar novos quadros que faltam através de algum algoritmos de estimação de velocidade.

Na interpolação de quadros o receptor, ao detectar a falta do quadro gera um novo quadro intermediário entre os já existentes, permitindo que uma animação seja executada mais suave e clara [35].

A Figura 3.1 mostra o processo de interpolação. A aplicação realiza a predição do movimento baseado na diferença de dois quadros preenchendo a falha causada pela perda.

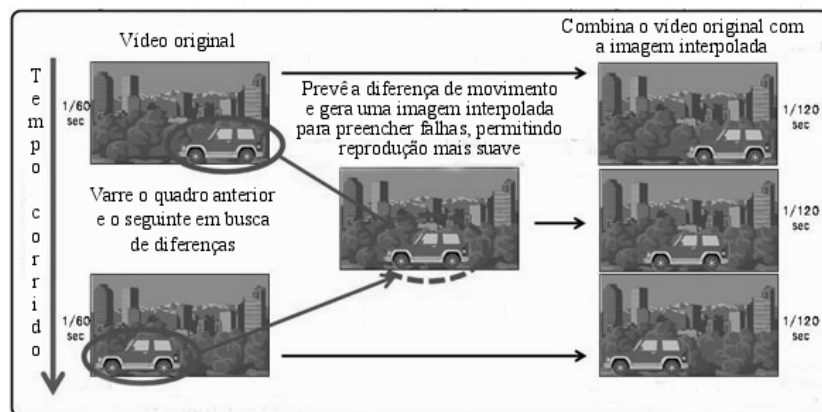


Figura 3.1: Interpolação de quadros [35]

3.2.1 Distribuição de vídeo

Uma das formas de classificação do fluxo de vídeo é de acordo com a forma como ele é disponibilizado [17], conforme descrito a seguir:

Fluxo de vídeo armazenado: Também chamado de vídeo sob demanda, refere-se ao vídeo gravado e armazenado em um servidor que recebe o pedido de transmissão dos usuários na rede. O servidor responde a solicitação enviando uma sequência de pacotes com os dados do vídeo ao cliente que por sua vez o decodificará e exibirá ao usuário. Atualmente os principais padrões de codificação são H.264 e MPEG-4 [35], mas a recomendação H.265, lançada em abril de 2013 tem como objetivo de ser a próxima geração de codificação [28]. Geralmente neste tipo de vídeo a mídia o servidor transmite o vídeo a uma taxa maior do que a de reprodução. O vídeo é armazenado em *buffer*, reduzindo o efeito da flutuação da rede, enquanto é reproduzido a uma taxa constante. Serviços como ©YouTube e ©Netflix são exemplos de serviços de distribuição de vídeo sob demanda.

Fluxo contínuo de vídeo ao vivo: Popularmente conhecido como vídeo ao vivo, este fluxo pode ser representado por um programa de televisão ao vivo. A Figura 3.2 mostra um exemplo de distribuição deste tipo de vídeo. O vídeo é capturado, codificado e transmitido imediatamente pela rede na mesma taxa em que é gerado para dezenas ou milhares de espectadores. Como o fluxo se caracteriza por ser de uma única fonte para milhares de clientes, a transmissão via multicast seria a solução mais eficiente. Como o TCP não suporta multicast, todo multicast utiliza UDP, logo poderá haver perda de pacotes e cabe a aplicação lidar com estas perdas. Neste tipo de vídeo a codificação deve ser feita em tempo real imediatamente após a captura e antes da transmissão. Geralmente o *buffer* utilizado para vídeo ao vivo

é bastante alto para lidar com todo tipo de *jitter*, podendo inserir até 15 segundos de atraso [54].

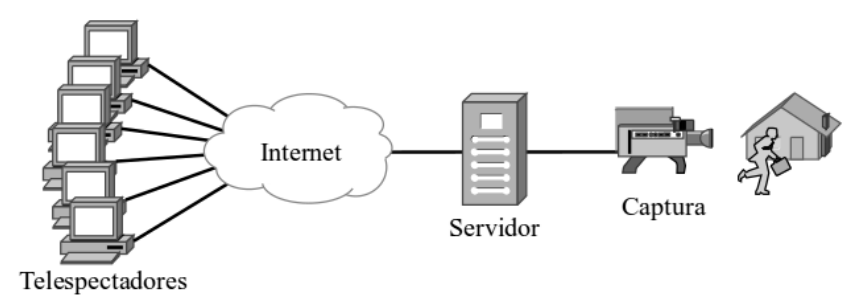


Figura 3.2: Exemplo de distribuição de vídeo ao vivo

Fluxo contínuo interativo ao vivo: Também pode ser chamado de videoconferência. É gerado por aplicações em que a resposta de uma das pontas depende da reação da outra, sendo impossível prever a reação do fluxo na rede. Uma videoconferência no ©Skype é um exemplo de vídeo interativo ao vivo. Neste tipo de distribuição o usuário é um gerador e receptor de tráfego, onde dois ou mais membros conectados a câmeras interagem entre si por difusão. O fluxo de dados deste tipo de vídeo é muito intenso e em ambas as direções. Nesses sistemas, para remoção do *jitter* são necessários mecanismos como marcas de tempo ou circuitos virtuais, pois grandes atrasos podem tornar a comunicação inviável.

Cada tipo de vídeo possui suas características de geração e requerimentos de transmissão para que trafegue pela Internet com qualidade.

3.2.2 Arquiteturas de transmissão de vídeo

Nos últimos anos, diversas formas de transmitir vídeo foram propostas. Baochun Li em [34] faz uma revisão das três principais arquiteturas de distribuição de vídeo ao longo dos últimos 20 anos. A popularização da Internet e a demanda crescente por qualidade geraram muitos desafios para que a transmissão de vídeo fosse possível. As próximas seções apresentam as principais formas de distribuição de vídeo usadas neste período.

Arquitetura cliente-servidor

As primeiras aplicações normalmente consideravam transmitir os fluxos de vídeo por uma arquitetura cliente-servidor onde um único servidor fornecia os vídeos para todos os usuários que o solicitassem. A Figura 3.3 mostra um cliente requisitando um vídeo ao servidor que responde com um fluxo de vídeo que será decodificado e exibido ao usuário. E

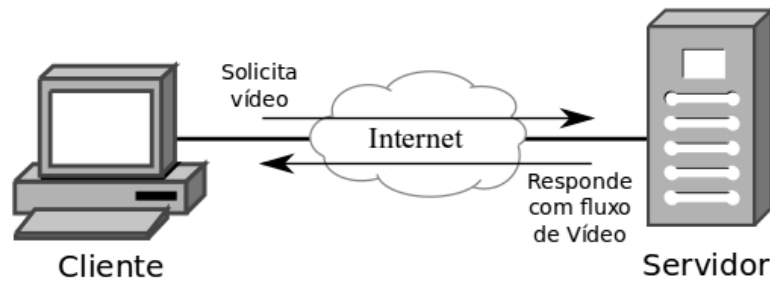


Figura 3.3: Aplicação Cliente Servidor

para cumprir este objetivo foram desenvolvidas diversas técnicas para fornecer qualidade de serviço como o *multicast* [13] [15] para uma transmissão mais eficiente e o Protocolo de Reserva de Recursos (RSVP - *Resource reSerVation Protocol*) [3] para reserva de recursos.

Para a transmissão de aplicações de tempo real outro protocolo de transporte foi desenvolvido. Chamado de Protocolo de Tempo Real (RTP - *Real Time Protocol*) [23], este protocolo propõe o controle do *jitter*, perda de pacotes e marcas de tempo.

Posteriormente para reduzir o efeito de congestionamento, diversas pesquisas [17], [54] propuseram protocolos para controle de taxa dos fluxos de vídeo e outras técnicas como *proxy caching*, que armazena vídeos frequentemente visualizados em servidores distribuídos, aumentando a escalabilidade e reduzindo a latência.

Porém, esta arquitetura possui dificuldade em atender milhares de milhões de clientes, sendo necessário a adoção de outras técnicas de distribuição de conteúdo que comporte a crescente demanda.

Arquitetura *peer-to-peer*

A solução para a escalabilidade foi encontrada na implementação da transmissão de vídeo através da arquitetura Par-a-Par (P2P - *Peer-to-Peer*) que foi proposta e aplicada a sistemas reais como o *PPLive* [42] e *CoolStreaming* [11] durante a primeira década do século XXI.

A Figura 3.4 exemplifica a arquitetura P2P para a transmissão de um vídeo ao vivo. Neste sistema, as estações que participam da recepção, também transmitem pedaços do vídeo reduzindo a carga sobre o servidor.

Este serviço é baseado no protocolo *BitTorrent*, sucessor de populares ferramentas como *Napster* e *Kazaa*. Nesta arquitetura de distribuição é possível fornecer fluxo de vídeo a milhões de usuários consumindo uma modesta quantia de largura de banda onde cada *host* será receptor e transmissor ao mesmo tempo.

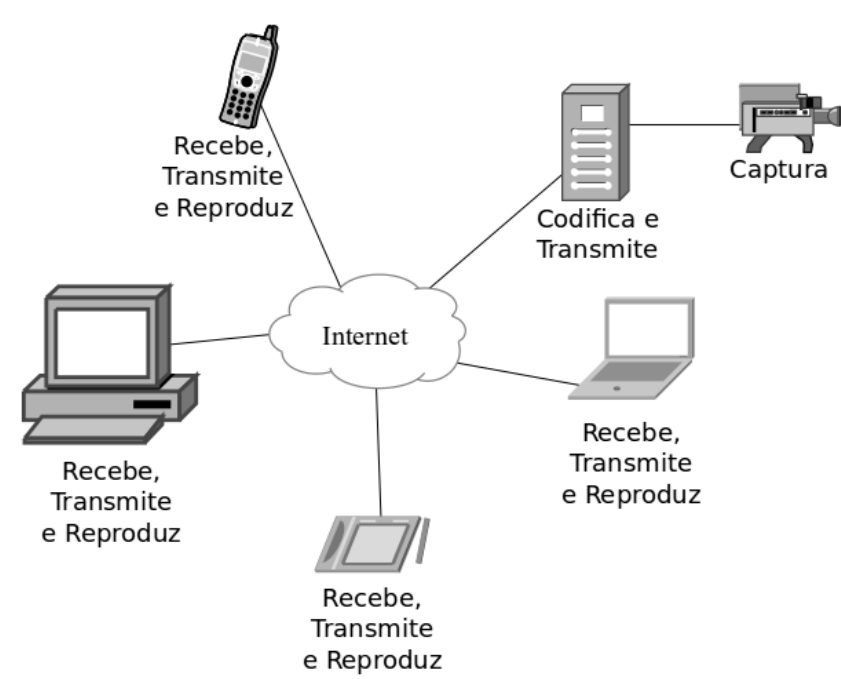


Figura 3.4: Aplicação P2P ao vivo

Porém, para se usar o vídeo por P2P geralmente é necessário instalar um software dedicado que implemente os protocolos necessários. Devido a isto vários problemas de segurança surgem, pois mais portas devem ser liberadas no *firewall* e novas formas de autenticação devem ser criadas para garantir a integridade das informações.

Apesar de existirem centenas de canais de TV ao vivo disponíveis, este serviço não fez sucesso no ocidente, sendo que a maior parte do serviço é oferecido para países asiáticos, como China, Índia e Japão. Geralmente aplicações *torrent* não são bem vistas em países onde as leis de direito autorais são rígidas.

Arquitetura DASH

Visando a conveniência para o usuário assistir vídeo direto do navegador, nos últimos anos foram desenvolvidos protocolos para distribuição de fluxo de vídeo sobre Protocolo de Transferência de Hipertexto (HTTP - *Hypertext Transfer Protocol*). O protocolo mais utilizado atualmente é chamado de DASH (*Dinamic Adaptative Streaming Over HTTP*). Com esta arquitetura o fluxo de vídeo é fragmentado e enviado em pequenas mensagens HTTP que são baixadas enquanto a mídia é executada. O servidor fragmenta o vídeo e transmite como uma série de pequenas mensagens HTML.

Aproveitando-se da escalabilidade e de diversos recursos do próprio HTML, esta técnica permite que o usuário faça uma busca rápida no vídeo durante a execução. Este mecanismo também permite que o vídeo adapte a taxa de transmissão de forma con-

veniente, de acordo com a largura de banda disponível, sendo ideal em ambientes que requerem mídia dinâmica e adaptativa. E como os fragmentos são pequenos, qualquer perda de informação sofrerá pouco impacto na exibição do vídeo [35].

Atualmente este é o padrão adotado por empresas como ©Netflix, ©YouTube e ©Hulu. Seu padrão foi desenvolvido pelo grupo MPEG e se tornou um padrão internacional publicado como ISO/IEC 23009-1:2012 em abril de 2012. [34].

3.3 Qualidade de serviço para vídeo na Internet

Qualidade de serviço (QoS - *Quality of Service*) é um conceito que reflete a garantia de que um conjunto de métricas mantenham-se dentro de certo valor esperado [17], desde que o cliente se comprometa a gerar somente o tráfego acordado. Porém, para o usuário final a qualidade se reflete em um conjunto de métricas mais subjetivas que reflitam a Qualidade da Experiência (QoE - *Quality of Experience*) enquanto utiliza a aplicação.

As métricas de QoE geralmente se baseiam percepção subjetiva do usuário. Elas são tipicamente usadas para prever limites de percepção, tais como a visibilidade de alguns elementos na mídia relacionados ao conforto da utilização da aplicação. Estes limites são guiados pela visibilidade de elementos como borrões, ruídos, blocagem, diferença de cores, tremulações e outros tipos de distorções.

Estas métricas são usadas para definir o limiar da percepção da qualidade da imagem de forma a encontrar o *tradeoff* ótimo entre qualidade e distorção, geralmente associada a compressão do objeto [41].

As principais métricas de QoS utilizadas na Internet são [31]:

- **Atraso:** Também chamado de *delay* (d) representa o tempo total em segundos a partir do momento que é iniciada a transmissão de um pacote na rede até o momento em que ele é completamente recebido pelo destino, incluindo atraso de enfileiramento, atraso de processamento, atraso de propagação e atrasos de transmissão. Outro tipo de atraso também chamado de RTT (*Round Trip Time* - tempo de ida e volta) é o atraso de se enviar uma mensagem e receber sua resposta. Este é o tempo exibido em aplicações como PING.
- **Jitter:** Na literatura o *jitter* também é chamado de variação do atraso ou flutuação estatística do atraso. Na Internet cada pacote pode ter um atraso diferente. A diferença entre estes atrasos é chamado de *jitter*. Esta métrica é muito importante em aplicações multimídia pois reflete na forma como o vídeo é exibido. Em alguns casos, como em transmissões de vídeo ao vivo, é preferível um atraso maior, porém com poucas variações, do que um *jitter* elevado.

- **Largura de banda:** Representa a capacidade (c) máxima teórica em bits por segundo que a rede pode transmitir. Quanto maior a largura de banda disponível, mais informação pode ser trafegada em um menor período de tempo.
- **Carga:** A carga (l) é todo o fluxo encaminhado por um enlace ou roteador em um determinado período de tempo. A utilização de um enlace U é dada pela relação $U = \frac{l}{c}$. Em uma rede de comutação de pacotes, quando a carga ultrapassa a capacidade do enlace o nó começa a descartar pacotes.
- **Confiabilidade:** A confiabilidade representa a garantia que a rede tem de entregar pacotes a seu destino. Uma rede que descarta muitos pacotes devido a falhas ou congestionamento é pouco confiável e obriga que protocolos como o Protocolo de Controle de Transporte (TCP - *Transport Control Protocol*) façam muitas retransmissões ou o destino receberá a informação incompleta ou errada.
- **Disponibilidade:** Já a disponibilidade representa a taxa em que o sistema encontra-se disponível para sua utilização dentro de um intervalo de tempo. Devido a fatores externos, como chuvas ou outros incidentes a rede pode parar de funcionar deixando de ficar disponível aos usuários. Por exemplo, uma disponibilidade de 99,99% representa uma rede que não funciona aproximadamente 53 minutos em um ano e este período pode ser crítico para algumas aplicações.

A Tabela 3.2 apresentada por Tanenbaum [54] [53], compara os requisitos de QoS para uma boa experiência em diversas aplicações.

Tabela 3.2: Comparação dos requisitos de QoS de diversas aplicações [54]

Aplicação	Confiabilidade	Atraso	<i>Jitter</i>	Largura de Banda
E-mail	Alta	Baixa	Baixa	Baixa
Transferência de arquivos	Alta	Baixa	Baixa	Média
Acesso WEB	Alta	Média	Baixa	Média
Login Remoto	Alta	Média	Média	Baixa
Áudio por Demanda	Baixa	Baixa	Alta	Média
Vídeo por Demanda	Baixa	Baixa	Alta	Alta
VoIP	Baixa	Alta	Alta	Baixa
Videoconferência	Baixa	Alta	Alta	Alta

As aplicações que transmitem informações como terminal remoto com TELNET ou transferência de algum arquivo compactado via Protocolo de Transferencia de Arquivos (FTP - *File Transfer Protocol*) são muito sensíveis a erros, mas pouco sensíveis a atraso. Apesar de longos atrasos serem desconfortáveis a aplicação consegue funcionar corretamente. Apesar disto, estas aplicações precisam de mecanismos que detectem ou corrijam erros, pois um único bit errado pode se propagar e o receptor receberá dados ilegíveis.

AS mídias de áudio e vídeo são relativamente resistentes a perdas, uma vez que alguns bits errados podem modificar apenas alguns *pixels* na tela e este erro pode passar despercebido pelo olho humano. Apesar disto, as aplicações de vídeo e áudio são sensíveis a atrasos e *jitter* para funcionar corretamente. Além disto, as aplicações de vídeo requerem alta largura de banda. Enquanto uma transmissão via FTP poderia tolerar vários segundos de atraso, uma vídeo conferencia não suportaria uma centena de milissegundos.

Técnicas de qualidade de serviço

Oferecer qualidade na transmissão do fluxo de uma aplicação pela Internet é extremamente difícil uma vez que o roteamento utiliza a política do melhor esforço (*Best-effort*) e não realiza tratamento diferenciado dos pacotes de aplicações mais exigentes. Dessa forma diversas técnicas surgiram para tentar prover QoS às redes IP. De acordo com [17], as técnicas mais comuns para garantir QoS são: políticas de filas, *traffic shapping*, reservas de recursos, controle de admissão e superdimensionamento.

Nas políticas de filas os roteadores tratam cada tipo de pacote de forma apropriada dependendo de algum sistema de classificação. Existem basicamente três tipos de filas:

- **FIFO (*First-in-First-Out*)**: Antes de serem encaminhados, os pacotes são armazenados em *buffer* por ordem de chegada, assim, o primeiro pacotes que chegar será o primeiro a sair, independente do seu conteúdo.
- **Filas de Prioridade**: Neste tipo, os pacotes são classificados de acordo com sua prioridade antes de entrar na rede. Ao chegar a um roteador os pacotes são encaminhados para *buffers* diferentes. Os pacotes de menor prioridade só poderão ser encaminhados depois que todos os pacotes de maior prioridade forem encaminhados. Neste modelo, caso haja muitos pacotes de alta prioridade, os de menor prioridade sofrerão grandes atrasos.
- **Filas de Encaminhamento Ponderadas**: Para lidar com a justiça na rede e os pacotes de menor prioridade também sejam encaminhados, a cada *buffer* de fila em um roteador é atribuído um peso. Cada fila terá uma quantidade de pacotes encaminhados de acordo com o peso da fila, sem deixar de encaminhar os de menor peso.

Em técnicas do tipo *traffic shapping*, o tráfego é modelado de forma a controlar sua taxa de transmissão. as duas das técnicas mais comuns são o Balde Furado (*Leaky Bucket*) e o Balde de Fichas (*Token Bucket*), conforme descrito a seguir:

Leaky Bucket: a técnica *Leaky Bucket* consiste em modelar o tráfego de forma que seja transmitido de forma suave. Assim, em vez de transmitir dois segundos de um

tráfego a 12 Mbps, por exemplo, esta técnica permitirá a transmissão sob taxa de 3 Mbps por 8 segundos

Token Bucket: a técnica chamada de *Token Bucket* atribui um *token* ao tráfego após um certo intervalo de tempo. O roteador encaminhará o tráfego a taxa máxima enquanto houver *tokens*. Caso não haja dados a serem transmitidos, os *tokens* são armazenados até um limite. Assim, um pico de tráfego muito intenso pode ser distribuído em um tráfego mais uniforme. Diferente do *Leaky Bucket*, um tráfego ocioso pode acumular *tokens* para quando precisar.

Duas técnicas muito utilizadas em tecnologias com circuitos virtuais como o ATM (*Asynchronous Transfer Mode* - Modo de Transferência Assíncrono) são a reserva de recursos e o controle de admissão, como descrito a seguir:

Reserva de Recursos: Na reserva de recursos os roteadores ao longo do caminho reservam antecipadamente recursos de rede, como largura de banda, *buffers* e processamento necessários para a transmissão. Caso não haja recursos necessários para realizar a reserva, pode ocorrer o controle de admissão.

Controle de Admissão No controle de admissão, caso certos parâmetros do tráfego não estejam dentro dos esperados pela rede a entrada do fluxo na rede pode ser negada. Por exemplo, um certo tipo de fluxo precisa reservar uma largura de banda mínima na rede. Caso a rede não disponha desse recurso e não haja forma de renegociá-lo, o tráfego pode ser rejeitado na rede. Este controle pode ser utilizado para negar a transmissão de um tráfego não permitido pelas políticas locais ou que possam comprometer o funcionamento da rede

Por fim, o superdimensionamento é considerado uma solução empírica e simples para se oferecer qualidade de serviço. A ideia é fornecer largura de banda excedente, capacidade extra de processamento dos roteadores e espaço de *buffer* ilimitado onde os pacotes serão transmitidos com enorme facilidade. Porém, o grande problema desta solução que frequentemente é muito cara e pouco determinística [53].

QoS sob IP

Em meados dos anos 90 o IETF desenvolveu dois modelos de arquiteturas utilizando diversas técnicas de QoS para tentar garantir a qualidade de serviço na Internet.

O primeiro modelo foi a arquitetura Serviços Integrados (IntServ - *Integrated Service*) desenvolvida entre 1995 e 1997. Ele utiliza o protocolo RSVP [3] que realiza a reserva de recursos em redes de multidifusão. O receptor escolhe a origem e os nós da rede podem

reproduzir o fluxo para outros nós de forma semelhante ao *multicast*. Porém, apesar de apresentar bom desempenho, esta tarefa é bastante complexa e necessita que o estado dos fluxos seja armazenado ao longo do caminho.

Para aumentar a escalabilidade do serviço, a arquitetura de Serviços Diferenciados (DiffServ - *Differentiated Service*) foi desenvolvido. Ele é um modelo baseado em classe, ou seja, uma classe é atribuída a cada pacote que se deseja transmitir e, baseado nesta classe, a rede o encaminha para o próximo salto. Cada pacote é marcado no campo DSCP (*Differentiated Services Code Point*) do cabeçalho IP. As regras de como os pacotes são tratados são chamadas de Comportamento Por Salto (PHB - *Per-Hop Behavior*) e, em ordem de prioridade, são definidos como Padrão (DE - *Default*), Encaminhamento Assegurado (AF - *Assured Forwarding*) e Encaminhamento Acelerado (EF - *Expedited Forwarding*). Esta técnica permite maior escalabilidade que o *IntServ*, mas não gera bons resultados se a maior parte do tráfego for do tipo EF ou AF.

3.4 FTV

A *Free viewpoint TeleVision* (FTV) é uma nova aplicação de vídeo ao vivo que permite o espectador ver um cenário tridimensional escolhendo o ângulo de visão livremente. A FTV faz parte de um conjunto de tecnologias de imersão, trazendo ambientes distantes para próximo do usuário. [55]

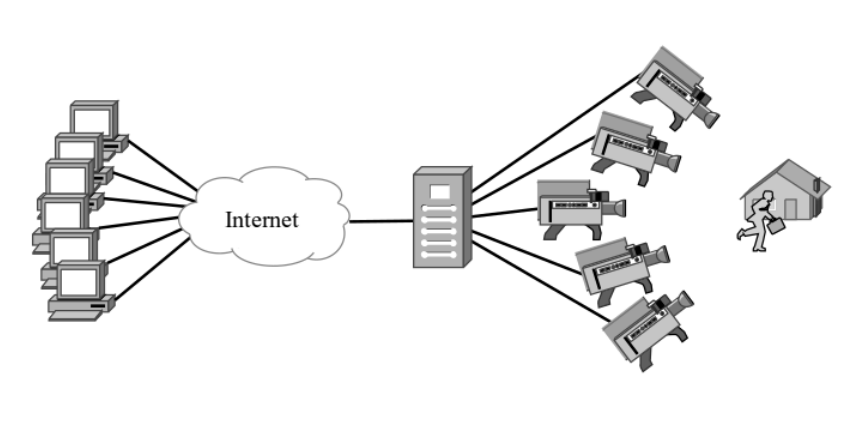


Figura 3.5: Aplicação FTV

Para esta aplicação, um arranjo de câmeras capturam uma cena que é processada, codificada e transmitida pela rede ou armazenada, como ilustrado na Figura 3.5. O receptor recebe os vídeos codificados e cria uma imagem sintética do ângulo desejado que pode ser exibida em vários tipos de dispositivos.

A arquitetura da FTV da Figura 3.6 e compreende os seguintes passos:

- **Captura:** O fluxo FTV é capturado por diversas câmeras em diferentes arranjos, dependendo do grau de liberdade que foi projetado. Para aplicações que oferecem movimentação em apenas uma dimensão, pode ser necessário cerca de 300 câmeras, enquanto uma liberdade bidimensional pode requerer uma matriz de milhares [56].
- **Correção:** Após a captura o fluxo deve ser tratado para corrigir diferenças de iluminação, cor, paralaxe, etc, antes que possam ser enviados para a próxima etapa.
- **Estimativa de Profundidade:** Nesta fase, com base na diferença dos ângulos de cada câmeras adjacentes é calculada a profundidade dos objetos na cena, gerando um mapa de profundidade que será transmitido ou armazenado junto com os fluxos.
- **Codificação:** Antes que possam ser transmitidos ou armazenados, todos os fluxos com os mapas de profundidade são codificados usando um padrão de codificação como H.264 ou H.265.
- **Transmissão ou armazenamento:** Um dos objetivos da FTV é a transmissão de vídeo 3D ao vivo, mas também pode ser utilizada para exibir vídeo armazenado. O fluxo de vídeo pode ser armazenado em qualquer tipo de mídia que suporte armazená-la ou transmitido pela internet. O tipo de mídia ou protocolo ainda não foi especificado, tendo em vista que a aplicação está em fase de padronização.
- **Decodificação:** Ao chegar ao destino é necessário decodificar o vídeo e o mapa de profundidade para que o vídeo possa ser processado.
- **Interpolação:** A fase de interpolação consiste na junção da visão de duas câmeras de vídeo mais o mapa de profundidade afim de gerar a imagem do novo angulo. Este processo não exige tanto poder de processamento quanto a extração do mapa de profundidade, mas pode impedir que alguns dispositivos com baixo poder de processamento, como *tablets* e celulares não exibam o vídeo corretamente
- **Interface:** A FTV está sendo desenvolvida para suportar dezenas de dispositivos diferentes. Ela poderá ser exibida em dispositivos tradicionais como celulares, notebooks, TVs 3D e até novos tipos de dispositivos 3Ds como *Seelinder* [60] ou algum

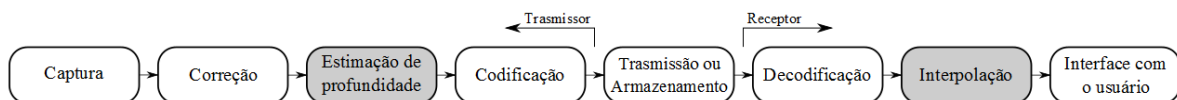


Figura 3.6: Arquitetura do FTV [55]

dispositivo estereoscópico ou autoestereoscópico. A Figura 3.7 mostra diversos dispositivos que podem possibilitar a exibição do fluxo FTV.



Figura 3.7: Exemplo de dispositivos usados com FTV [59], [60], [22]. [43].

O MPEG concentra-se na elaboração do padrão para o FTV desde 2001. A primeira etapa foi a padronização da codificação para *multiview vídeo*, a MVC (*Multiview Video Code*). A MVC é uma extensão para o padrão H.264 que permite a codificação e transmissão de múltiplos fluxos de vídeo em um único fluxo. Seus sucessores são o MV-HEVC (*Multiview Video - High Efficiency Video Coding*) e 3D-HEVC (*3D - High Efficiency Video Coding*) que compõem o padrão H.265. A segunda etapa da padronização compreende o 3DV (*3D Vídeo*) que permite a adaptação do fluxo e da exibição do vídeo em dispositivos finais.

Em agosto de 2013 o MPEG começou uma nova fase da padronização focando nas aplicações *Super Multiview* que permitem apenas o movimento horizontal e *Free Navigation* que permitem movimentos livre nos três eixos) [56] [55]. Diversas aplicações já estão previstas para a FTV como por exemplos Estádios Virtuais, Parque temático, jogos eletrônicos, monitoramento e segurança, inspeção industrial, astronomia, medicina, geografia, sala de aula virtual, entre outras [56].

3.5 Tipos de tráfego

O conhecimento prévio das características do tráfego que entra em uma rede é fundamental para garantir QoS em uma rede. A modelagem correta do tráfego ajuda a prever

o comportamento real do tráfego, permitindo uma melhor utilização de recursos da rede.

A caracterização do tráfego tem como objetivo extrair informações que possam representar o tráfego analisado. Com estas características definidas é possível construir um modelo matemático que possa representar o comportamento estudado. Assim, com o modelo matemático elaborado é possível realizar simulações em cenários hipotéticos que permitem analisar as fronteiras do comportamento de um sistema real.

Textos, imagens, vídeos e outros tipos de informação digital compõem o tráfego Internet. Cada uma possui características próprias, mas de forma geral possuem três tipos de classe:

Constante Bit Rate (CBR): O tráfego com taxa de bits constantes a taxa de bits do tráfego não varia com tempo, sendo sempre o mesmo. Por exemplo, o sistema telefônico tradicional a cabo digitaliza a voz usando PCM a uma taxa constante de 64 kbps. Além disso, o fluxo de vídeo e áudio são capturados e reproduzidos a taxa constante. Apenas após a compressão que o fluxo se torna variável.

Variable Bit Rate (VBR): Neste tipo de tráfego a taxa de bits é variável, ou seja, o tráfego varia com o tempo, como mostra a Figura 3.8. Este tipo de tráfego apresenta pontos de máximos e mínimos de atividade distribuídos em torno de uma média e pode ser caracterizado com diversos parâmetros estatísticos. O tráfego de vídeo e o áudio, quando compactado e transmitido para a Internet, são do tipo VBR.

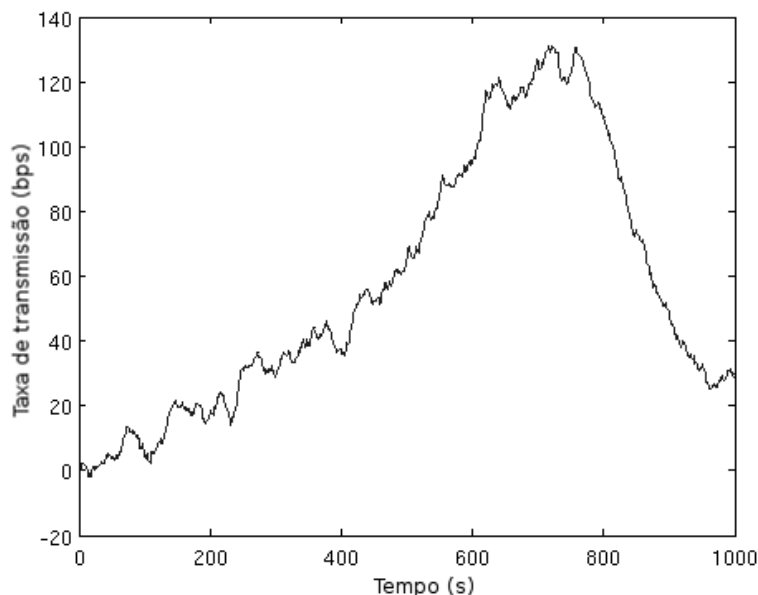


Figura 3.8: Tráfego VBR

Burst ou Rajada: Rajada é um tipo de tráfego que se caracteriza por picos de atividade alternados com menor atividade. A Figura 3.9 mostra um exemplo de rajada onde o tráfego recebe dois bruscos aumento de atividade antes de retornar ao comportamento normal. As aplicações como Web e transferências FTP geram este tipo de tráfego. Tanto a duração da rajada quanto o intervalo entre elas são variáveis aleatórias.

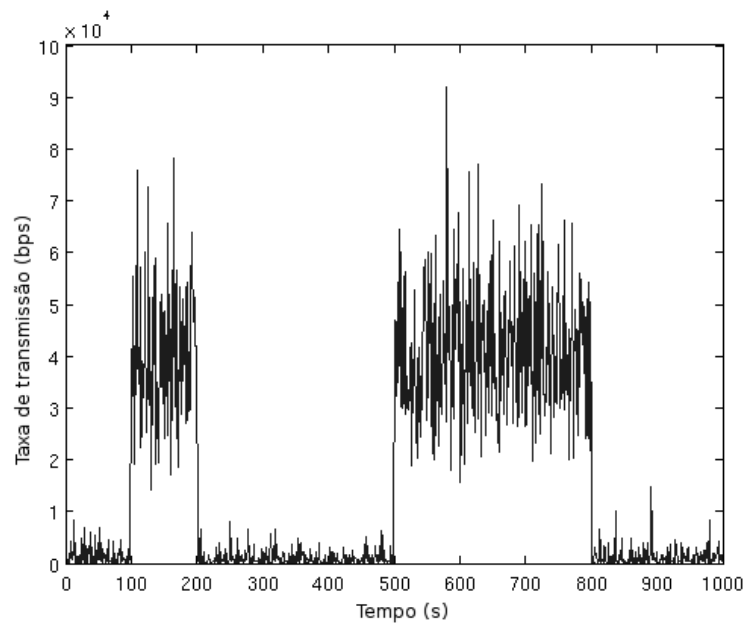


Figura 3.9: Tráfego em Rajadas

Destes três tipos de tráfego o VBR e o de rajadas não são determinísticos e podem ser modelados usando algum processo estocástico. Por exemplo, o tráfego telefônico consegue ser bem representado através do processo de Poisson usando cadeias markovianas.

O processo de Poisson foi utilizado com sucesso por décadas, porém, apesar das redes de computadores terem quase 50 anos, apenas nos últimos 20 anos foi provado que o tráfego LAN (*Local Area Network*), WAN (*Wide Area Network*) e multimídia apresentam características de longa dependência, onde os picos de tráfego podem durar mais do que o previsto, aumentando a chance de ocorrer um estouro de *buffer* e os pacotes excedentes serem descartados.

Os processos estocásticos podem ser divididos em curta dependência, onde sob baixa autocorrelação os valores de sua distribuição não dependem um do outro, e longa dependência, também chamado de processos com memória, onde um incremento do tráfego pode indicar uma tendência dele continuar crescendo e vice-versa. Os processos de Poisson são considerados de curta duração pois seus incrementos/decrementos são independentes.

Enquanto os processos de longa dependência podem apresentar um comportamento autossimilar, cuja autocorrelação não varia com a escala.

3.5.1 O Tráfego de Poisson

O matemático dinamarquês Agner Krarup Erlang que trabalhava na central telefônica de Copenhague realizou os primeiros estudos teóricos sobre o tráfego telefônico. Em 1909 descobre que as chamadas telefônicas podem ser aproximadas a um modelo de Poisson. Como consequência, a taxa de bloqueios do sistema B é dada pela equação 3.1.

$$B(d, L) = \frac{d^L / L!}{\sum_{k=0}^L (d^k / k!)} \quad (3.1)$$

No modelo de Poisson, o tráfego alterna picos de atividade com tráfegos menores em torno de uma média. Como resultado, o uso de filas geraria um rápido escoamento em alguns nós e em consequência, atrasos previsíveis. Ao se juntar vários tráfegos de Poisson o resultado é um tráfego mais suavizado, assemelhando-se ao tráfego constante, facilitando seu estudo.

O tráfego telefônico geralmente é modelado utilizando o modelo de Poisson, onde picos de tráfego alternam em torno de uma média. Mas esse modelo não consegue representar uma rede de datagramas com precisão dado que vários estudos mostram que o tráfego de rede não possui o comportamento de Poisson e sim de dependência de longa duração (LRD - *Long Range Dependence*) [7] [14] [33] [44].

3.5.2 O Tráfego Autossimilar

A partir da década de 90, vários estudos já demonstraram que diversos tipos de tráfego em redes apresentam características de autossimilaridade. Leland et al. encontrou que o tráfego LAN é autossimilar [33], Beran et al. verificou que o tráfego VBR de vídeo é autossimilar [7] e Fendman também encontrou autossimilaridade no tráfego WAN [14]. Ou seja, ele está presente em toda a Internet e conclui-se que os recursos previstos pelos modelos tradicionais são muito limitados, influenciando o desempenho da rede e afetando diretamente as métricas de QoS.

No tráfego autossimilar os picos de atividade podem ter duração maior do que o previsto nos modelos de Poisson gerando congestionamentos e perda de pacotes. Essa persistência é chamada de LRD e é uma característica encontrada em fractais [33].

O tráfego autossimilar apresenta duas características muito importantes para sua caracterização: a invariância à escala e persistência temporal, onde os picos duram mais do que os do processo de Poisson. Essa invariância a escala pode ser medida pelo parâmetro

de Hurst, uma estimativa da dimensão fractal do processo estocástico que indica seu grau de autossimilaridade.

Seja um processo $Y(t), t \in \mathbb{R}$. Diz-se que $Y(t)$ é autossimilar com parâmetro de Hurst $0,5 \leq H \leq 1$ se $Y(t) \equiv a^{-H}Y(at), a > 0, t \leq 0$ satisfaz as equações 3.2, 3.3 e 3.4, correspondentes a Média, variância e autocorrelação, respectivamente.

$$E[Y(t)] = \frac{E[Y(at)]}{a^H} \quad (3.2)$$

$$var[Y(t)] = \frac{var[Y(at)]}{a^{2H}} \quad (3.3)$$

$$R_y(t_1, t_2) = \frac{R_y(at_1, at_2)}{a^{2H}} \quad (3.4)$$

O processo autossimilar gera rajadas em todas as escalas. Graficamente falando, quando se altera a escala de visualização, o tráfego continua apresentando as mesmas características, enquanto no processo de Poisson esta propriedade não existe.

O Movimento Browniano Fracionário (fBm) é um exemplo de processo estocástico autossimilar, proposto por Mandelbrot e Van Ness [51], definido como $A(t) = \beta t^H$, onde β é uma variável aleatória com distribuição normal, média zero, variância igual a um e H o índice de autossimilaridade. Sua variância é dado pela equação 3.6 e sua equação de correlação é dado pela equação 3.7.

$$E[A(at)] = a^H E[A(t)] = 0 \quad (3.5)$$

$$var[A(at)] = a^{2H} var[A(t)] = a^{2H} t^{2H} \quad (3.6)$$

$$R_{B_H}(at, as) = \frac{1}{2} [(at)^{2H} + (as)^{2H} - |at - as|^{2H}] = a^{2H} R_{B_H}(t, s) \quad (3.7)$$

Outro processo muito utilizado para modelagem de tráfego autossimilar é chamado de Ruído Gaussiano Fracionário (fGn) que representa o incremento do processo fBm definido como $X(t) = A(t) - A(s)$. Este processo possui dependência de longo prazo e variância com decaimento lento quando $0,5 \leq H \leq 1,0$. Sua função de autocorrelação é dada pela equação 3.8.

$$R(k) = \frac{1}{2} (|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}) \quad (3.8)$$

Por exemplo, a série da Figura 3.10 foi gerado utilizando fBm com parâmetro de Hurst igual a 0,8 e 10.000 amostras. Existem diversos métodos para estimar o parâmetro de

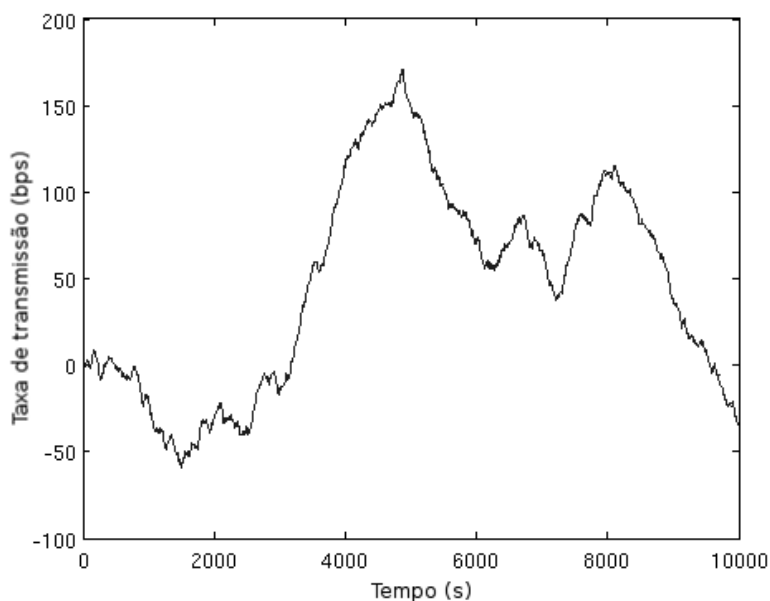


Figura 3.10: Exemplo de série temporal com LRD e parâmetro $H = 0,8$.

Hurst de uma amostra. Dentre os mais conhecidos estão os métodos Variance-Time-Plot (VTP), Higuchi, Abry-Veitch, Neurocomputação e Kettani-Gubner [30]. Os métodos de Abry-Veitch e Kettani-Gubner são os mais populares por possuírem menor complexidade a alta precisão, como descrito a seguir.

- **Abry-Veitch** Este método baseado na transformada *wavelet* foi desenvolvido em meados de 1998 em [1] por Abry e Veitch. Este método bastante robusto foi implementado em ferramentas como MATLAB® é muito utilizado para análise de grande quantidade de dados, porém o método de estimativa requer alto poder de processamento, dependendo da quantidade de dados.
- **Kettani-Gubner:** Este método apresentado em [30], considera que o tráfego LAN é um processo Gaussiano autossimilar exatamente de segunda ordem para deduzir a expressão da autocorrelação 3.9 e o parâmetro de Hurst 3.10. Com base nestas aproximações é possível implementar o estimador com bastante simplicidade. De acordo com [30], este método é mais rápido que o método *wavelet* e possui grande nível de precisão.

$$\hat{\rho}_n(k) = \frac{\hat{\gamma}_n(k)}{\hat{\sigma}_n^2} \quad (3.9)$$

$$\hat{H}_n = \frac{1}{2}[1 + \log_2(1 + \hat{\rho}_n(1))] \quad (3.10)$$

Norros em [39] propõe um modelo que utiliza o fBm normalizado para a descrição sistemas de armazenamento com filas onde $A(t) = at + \sigma Z(t)$ representa o número de bits que chegam em um intervalo $(0, t]$. Este processo possui a média, variância e covariância expressos pelas equações 3.11, 3.12 e 3.13 respectivamente.

$$E[A(t)] = a \quad (3.11)$$

$$var[A(t)] = t^{2H} \quad (3.12)$$

$$C_A(t, s) = \frac{1}{2}[s^{2H} + t^{2H} - (s - t)^{2H}], t > s > 0 \quad (3.13)$$

A média, o desvio padrão e o H do modelo de Norros capturam corretamente as características de um tráfego real, podendo ser utilizado para representação do tráfego agregado, resultado da multiplexação de diversas fontes em um nó, mesmo que o tráfego não seja exatamente autossimilar, mas assintoticamente autossimilar.

Norros então define o processo envelope para um fBm como a equação 3.14, no qual $\kappa = \sqrt{-2 \ln \varepsilon}$ determina a probabilidade de que $A(t)$ ultrapasse $\hat{A}(t)$ no instante t .

$$\hat{A}(t) = at + \kappa \sqrt{\sigma^2 t} = at + \kappa \sigma t^{\frac{1}{2}} \quad (3.14)$$

A lei de Hust estabelece que a variância de um incremento deste processo deve ser dada por $var[A_H(t = s) - A_H(t)] = \sigma^2 s^{2H}$, com $H \in [1/2, 1]$, desta forma o processo envoltória é definido pela equação 3.15. Pode-se observar que o processo envelope fBm é um caso particular quando $H = 1/2$.

$$\hat{A}(t) = at + \kappa \sigma t^H \quad (3.15)$$

Agregação do tráfego autossimilar

O tráfego de redes é a agregação de milhares de fluxos de diversas aplicações com características distintas. Conhecendo as características de cada tipo de fluxo é possível estimar as características do tráfego agregado.

O tráfego autossimilar pode ser aproximado por um processo fBm. Este tipo de tráfego é caracterizado por três parâmetros: o parâmetro H , a variância σ^2 e a taxa média \bar{a} .

$$\sigma_{12}^2 = \sigma_1^2 + \sigma_2^2 + 2cov(X, Y) \quad (3.16)$$

A variância possui a propriedade de que a variância da soma de duas variáveis aleatórias é igual a soma das variâncias de cada variável, mais duas vezes a covariância

destas variáveis, como mostra a equação 3.16. Porém, quando as variáveis aleatórias são independentes, a covariância entre elas é igual a zero e a equação se resume a equação

$$\sigma_g^2 = \sum_s \sigma_s^2 \quad (3.17)$$

Segundo Solis Barreto em [51], Orenstein P. et al. propôs também um método bastante preciso para obtenção do parametro H agregado como é mostrado na Equação 3.18. Solis também comparou esta equação com outra proposta também por Orenstein, porém esta apresentou melhores resultados

$$H_g = \frac{\sum_s H_s \sigma_s^2}{\sum_s \sigma_s^2} \quad (3.18)$$

Uma característica que se extrai destas equações de variância agregada e H agregado é a influência da variância no parâmetro H agregado. Um tráfego com alto H e alta variância aumenta o H total da rede, quanto um H baixo e uma variância baixa diminui o H_g .

3.6 Caracterização do tráfego de vídeo

Os primeiros estudos sobre a natureza autossimilar do vídeo foram realizados por Mark Garrett em meados de 1993, onde ele digitalizou o filme *Star Wars* e o codificou usando M-JPEG (que posteriormente se tornou o padrão MPEG-1) [44]. Em seu trabalho, Garrett utilizou a *rescaled range analysis* (ou *R/S statistic*) além do espectro de potência de Fourier (também chamado de Periodograma), para estimar o parâmetro de Hurst.

Garret e Willinger foram os primeiros a estudar a caracterização e modelagem de tráfego de vídeo VBR em [20] como autossimilar, encontrando $H = 0,8$. No vídeo analisado o coeficiente de autocorrelação do tráfego decresce muito lentamente, indicando características de LRD.

Reljim em[44] realizou a caracterização fractal e multifractal do fluxo de vídeo H.264 do filme Tropas Estelares, enquanto Malindi em [36] apresentou a modelagem do fluxo H.264 considerando sua autossimilaridade. Devido a recente padronização do H.265, até a presente data não foi reportado nenhum trabalho de caracterização considerando a natureza fractal do vídeo.

A Figura 3.11 mostra a invariância a escala presente no tráfego de vídeo codificado com o padrão H.264 [44] com 25 quadros por segundo. Esta invariância a escala presente em processo autossimilares é claramente visível nesta imagem. Ainda neste trabalho é apresentada a tabela 3.3 mostrando diferentes valores para H dado diferentes quantizações

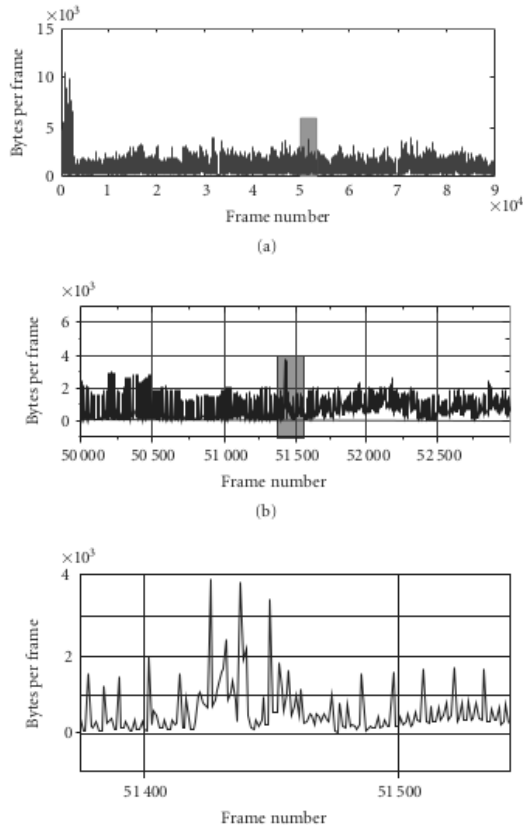


Figura 3.11: Invariância a escala do filme Tropas estelares [44]

e diferentes quantidades de amostras para o mesmo vídeo. Estes valores foram calculados usando periodograma e apresentam valores próximos a 0.8, independente da quantidade da amostra e independente da compressão utilizada no vídeo. Como pode-se observar, 2000 quadros com parâmetro de quantização igual a 5, que representam 80 segundos de vídeo, possuem o valor de H próximo de 90.000 quadros com parâmetro de quantização igual a 25.

Assim, o tráfego de vídeo autossimilar pode ser modelado com o Processo Envoltória Fractal (FEP - *Fractal Envelop Process*) em que representa o limite superior de um sistema de filas com chegadas autossimilar [16] [39]. Desta forma o tamanho da fila em cada nó é dado pela equação 3.19, onde \bar{a} representa o tráfego médio, no tempo t onde $\kappa = \sqrt{-2\ln\varepsilon}$ e ε é a probabilidade do tráfego ultrapassar o *buffer*, podendo significar a probabilidade de perda de pacotes e H é o coeficiente de autossimilaridade. φ representa o desvio padrão e C_v .

$$\hat{q} = \bar{a}t + \kappa\sigma t^H - ct \quad (3.19)$$

O tamanho máximo que a fila pode alcançar é calculado pela derivada da fila em função do tempo igual a zero (equação 3.20). Assim, o tempo em que a fila alcança o pico

Tabela 3.3: Parâmetro H para diferentes quantizações e quantidades diferente de amostras [44]

Video trace length: number of frames	Hurst indices for different quantization scales, q_p , or compression ratio, CR		
	$q_p = 5$ CR = 3.5	$q_p = 15$ CR = 15.9	$q_p = 25$ CR = 79.0
500	0.7802	0.94298	0.75985
800	0.78637	0.78477	0.80582
1000	0.79554	0.79481	0.80701
3000	0.80417	0.8041	0.79143
5000	0.80282	0.80181	0.69483
10 000	0.81522	0.82934	0.80202
20 000	0.81245	0.83263	0.8018
30 000	0.81538	0.84327	0.80451
50 000	0.81421	0.81227	0.81262
70 000	0.81219	0.84631	0.81308
90 000	0.8953	0.85598	0.80762

é dado pela equação 3.21, onde pode-se extrair a equação 3.22.

$$\frac{d\hat{q}}{dt} = 0 \quad (3.20)$$

$$t_{max} = \left(\frac{\kappa\sigma H}{c - \bar{a}} \right)^{\frac{1}{1-H}} \quad (3.21)$$

$$q_{max} = (c - \bar{a})^{\frac{H}{H-1}} (\kappa\sigma)^{\frac{1}{1-H}} (H)^{\frac{H}{1-H}} (1 - H) \quad (3.22)$$

$$d = \frac{q_{max}}{c} \quad (3.23)$$

Desta forma o atraso máximo para $1 - \varepsilon$ do tráfego é dado pela equação 3.23. O gráfico 3.12 mostra o crescimento do atraso em função da utilização para um coeficiente de variação igual a 1,5 e ε igual a 0,001. Pode-se observar que mesmo com utilização de apenas 15% do enlace o atraso pode atingir quase 600 milissegundos em apenas um nó, dependendo do coeficiente de autossimilaridade. E para utilizações maiores o crescimento pode chegar a milhares de segundos, como mostra a Figura 3.13, onde a reta de maior inclinação representa o atraso quando o $H = 0,9$ e a de menor inclinação o $H = 0,5$.

Por outro lado, desde que o processo fBm não exceda $\hat{A}_H(t)$ com probabilidade $1 - \varepsilon$, o tamanho máximo da fila será limitado por q_{max} com a mesma probabilidade. Então

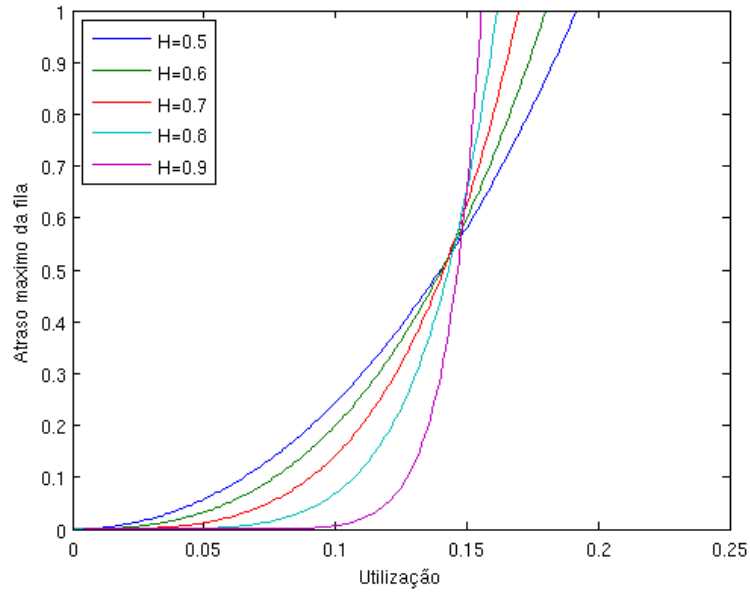


Figura 3.12: Atraso máximo da fila em função da utilização para Processo envelope

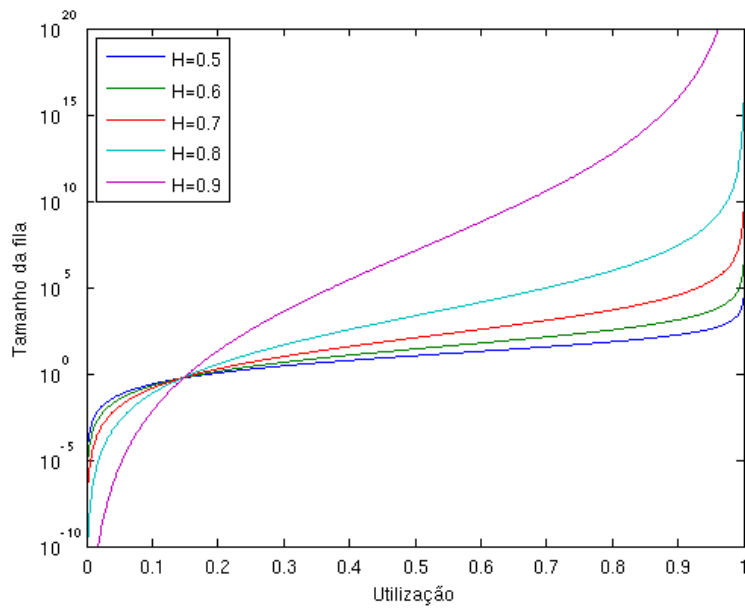


Figura 3.13: Crescimento do atraso até 99% de utilização

existe uma capacidade \hat{c} onde q_{max} será o tamanho máximo do buffer com tamanho K . Ou seja, existe a probabilidade de ε da fila começar a descartar pacotes se a capacidade for \hat{c} , onde \hat{c} é dado pela equação 3.24. Este resultado é utilizado para estimar a largura de banda necessária para a transmissão do tráfego autossimilar [16].

$$\hat{c} = \bar{a} + K^{\frac{H-1}{H}} (\kappa\sigma)^{\frac{1}{H}} H(1-H)^{\frac{1-H}{H}} \quad (3.24)$$

Na prática a maior parte do tráfego se encontra abaixo desta curva, mas, como o comportamento do tráfego é autossimilar, existe a probabilidade de ε do tráfego ultrapassar este valor.

3.7 Caracterização do tráfego FTV

O tráfego de vídeo FTV é composto por centenas de fluxos de vídeo mais o mapa de profundidade. Como ainda não foi definido um padrão para a codificação e transmissão dos fluxos FTV, supõe-se que cada fluxo será codificado e transmitido separadamente, assim como o mapa de profundidade.

Entende-se que o padrão de vídeo 3D possui três formas de codificação: *Multiview Video* que tenta codificar os quadros de forma que explore a redundância dos pixels entre quadro, *Frame Sequential* que mescla duas sequências de quadros antes de aplicar alguma codificação, como se fosse um vídeo normal com o dobro taxa de quadros e *side-by-side* que dobra a resolução horizontal do quadro e combina os vídeos de tal forma que gere uma simples sequência de quadros para que possa ser codificado como se fosse um vídeo com única visão [50]. Porém, nenhuma destas técnicas pode funcionar de forma eficiente para a FTV pois não são escaláveis o suficiente para suportar centenas ou milhares de fluxos.

Desta forma, para o vídeo ao vivo, cada fluxo de cada câmera pode ser codificado individualmente e todos os seus quadros de um determinado período devem ser transmitidos dentro de um intervalo igual a $1/f$, onde f é a taxa de quadros. Por exemplo, se o vídeo possui 24 quadros por segundos, o tamanho médio do quadro é 20 Kbytes e é composto de 100 câmeras, cada quadro deve possuir um atraso de transmissão máximo igual a 416,7 μ s, o que exigiria uma conexão com taxa média de 393.216,0 Mbps.

Este tráfego deve exigir uma enorme quantidade de largura de banda, é autossimilar e possui alta variância. Esses parâmetros dependerão da taxa de compressão, da codificação e da resolução do vídeo.

A taxa de compressão é definida pelo parâmetro de quantização (Q). Quanto maior o Q , mais bits são gastos para armazenar informação, conseqüentemente maior a qualidade e maior o tamanho do quadro.

Foram analisados diversos traces do vídeo *Tears of Steel* disponíveis em [50] com resolução 4K (4096 x 1744) para os padrões H.265 e H.264. A tabela 3.4 mostra os valores para a o Parâmetro de quantização, tamanho médio dos quadros, coeficiente de variação, taxa média de transmissão, taxa de transmissão de pico e parâmetro H .

A Codificação H.265 possui uma compressão maior do que H.265, porém, uma variância maior. Na média o coeficiente de variação esperado é aproximadamente 1,5, indicando

Tabela 3.4: Características do vídeo

Codificação	Q	C_v	Taxa média	Taxa de pico	H
H.265	20	0,956	53,964 Mbps	391,158 Mbps	0,85
H.265	25	1,336	14,593 Mbps	259,385 Mbps	0,86
H.265	30	1,797	5,676 Mbps	156,805 Mbps	0,87
H.265	35	1,952	2,536 Mbps	82,793 Mbps	0,87
H.265	40	1,998	1,177 Mbps	37,012 Mbps	0,87
H.264	20	0,816	77,935 Mbps	464,663 Mbps	0,86
H.264	25	1,037	27,133 Mbps	324,992 Mbps	0,86
H.264	30	1,461	9,204 Mbps	205,569 Mbps	0,86
H.264	35	1,546	4,506 Mbps	117,260 Mbps	0,87
H.264	40	1,417	2,392 Mbps	60,390 Mbps	0,88

que o desvio padrão é 50% maior do que a média. Como descrito anteriormente, esta variação afeta fortemente o parâmetro H do tráfego agregado, tornando o tráfego da rede mais autossimilar. Outra característica observada foi que o tráfego do vídeo em 4K de pode atingir picos de quase 500 Mbps, o que o torna muito difícil de oferecer qualidade sem superestimar os recursos da rede.

Assim, cada fluxo de vídeo FTV é tratado pela rede como um fluxo de vídeo individual com o mesmo par origem-destino que pode ser roteado por *multicast* ou *unicast*. Em protocolos de roteamento como o OSPF não há a identificação dos fluxos sendo todo o roteamento feito apenas baseado nos endereços de origem e destino e no custo dos enlaces. Conseqüentemente, os pacotes de todos os fluxos FTV podem seguir todos por um único menor caminho ou podem ser divididos por vários menores caminhos por uma técnica citada no capítulo 2 chamada ECMP.

Não realizar a divisão dos fluxos implica em uma enorme quantidade de tráfego de vídeo que pode atingir dezenas de Gigabits passando pelos enlaces em um único caminho e devido principalmente a natureza autossimilar do tráfego de vídeo VBR, poderia rapidamente preencher o *buffer* dos roteadores gerando grandes congestionamentos, perdas de pacotes e prejudicando todas as outras aplicações da rede.

Utilizar o roteamento com ECMP acaba por se tornar a única opção viável para este tipo de tráfego realizando o balanceamento de carga ao longo de vários caminhos. Infelizmente este procedimento pode criar longos *Jitters* e inversão na ordem dos pacotes que devem ser eliminados com o uso de *buffers* no lado do receptor. De acordo com [54], para aplicações ao vivo atraso de 15 segundos para se iniciar uma aplicação de vídeo ao vivo é completamente aceitável.

3.8 Resumo do capítulo

Este capítulo apresentou o vídeo digital, formas de transmiti-lo pela internet, as principais características de seu tráfego e uma nova aplicação chamada FTV que exige muita largura de banda e uma arquitetura especial. Foi visto as principais métricas relacionadas a transmissão digital e diversas técnicas usadas para garantir qualidade. No próximo capítulo será apresentada uma proposta para engenharia de tráfego e otimização de roteamento para o vídeo FTV, assim como os resultados da simulação desta proposta.

Capítulo 4

Metodologia de Otimização de Roteamento e Resultados Experimentais

Neste capítulo é detalhada a proposta de metodologia de engenharia de tráfego para otimização do roteamento em função dos requerimentos de QoS do tráfego de vídeo. Posteriormente, a proposta é validada através de um conjunto de simulações e os resultados serão comparados e analisados.

4.1 Proposta

4.1.1 Objetivos gerais

O objetivo geral da metodologia proposta é realizar a otimização de desempenho do roteamento intradomínio para fluxos de vídeo autossimilar em arquiteturas cliente-servidor. Para tal, foram definidos um conjunto de objetivos específicos, detalhados na seção seguinte.

4.1.2 Objetivos específicos

1. Caracterizar matematicamente os fluxos do tráfego de vídeo de alta resolução em uma rede IP com base no modelo autossimilar;
2. Propor uma técnica de engenharia de tráfego baseada em protocolos tradicionais de roteamento intradomínio que tenha o parâmetro de Hurst e a utilização do enlace como critérios para escolha das rotas, de tal forma que determine um valor do atraso máximo para um conjunto de fluxos FTV na rede;

3. Validar a técnica proposta em uma topologia de rede WAN.

4.2 Modelo de engenharia de trafego

Este trabalho foi inspirado no modelo de engenharia de tráfego de Fortz e Thorup [18] que desenvolvem um modelo de custos para otimização da rede com protocolos tradicionais de roteamento de menor caminho em função da utilização, como mostra a equação 4.1.

$$\varphi' = \begin{cases} 1 & 0 \leq l/c < 1/3 \\ 3 & 1/3 \leq l/c < 2/3 \\ 10 & 2/3 \leq l/c < 9/10 \\ 70 & 9/10 \leq l/c < 1 \\ 500 & 1 \leq l/c < 11/10 \\ 5000 & 11/10 \leq l/c < \infty \end{cases} \quad (4.1)$$

Percebe-se que o custo do enlace cresce em função do aumento da fila nos nós. Este modelo considera o tráfego com chegadas exponenciais de Poisson e utiliza a equação 4.2 para o cálculo do tamanho da fila, como mostra a Figura 4.1

$$\hat{q} = \frac{\rho}{1 - \rho} \quad (4.2)$$

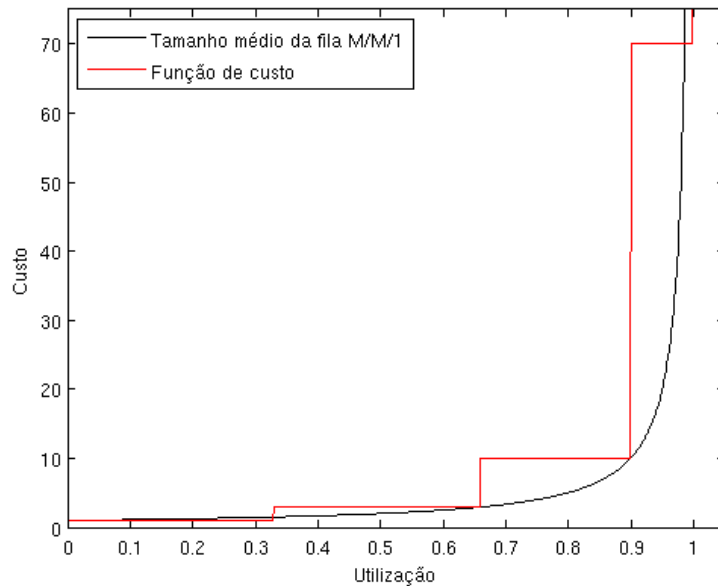


Figura 4.1: Crescimento do custo em função da utilização da fila M/M/1 para o modelo de Fortz

A partir da função de custo do Fortz foi extraído o problema de otimização linear, chamado de LPM-Fortz (*Linear Program Model of Fortz* - Modelo de programação linear de Fortz) cujo a função objetivo é dada pela equação 4.3, a equação 4.5 é a restrição de continuidade do fluxo, a equação 4.6 define a carga e da equação 4.7 à equação 4.12 são definidos os custos.

$$\text{Minimizar } \Phi = \sum_{(i,j) \in A} \phi_{(i,j)} \quad (4.3)$$

$$\text{Sujeito a :} \quad (4.4)$$

$$\sum_{j \in N} f_{(i,j)}^{(s,t)} - \sum_{j \in N} f_{(j,i)}^{(s,t)} = \begin{cases} d(s,t) & \text{se } i = s \\ -d(s,t) & \text{se } i = t \\ 0 & \text{se } i \neq s, t \end{cases} \quad i, s, t \in N \quad (4.5)$$

$$l(i, j) = \sum_{(s,t) \in N} f_{(i,j)}^{(s,t)} \quad (4.6)$$

$$\phi_{(i,j)} \geq l(i, j) \quad (4.7)$$

$$\phi_{(i,j)} \geq 3l(i, j) - \frac{2}{3}c(i, j) \quad (4.8)$$

$$\phi_{(i,j)} \geq 10l(i, j) - \frac{16}{3}c(i, j) \quad (4.9)$$

$$\phi_{(i,j)} \geq 70l(i, j) - \frac{178}{3}c(i, j) \quad (4.10)$$

$$\phi_{(i,j)} \geq 500l(i, j) - \frac{1468}{3}c(i, j) \quad (4.11)$$

$$\phi_{(i,j)} \geq 5000l(i, j) - \frac{19468}{3}c(i, j) \quad (4.12)$$

Mas, a crítica sobre este modelo é que no caso do tráfego de vídeo foi evidenciado que o mesmo não segue um processo de Poisson e o modelo de filas M/M/1 não conseguiria representar mais o comportamento do *buffer* nos nós da rede.

A Figura 4.2 exemplifica o crescimento máximo da fila para 99,9% do tráfego autossimilar no modelo FEP, com Coeficiente de Variação ($C_v = \sigma/\bar{l}$) igual a 0,1, 0,5 e 1,5, com coeficiente H variando de 0,5 a 0,9 em função da utilização com o atraso médio da fila com o modelo de Poisson, como descrito nos capítulos anteriores. Evidentemente o modelo FEP apresenta o crescimento muito mais acentuado em todos os casos, não apenas por representar um limite superior, mas pela dependência de longa duração presente no tráfego autossimilar. Considerando ainda que o tráfego de vídeo em alta resolução possui coeficiente de variação médio de 1,5, o *buffer* dos roteadores tende a se esgotar muito mais rapidamente do que o previsto pelo modelo de Poisson.

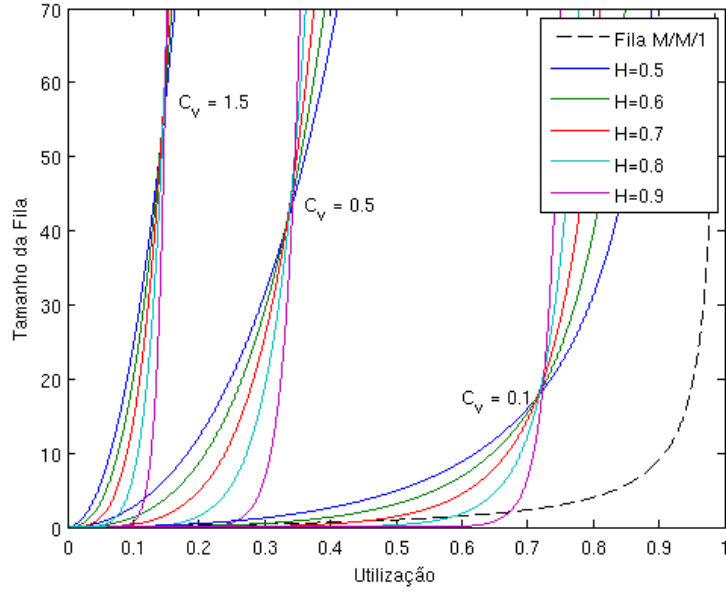


Figura 4.2: Comparação do tamanho da fila para vários modelos.

A partir da análise da Figura 4.2, considerando o crescimento médio da fila com coeficiente igual a 1,5, para 99,9% do tráfego do vídeo e o H igual a 0,87, características do vídeo que se deseja transmitir, neste caso o filme Tears of Steel caracterizado anteriormente. A partir deste comportamento foi desenvolvido um novo conjunto de custos comparando o tamanho da fila utilizado no LPM-Fortz com o tamanho da fila no modelo FEP, onde foi encontrado os intervalos de 0 a 20% de utilização distribuídos como mostrado na Equação 4.13 e os custos foram definidos acompanhando o crescimento da fila do FEP, como mostra a Figura 4.3.

$$\varphi' = \begin{cases} 1 & 0 \leq l/c < 2,5/100 \\ 5 & 2,5/100 \leq l/c < 5/100 \\ 20 & 5/100 \leq l/c < 8,5/100 \\ 150 & 8,5/100 \leq l/c < 15/100 \\ 1300 & 15/100 \leq l/c < 20/100 \\ 17000 & 20/100 \leq l/c < \infty \end{cases} \quad (4.13)$$

Com a função de custos definido é possível extrair um novo modelo de programação linear chamado de LPM-FEP (*Linear Program Model of Fractal Envelope Process - Modelo de Programação Linear do Processo Envelope Fractal*) mostrado da equação 4.14 à equação 4.23. O coeficiente angular dos segmentos é dado pelo custo dentro do intervalo enquanto os outros parâmetros são obtidos através de regressão linear, observando que neste modelo, $\phi_{(i,j)}$ representa o custo do enlace ao considerar que 99% do tráfego fique

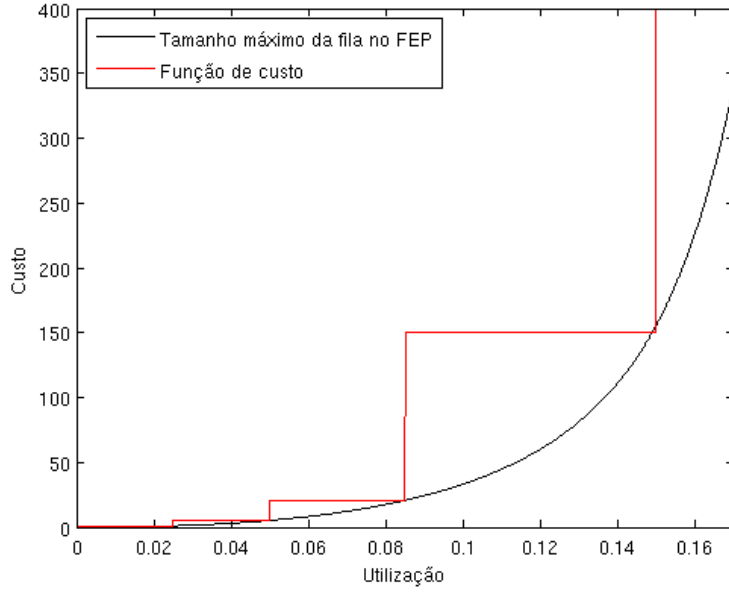


Figura 4.3: Crescimento do custos do enlace em função da utilização para o FEP

abaixo da curva do processo envelope.

$$\text{Minimizar } \Phi = \sum_{(i,j) \in A} \phi_{(i,j)} \quad (4.14)$$

$$\text{Sujeito a :} \quad (4.15)$$

$$\sum_{j \in N} f_{(i,j)}^{(s,t)} - \sum_{j \in N} f_{(j,i)}^{(s,t)} = \begin{cases} d(s,t) & \text{se } i = s \\ -d(s,t) & \text{se } i = t \\ 0 & \text{se } i \neq s, t \end{cases} \quad i, s, t \in N \quad (4.16)$$

$$l(i, j) = \sum_{(s,t) \in N} f_{(i,j)}^{(s,t)} \quad (4.17)$$

$$\phi_{(i,j)} \geq l(i, j) \quad (4.18)$$

$$\phi_{(i,j)} \geq 5l(i, j)H(i, j) - \frac{1}{10}c(i, j) \quad (4.19)$$

$$\phi_{(i,j)} \geq 20l(i, j)H(i, j) - \frac{17}{20}c(i, j) \quad (4.20)$$

$$\phi_{(i,j)} \geq 150l(i, j)H(i, j) - \frac{119}{10}c(i, j) \quad (4.21)$$

$$\phi_{(i,j)} \geq 1300l(i, j)H(i, j) - \frac{922}{5}c(i, j) \quad (4.22)$$

$$\phi_{(i,j)} \geq 17000l(i, j)H(i, j) - 3325c(i, j) \quad (4.23)$$

Uma vez que o objetivo também é atribuir custos menores a enlaces com menor autossimilaridade agregada, e conseqüentemente menores atrasos máximos, o parâmetro

de Hurst agregado aparece como multiplicador do coeficiente angular, como regulador do crescimento da função de custo, onde deverá gerar custos maiores a enlaces com um maior H_g e custos menores a enlaces com um menor H_g .

A Figura 4.4 compara o crescimento do custo do enlace em função da utilização entre o LPM-FEP com diversos H_g e o LPM-Fortz em escala logarítmica, para tráfegos com as mesmas características. Pode-se observar o crescimento acentuado do custo para até 50% de utilização média, onde o custo não ultrapassa 10000, o que é perfeitamente aceitável para o protocolo OSPFv2 que utiliza-se 16 bits para a divulgação dos custos do enlace, em que o custo varia em função do H do tráfego agregado.

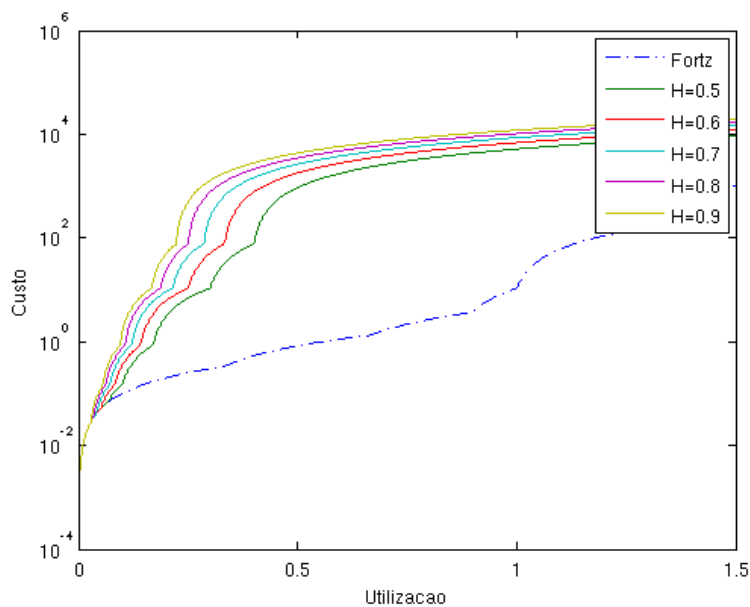


Figura 4.4: Comparação do crescimento do custo do enlace para o LPM-FEP com vários valores de H e o LPM-Fortz (azul pontilhado).

Uma vez que o tráfego autossimilar pode atingir atrasos muito elevados devido as suas rajadas de longa duração, o modelo LPM-FEP tem como objetivo redistribuir o tráfego reduzindo a autossimilaridade ao longo da rota afim de reduzir os atrasos de enfileiramento em função da utilização do enlace. Reduzir valor de H no tráfego agregado ou evitar que este valor alcance valores próximos de 1, poderia promover na rede um menor congestionamento e conseqüentemente, melhores valores de atraso.

Uma vez que um tráfego de alta intensidade com alta variância pode preencher todo o *buffer* rapidamente, mesmo com aproximadamente 25% de utilização média, como visto na Figura 4.2, o LPM-Fortz não consegue manipular o tráfego de forma eficiente, pois até 33% de utilização média o crescimento do seu custo é linear. Apenas por observar este gráfico, deduz-se que o controle dos tráfego deve ser, não antes dos 100%, mas antes de

30%. Dessa forma, o modelo LPM-FEP possui grande potencial para reduzir os atrasos máximos ao distribuir a carga pela rede utilizando o parâmetro H como um elemento no cálculo da métrica do enlace.

4.3 Cenário de simulação

4.3.1 Topologia da rede

Para a análise experimental foi considerada como topologia o *backbone* da rede da Rede Nacional de Pesquisa (RNP), uma rede IP que conecta 27 cidade com 80 enlaces pelo Brasil com capacidades de 100 Mbps a 20 Gbps, conforme Figura 4.5 disponível em [46]. Foi considerado que a rede utiliza o protocolo OSPFv2 definido no RFC 2328 [38] sob IPv4. As capacidades dos enlaces foram alteradas para reduzir a homogeneidade da rede e permitindo induzir o balanceamento de carga. A Figura 4.6 apresenta a topologia em forma de grafo.

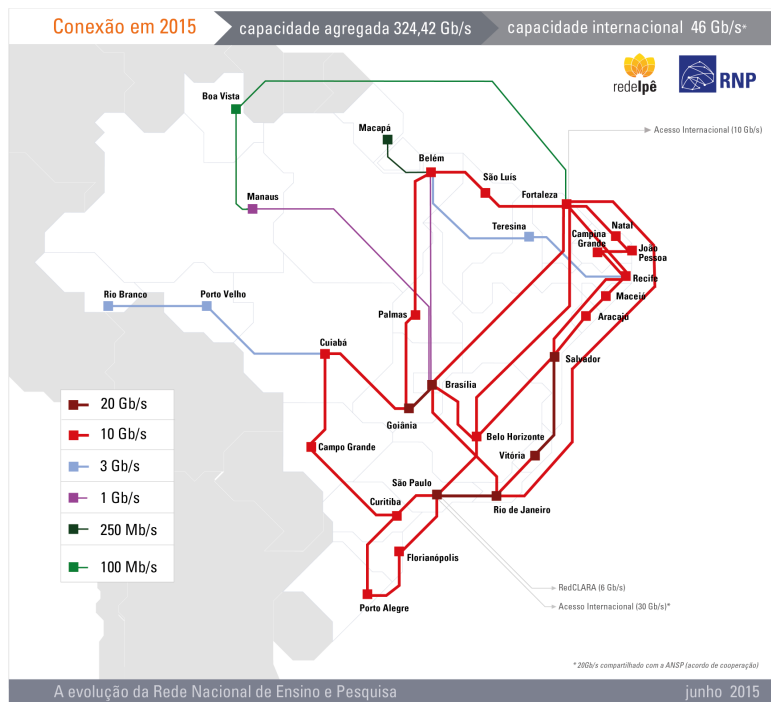


Figura 4.5: Mapa da Rede Nacional de Pesquisa.

Foi definido também que a rede utiliza o serviço de transmissão não orientado a conexão. O encaminhamento é realizado com as informações dos endereços de origem e destino e o roteamento feito apenas pela métrica do enlace.

O serviço de transmissão não orientado a conexão foi selecionado por ser é mais resistente a falhas. Os pacotes são encaminhados independentes e as rotas dependem apenas

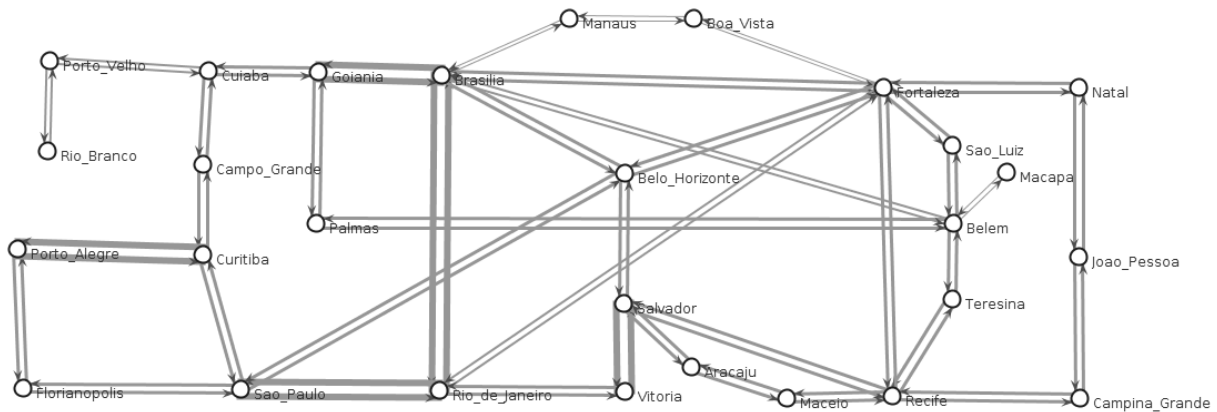


Figura 4.6: Topologia da Rede Ipê utilizada para simulação

do peso do enlace. Dessa forma é possível uma rede com maior escalabilidade, mas robusta e com grande simplicidade (consequentemente mais barata) que permite uma fácil adaptação as necessidades do tráfego.

O fato de utilizar um protocolo tradicional de menor caminho permitirá, neste caso, um cálculo dinâmico das métricas do enlace, podendo permitir no futuro uma adaptação mais elegante e simples à especificação original do protocolo.

A cada enlace foi atribuído um tráfego de fundo com autossimilaridade aleatória que varia de 0,5 a 0,8 dependendo do cenário e uma carga correspondente a 5%, 10%, 15% e 20% de utilização do enlace, como mostrado na Tabela 4.1. Este tráfego de fundo representa o fluxo agregado de outras aplicações em que a rede ofereceu serviço de transporte e que competem por recursos com o tráfego principal.

4.3.2 Métricas e parâmetros

No cenário de simulação foram consideradas as seguintes métricas:

- Utilização média: representa a razão da soma de toda a utilização da rede pelos enlaces que foram utilizados. A utilização média consegue expressar como o tráfego está distribuído por toda a rede, sendo uma baixa utilização média o recomendado.
- Atraso máximo médio da rede: é a razão da soma dos atrasos máximos de todos os enlaces pelo número de enlaces. Esta métrica é a mais importante neste trabalho pois reflete o limite superior médio que um pacote espera no *buffer* do roteador antes de ser encaminhado para o próximo salto. De acordo com Szigetzi em [52] o atraso médio em cada nó não deve ultrapassar 5 segundos para transmissão de vídeo em cada nó e Tanenbaum em [54] cita que até 15 segundos de atraso médio para o vídeo ao vivo é plenamente aceitável. Este valor se deve ao tráfego de vídeo

Tabela 4.1: Características dos Enlaces usados na simulação

Enlace	Capacidade	T. de fundo a 5%	T. de fundo a 10%	T. de fundo a 15%	T. de fundo a 20%
Rio Branco<->Porto Velho	3000 Mbps	150 Mbps	300 Mbps	450 Mbps	600 Mbps
Cuiaba<->Porto Velho	3000 Mbps	150 Mbps	300 Mbps	450 Mbps	600 Mbps
Boa Vista<->Manaus	1000 Mbps	50 Mbps	100 Mbps	150 Mbps	200 Mbps
Cuiaba<->Campo Grande	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Cuiaba<->Goiania	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Goiania<->Palmas	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Goiania<->Brasilia	20000 Mbps	1000 Mbps	2000 Mbps	3000 Mbps	4000 Mbps
Brasilia<->Belem	5000 Mbps	250 Mbps	500 Mbps	750 Mbps	1000 Mbps
Palmas<->Belem	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Macapa<->Belem	250 Mbps	12.5 Mbps	25 Mbps	37.5 Mbps	50 Mbps
Brasilia<->Manaus	1000 Mbps	50 Mbps	100 Mbps	150 Mbps	200 Mbps
Belem<->Sao Luiz	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Sao Luiz<->Fortaleza	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Boa Vista<->Fortaleza	1000 Mbps	50 Mbps	100 Mbps	150 Mbps	200 Mbps
Belem<->Teresina	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Campina Grande<->Joao Pessoa	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Porto Alegre<->Florianopolis	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Curitiba<->Campo Grande	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Curitiba<->Porto Alegre	20000 Mbps	1000 Mbps	2000 Mbps	3000 Mbps	4000 Mbps
Curitiba<->Sao Paulo	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Florianopolis<->Sao Paulo	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Brasilia<->Belo Horizonte	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Rio de Janeiro<->Vitoria	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Teresina<->Recife	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Sao Paulo<->Rio de Janeiro	20000 Mbps	1000 Mbps	2000 Mbps	3000 Mbps	4000 Mbps
Brasilia<->Fortaleza	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Fortaleza<->Natal	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Natal<->Joao Pessoa	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Fortaleza<->Recife	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Campina Grande<->Recife	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Belo Horizonte<->Fortaleza	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Belo Horizonte<->Salvador	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Fortaleza<->Rio de Janeiro	5000 Mbps	250 Mbps	500 Mbps	750 Mbps	1000 Mbps
Sao Paulo<->Belo Horizonte	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Brasilia<->Rio de Janeiro	20000 Mbps	1000 Mbps	2000 Mbps	3000 Mbps	4000 Mbps
Vitoria<->Salvador	20000 Mbps	1000 Mbps	2000 Mbps	3000 Mbps	4000 Mbps
Salvador<->Aracaju	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Aracaju<->Maceio	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Recife<->Maceio	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Salvador<->Recife	10000 Mbps	500 Mbps	1000 Mbps	1500 Mbps	2000 Mbps
Média	9731.25	486.58	973.13	1459.70	1946.25

ser geralmente unidirecional, pouco sensível a atrasos (o vídeo pode demorar vários segundos para iniciar) e ao *jitter* que pode ser compensado com o uso de *buffers*.

Para a simulação serão considerados quatro parâmetros da rede: a taxa de transmissão do tráfego de fundo, a taxa de transmissão do tráfego de vídeo e o índice de autossimilaridade de cada enlace.

O tráfego de fundo representa o tráfego que já está contido na rede e que por sua vez já foi roteado e está sendo transmitido. Este tráfego apresenta coeficiente de variação de 0,5 e um parâmetro H que varia de acordo com os casos estudados e que serão descritos em cada cenário.

O tráfego principal é um tráfego de alta intensidade que representa o fluxo de vídeo cuja fonte transmite para um destino. Este tráfego é medido em Mbps, é caracterizado apenas por um par origem-destino e é aquele que precisa ter garantia de qualidade de serviço.

Para estimar a largura de banda utilizada deste tráfego, foi considerada o estimador de banda passante do Processo Envelope Fractal (FEP - *Fractal Envelope Process*) [16]. Em seu cálculo foi considerado 17.595 quadros do filme *Tears of Steel* com parâmetro de Hurst igual 0,87, taxa de transmissão média de 5,676 Mbps, *buffer* de 424.000 bytes e probabilidade de perda de 10^{-3} . Desta forma estimou-se que é necessário 27.369,9 Kbps

para a transmissão de um fluxo de vídeo com a probabilidade de 0,1% do fluxo ultrapassar essa taxa.

Para a otimização dos pesos do enlaces foram considerados 10 fluxos de vídeo idênticos com estimativa de transmissão de 273.699.000 bps para a estimativa com o processo envelope e 56.768.356 bps para o tráfego médio.

4.3.3 Descrição da composição do tráfego nos cenários de simulação

Foram simulados 24 casos de simulação definidos com base em dois parâmetros: distribuição do parâmetro H na rede e intensidade do tráfego de fundo. Foram comparados para cada simulação três modelos: LPM-FEP, LPM-Fortz e InvCap, este último proposto pela CISCO [18] como configuração padrão de seus roteadores, em que o custo do enlace é definido como o inverso da capacidade. Para cada cenário foi aplicada uma carga correspondente as variações da taxa média do tráfego de vídeo FTV.

Quanto a distribuição do H na rede foram definidos dois cenários, descritos a seguir:

- **Cenário 1:** foram geradas dez topologias aleatórias apresentando 30% do tráfego de fundo dos enlaces com $H = 0,5$, ou seja, 30% dos enlaces não apresentam autossimilaridade e os restantes dos enlaces apresentam tráfego de fundo onde os coeficientes H variam de 0,5 a 0,85, que foram dispostos pelos enlaces da rede seguindo uma distribuição uniforme.
- **Cenário 2:** Foram gerados dez topologias aleatórias em que todos os enlaces apresentam um H que varia de 0,5 a 0,85 distribuídos de forma uniforme.

Para ambos os cenários foi aplicado o fluxograma da Figura 4.7. Com as informações extraídas da topologia e do fluxo de vídeo que se deseja transmitir, foi modelado um *script* de otimização em AMPL (*A Mathematical Programming Language* - Uma linguagem de programação matemática) [21]. Estas informações contém o parâmetro de Hurst agregado de cada enlace, a capacidade total dos enlaces, a matriz de demanda que contém os pares origem-destino de todo o tráfego tráfego de fundo e a largura de banda estimada do vídeo que será transmitido. O procedimento completo da simulação está descrito no algoritmo 1. No final do procedimento, para cada um dos 24 casos simulados, foi calculada a média entre as 10 topologias diferentes do atraso máximo médio da rede e utilização média, de forma refletisse a esperança das métricas na rede.

Para cada cenário foi coletado o parametro H_f , a taxa média do tráfego de fundo (\bar{b}) e o desvio padrão do tráfego de fundo (σ). Também foi coletado a capacidade de cada enlace (c) e o *buffer* dos nós (K). A demanda do tráfego principal, que é o tráfego de vídeo

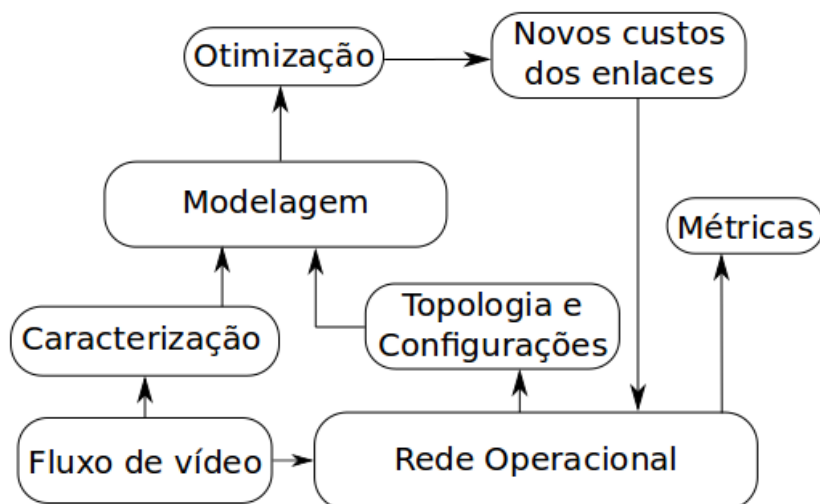


Figura 4.7: Fluxograma utilizado para a simulação

que se deseja transmitir foi gerada com base nas características dos vídeos apresentado no capítulo 3. O vídeo utilizado foi do filme *Tears of Stell* com resolução 4K (4096 x 1744 *pixels*), 24 quadros por segundo e taxa média (\bar{a}) de 5,676 Mbps. Com o estimador de largura de banda do FEP (BE), apresentado no capítulo 3, foi calculado que são necessários 27,3699 Mbps para transmissão com perdas de 0,1% de pacotes, coeficiente de variação igual a 1,8, H igual a 8,7 e *buffers* de 424.000 bytes. Para simular a alta demanda do tráfego FTV estes valores (BE e \bar{a}) foram multiplicados por 10, ou seja se considera a transmissão concorrente de 10 fluxos de vídeo FTV.

Após o cálculo do parâmetro de Hurst agregado e a variância agregada, no LPM-Fortz, a demanda do vídeo FTV é atualizada com a soma dos dez fluxos de vídeo, ou seja, 56,76 Mbps representando o tráfego de Poisson. Para a demanda do vídeo ns LPM-FEP foi utilizada o estimador de banda passante do FEP para os dez fluxos resultando em 273,699 Mbps. Observa-se que a expectativa da largura de banda do tráfego autossimilar foi 4,82 vezes maior que a média utilizada no modelo LPM-Fortz.

Após a modelagem dos dados da rede, foi utilizado um programa de resolução de programação linear chamado GLPSOL do pacote GLPK (*GNU Linear Programming Kit*)[21] para Linux onde a resposta do modelo corresponde a um novo conjunto de custos para todos os enlaces da rede.

Com os resultado do programa linear, os custos dos enlaces da rede são atualizados e é executado o TOTEM aplicada uma carga que varia de 100 Mbps a 2500 Mbps, para todos os modelos e para cada uma das 10 topologias, obedecendo a regra em que o tráfego de fundo mais a carga não devem ultrapassar 30% da capacidade do enlace pois com esta utilização os atrasos se tornam inaceitáveis. O modelo InvCap não precisa de custos, uma vez que o custo do enlace será o inverso de sua capacidade e o TOTEM efetua o cálculo

automaticamente.

Em todos os casos de estudo foi aplicado uma carga que variava de intensidade a partir de 100 Mbps até 2500 Mbps que corresponde ao intervalo de variações do tráfego de vídeo no enlace, mas de tal forma que o tráfego total sobre o enlace (tráfego de fundo mais tráfego de vídeo) não ultrapasse 3000 Mbps.

Algoritmo 1: Algoritmo de simulação

início

Passo 1: Obter informações da topologia e tráfego de fundo:

Extraia $\rightarrow H_f$, taxa média do tráfego de fundo de cada enlace (\bar{b}), desvio padrão (σ_f), *buffer* dos nós (K), capacidade dos enlaces (c);

Passo 2: Caracterizar tráfego de vídeo:

Extraia $\rightarrow H_v$, desvio padrão (σ_v), taxa média (\hat{a});

Passo 3: Calcular H agregado (H_g), variância agregada σ_g^2 e banda efetiva (BE):

Calcule $\sigma_g^2 \leftarrow \sigma_f^2, \sigma_v^2$;

Calcule $H_g \leftarrow H_f, H_v$;

Calcule $BE \leftarrow H_v, \sigma_v, \hat{a}, K, \sigma_v$;

Passo 4: Gerar matriz de tráfego (TM):

Para LPM-Fortz;

$TM_o \leftarrow \hat{a}x10, \bar{b}$;

Para LPM-FEP;

$TM_f \leftarrow BEx10, \bar{b}, H_g$;

Passo 5: Extrair pesos ϕ dos modelos:

Resolva LPM-Fortz $\leftarrow TM_o$;

Extraia LPM-Fortz $\rightarrow \phi_o$;

Resolva LPM-FEP $\leftarrow TM_f$;

Extraia LPM-FEP $\rightarrow \phi_f$;

Passo 6: Atualizar pesos e extrair utilização do enlace no TOTEM:

Seja $l = 100$; $l + \bar{b} \leq 3000\bar{b}$; $l = l + 100$;

Para LPM-Fortz, LPM-FEP:

TOTEM $\leftarrow l, \hat{b}, \phi_o$;

TOTEM $\rightarrow \rho$;

Para InvCap:

TOTEM $\leftarrow l, \hat{b}$;

TOTEM $\rightarrow \rho$;

Passo 7: Calcular métricas: ;

Para cada modelo:

Calcule Atraso máximo médio $\leftarrow H_g, \rho, c, \sigma_g, \rho$;

Calcule Utilização média da rede $\leftarrow \rho$;

fim

Para a transmissão do tráfego FTV também foram feitas algumas considerações. Uma vez que a transmissão que centenas de fluxos para milhões de clientes é algo inviável para a infraestrutura da rede, foi considerada a utilização de uma arquitetura alternativa. Scan-

darolli et al. em [49] propõe uma arquitetura que move a fase de interpolação para um servidor na nuvem de forma que a interpolação dos fluxos mais o mapa de profundidade possa ser feita em tempo real e transmitida para dispositivos com baixo poder de processamento, como mostra a figura 4.8. Desta forma, a emissora que transmite os vídeos se conecta a um servidor de interpolação mais próximo geograficamente do cliente final através de uma arquitetura cliente-servidor e por sua vez o servidor de interpolação se conecta aos clientes finais através de outro mecanismo de distribuição de vídeo, como DASH ou P2P, por exemplo. Este trabalho considera apenas essa conexão entre o transmissor e o servidor de interpolação.

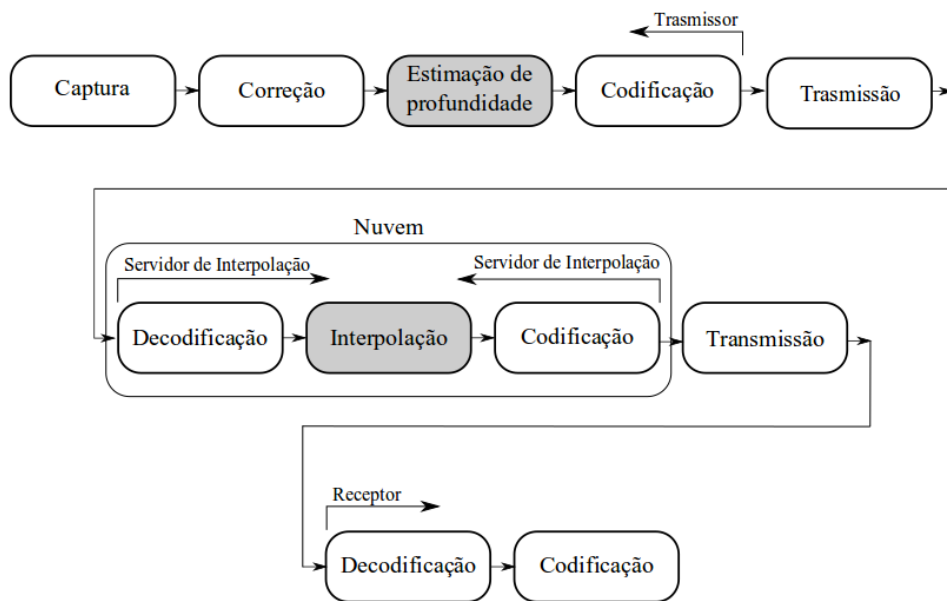


Figura 4.8: Arquitetura FTV utilizada neste trabalho proposta em [49]

Simulador TOTEM possui código aberto escrito em Java e XML e possui diversos métodos para engenharia de tráfego baseado em IP, MPLS, Híbrido IP/MPLS e processos de decisão BGP já implementado. Possui interface gráfica e linha de comando para roteamento SPF, CSPF e outros tipos de roteamento por pacotes e circuitos virtuais. Já possui diversos algoritmos para otimização através de diversas heurísticas, como Local Search, Busca Tabo e Column Generation Approach. A heurística Column Generation Approach foi a escolhida para resolução dos problemas neste trabalho, uma vez que foi a mesma utilizada por Fortz et al em [18] por considerar que o MCNFP é uma boa aproximação para o problema do roteamento IGP [19].

O TOTEM realizou todo o roteamento e encaminhamento do tráfego conforme o protocolo OSPF, retornando a utilização em cada nó da rede. Foi utilizado o ECMP tendo em vista que há necessidade de balanceamento de carga para a transmissão do vídeo por possuir altíssima largura de banda. A partir dos resultados de utilização de cada

enlace foi calculada a média do atraso máximo e da utilização nas 10 topologias com autossimilaridade aleatórias que foram geradas.

4.4 Resultados e discussão

4.4.1 Cenário 1

As Figuras 4.9 e 4.10 apresentam o atraso médio e a utilização média da rede em função da carga com o tráfego de fundo igual a 5% da capacidade dos enlaces. A taxa do tráfego de vídeo varia entre 100 Mbps e 2500 Mbps.

O LPM-FEP manteve o atraso máximo mais baixo do que o LPM-Fortz mesmo quando aplicada mais carga. Foram alcançados 5 segundos com 2300 Mbps de carga média na rede, 75% a mais da carga em que LPM-Fortz atingiu este mesmo atraso e junto com o tráfego de fundo corresponde a 28% da capacidade média da rede, porém, pode-se observar que a utilização média da rede do LPM-FEP foi levemente maior do que os outros modelos, porém, a diferença das utilizações foram de menos de 10%. Já o modelo InvCap apresenta a menor utilização de todos e uma curva de atrasos que acompanha o LPM-Fort.

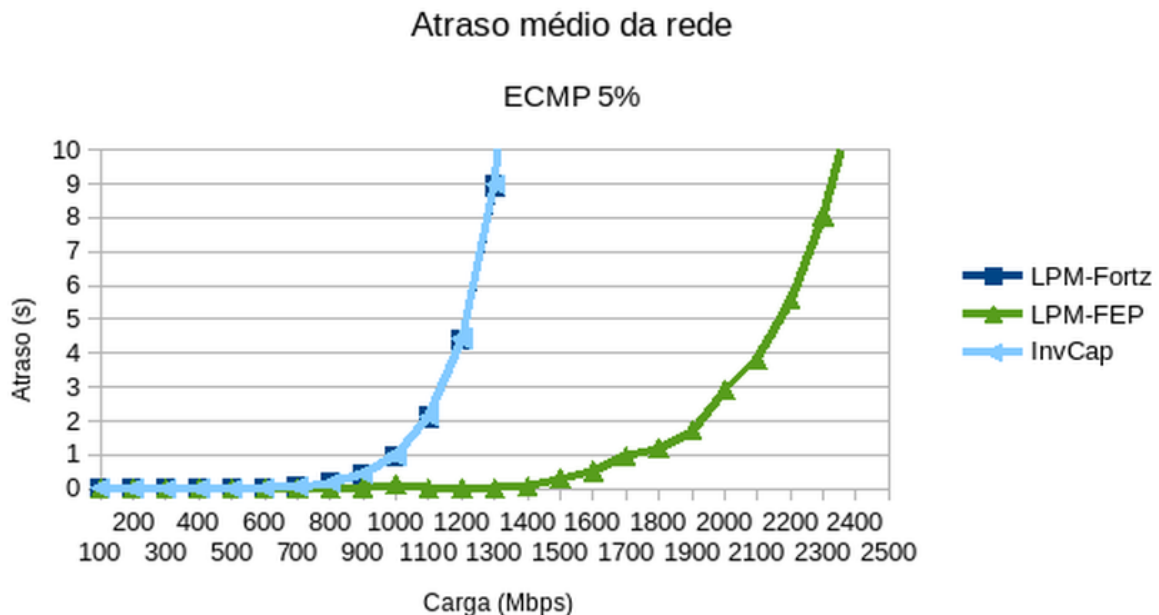


Figura 4.9: Atraso médio da rede com 5% de tráfego de fundo

Quando o tráfego de fundo aumenta para 10% há um pico de utilização média do LPM-FEP a 800 Mbps de carga, como pode-se verificar no gráfico da Figura 4.12 esta utilização

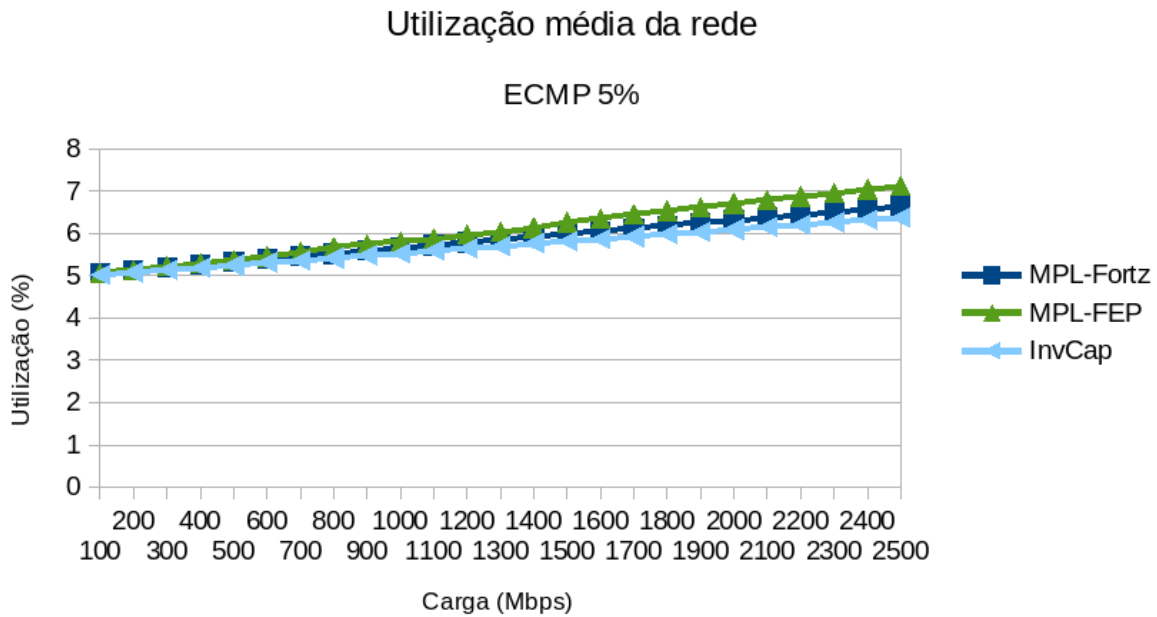


Figura 4.10: Utilização média da rede com 5% de tráfego de fundo

é aproximadamente 10% a mais do que a apresentada pelo LPM-Fortz sob a mesma carga. Porém, o atraso médio da rede do LPM-FEP mostrado na Figura 4.11, apresenta valores máximos médios de 5 segundos entre 1100 Mbps e 1200 Mbps de carga, que junto com o tráfego de fundo é equivalente a 22% da capacidade média da rede, enquanto o LPM-Fortz apresenta atraso de 5 segundos entre 900 Mbps e 1000 Mbps. O modelo InvCap neste cenário apresenta utilização menor do que todos os casos apresentados e o atraso acompanha o atraso do LPM-Fortz.

Com 15% de tráfego de fundo, como mostra a Figura 4.13, o LPM-FEP apresenta um atraso levemente maior do que o LPM-Fortz até a carga de 100 Mbps, porém com o aumento da carga, este modelo manteve atrasos muito mais baixos e também apresenta a mesma utilização que o LPM-Fortz, como mostra a Figura 4.14. Um fato observado na Figura 4.14 foi que o modelo do InvCap possui a menor utilização média, indicando uma boa distribuição do tráfego, entretanto, apresenta valores de atrasos maiores que a LPM-FEP e o atraso médio do LPM-FEP foi inferior a 5 segundos até a carga de 1500 Mbps, onde, junto com o tráfego de fundo equivale a 30,4% da capacidade média da rede, indicando um ganho de 87,5% de uso da capacidade com o mesmo atraso que o LPM-Fortz.

Como último caso deste cenário, a Figura 4.15 e 4.16 apresentam os gráficos de atraso máximo médio e utilização média, respectivamente, quando a carga é aplicada a enlaces com 20% do tráfego de fundo. Os atrasos do LPM-Fortz e InvCap se mantiveram abaixo

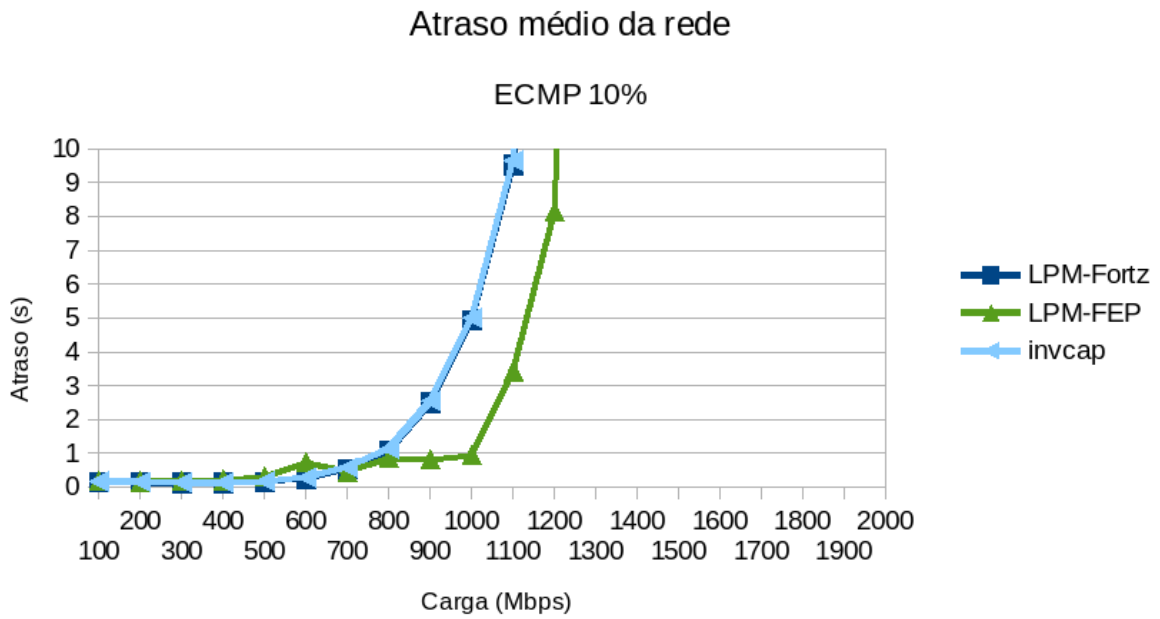


Figura 4.11: Atraso médio da rede com 10% de tráfego de fundo

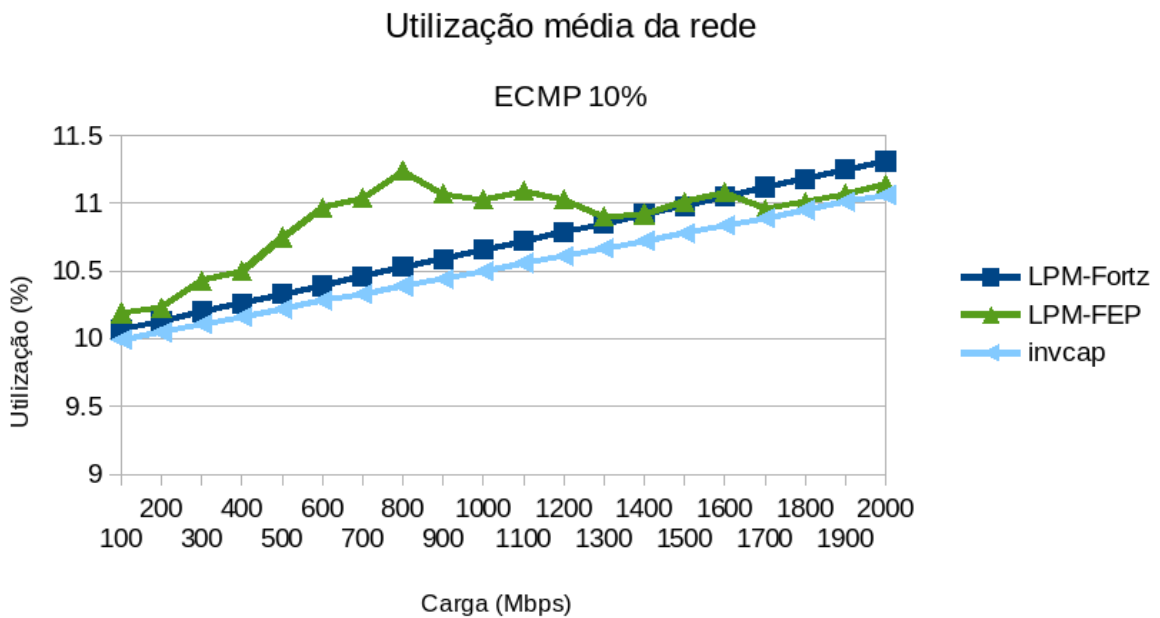


Figura 4.12: Utilização média da rede com 10% de tráfego de fundo

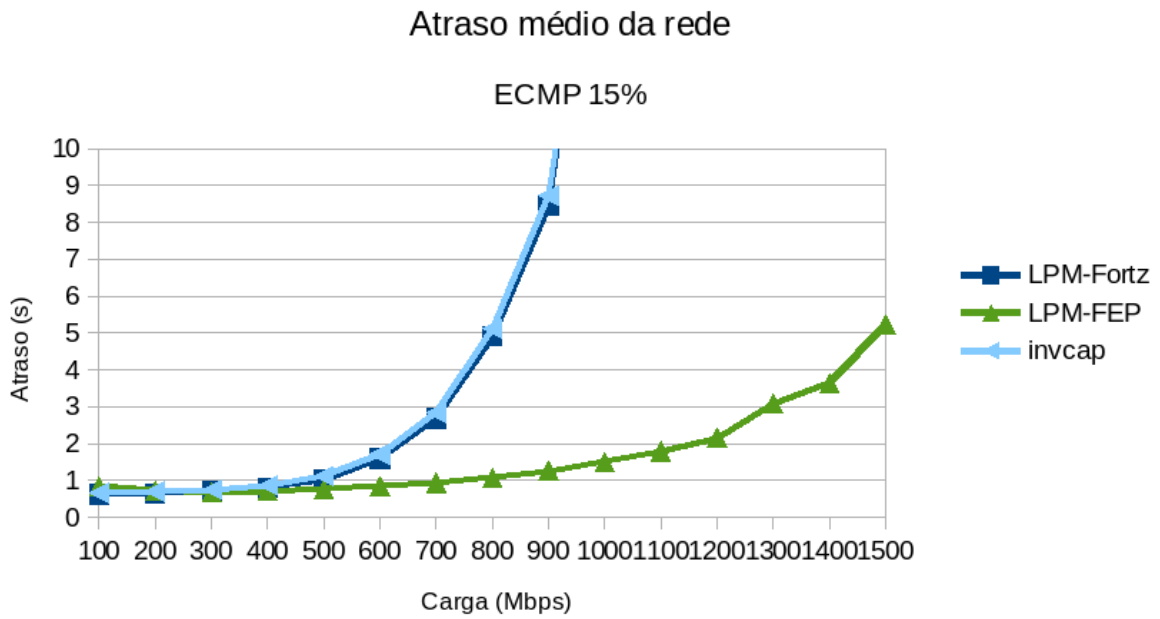


Figura 4.13: Atraso médio da rede com 15% de tráfego de fundo

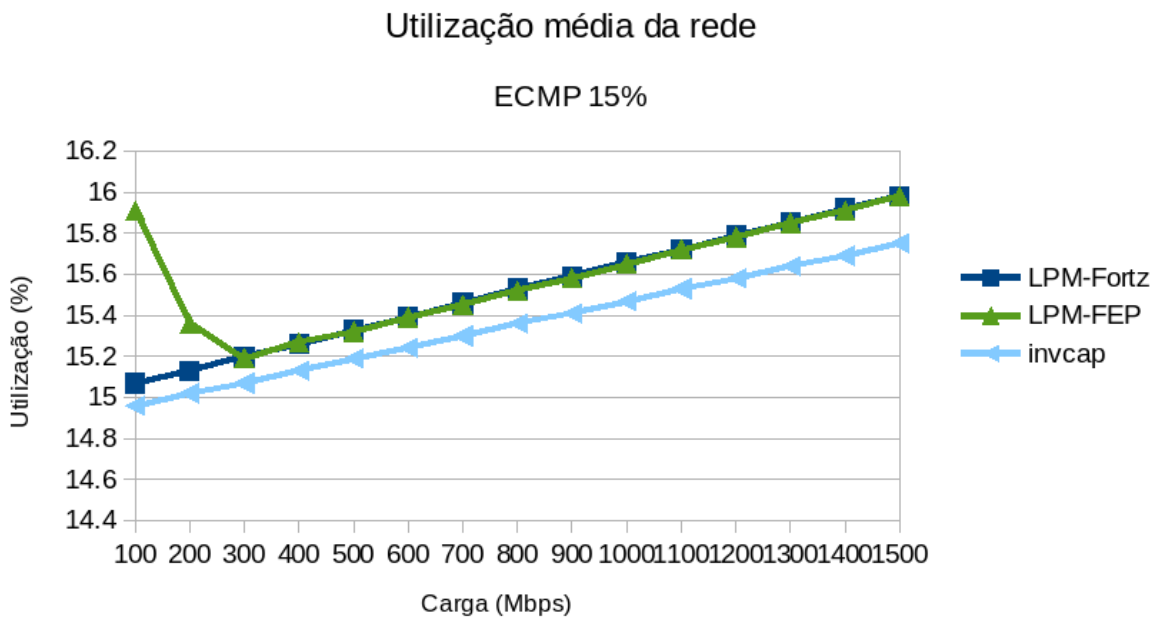


Figura 4.14: Utilização média da rede com 15% de tráfego de fundo

de 5 segundos até a carga de 500 Mbps. Este valor mais o tráfego de fundo corresponde a aproximadamente 25% da capacidade média do enlace. O modelo LPM-FEP manteve o atraso superior ao LPM-Fortz e Invcap até 600 Mbps, porém, a pesar do atraso oscilar entre 5 e 8 segundos, manteve esta variação para toda a carga aplicada, obtendo melhores resultados quando o enlace recebia mais carga. Pode-se observar no gráfico da Figura 4.16 que a utilização média da rede foi muito alta, comparado aos outros dois modelos, durante quase toda a simulação, o que demonstra a forte limitação da rede diante o aumento de carga do tráfego autossimilar que apresenta uma tendência de ter atrasos bem acima dos aceitáveis para esse tipo de aplicação.

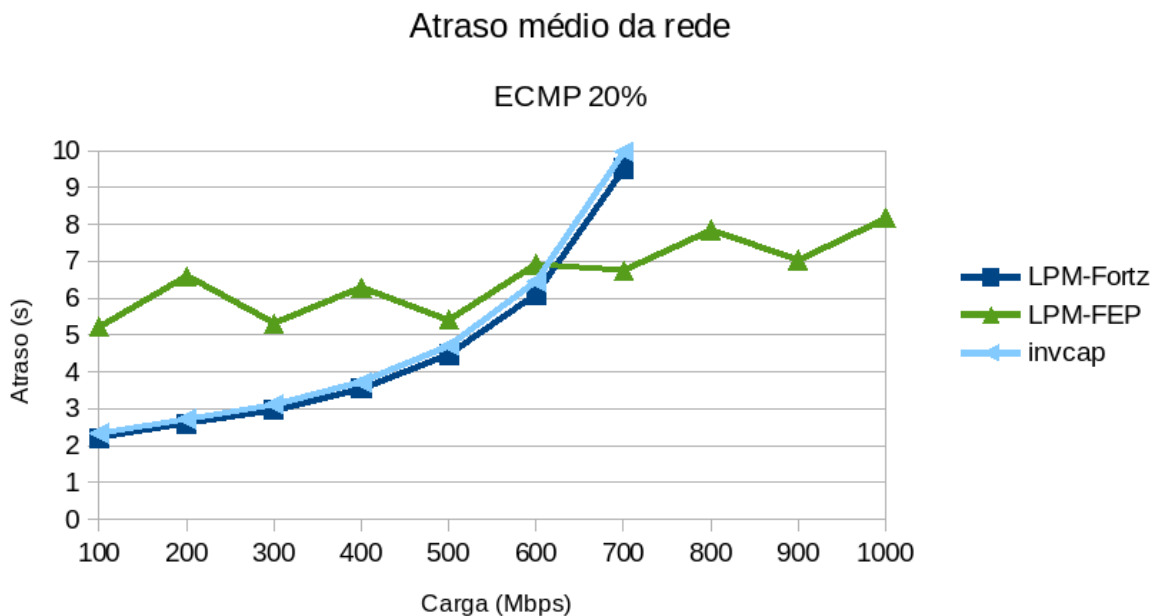


Figura 4.15: Atraso médio da rede com 20% de tráfego de fundo

4.4.2 Cenário 2

Neste cenário, todo o tráfego de fundo é autossimilar, conforme explicado na seção 4.3.3. As Figuras 4.17 e 4.18 representam o atraso máximo e a utilização média da rede com tráfego de fundo igual a 5% da capacidade do enlace. Este resultado se assemelha muito a resultado do cenário 1 onde o modelo LPM-FEP apresenta atrasos menores por toda a simulação e atrasos de 5 segundos quando a carga atinge 2200 Mbps. Estes 2200 Mbps mais o tráfego de fundo corresponde a aproximadamente 27,2% da capacidade média dos enlaces da rede. O modelo com o inverso da capacidade possui os mesmos atrasos do LPM-Fortz atingindo 5 segundos de atraso com pouco mais de 1200 Mbps, 45% menos

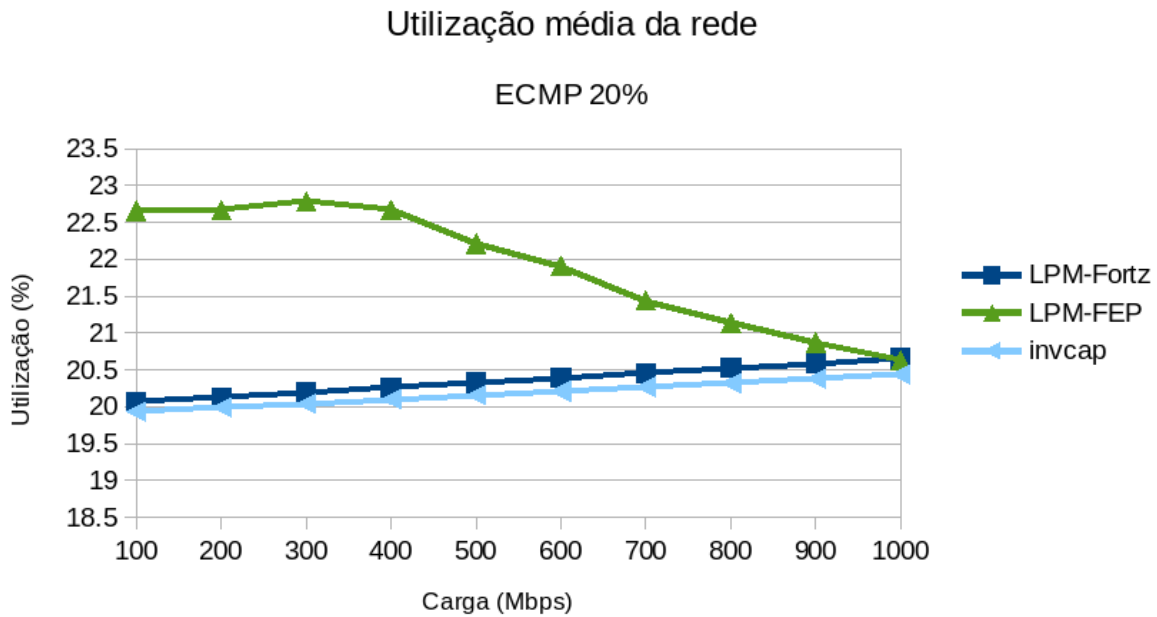


Figura 4.16: Utilização média da rede com 20% de tráfego de fundo

tráfego do que o LPM-FEP. A utilização média da rede do LPM-FEP foi levemente superior que no outro cenário, sendo que o modelo InvCap possui a menor utilização durante toda a simulação, entretanto apresenta atrasos maiores que o LPM-FEP.

As Figuras 4.19 e 4.20 apresentam os gráficos de atraso máximo médio e utilização média, respectivamente, para a topologia com 10% de utilização nos enlaces. Sob este tráfego de fundo o gráfico do LPM-FEP apresenta um atraso de aproximadamente 8 segundos quando 200 Mbps de carga é aplicado com rápida redução, se tornando menor do que o atraso calculado no LPM-Fortz. O modelo InvCap, neste cenário apresenta o mesmo atraso que o modelo de Fortz, apresentando atrasos de 5 segundos com carga de 1000 Mbps, enquanto que o LPM-FEP apresenta o mesmo atraso com cargas superiores a 1100 Mbps, com 10% a mais de carga do que os outros modelos, que somado ao tráfego de fundo corresponde a aproximadamente 22% da capacidade média da rede. Apesar de apresentar utilização até 10% a mais do que os outros modelos com baixas cargas, o LPM-FEP passa a apresentar utilização semelhante aos outros modelos a partir da carga de 1200 Mbps.

A Figura 4.19 mostra o atraso máximo médio quando o tráfego de fundo aumenta para 15% da capacidade do enlace, percebe-se uma distribuição mais uniforme do tráfego no LPM-FEP onde o atraso cresce mais lentamente, apresentando atrasos inferiores a 5 segundos em todo o intervalo de simulação, que somado ao tráfego de fundo corresponde a 30% da capacidade média da rede. O modelo InvCap possui um atraso um pouco

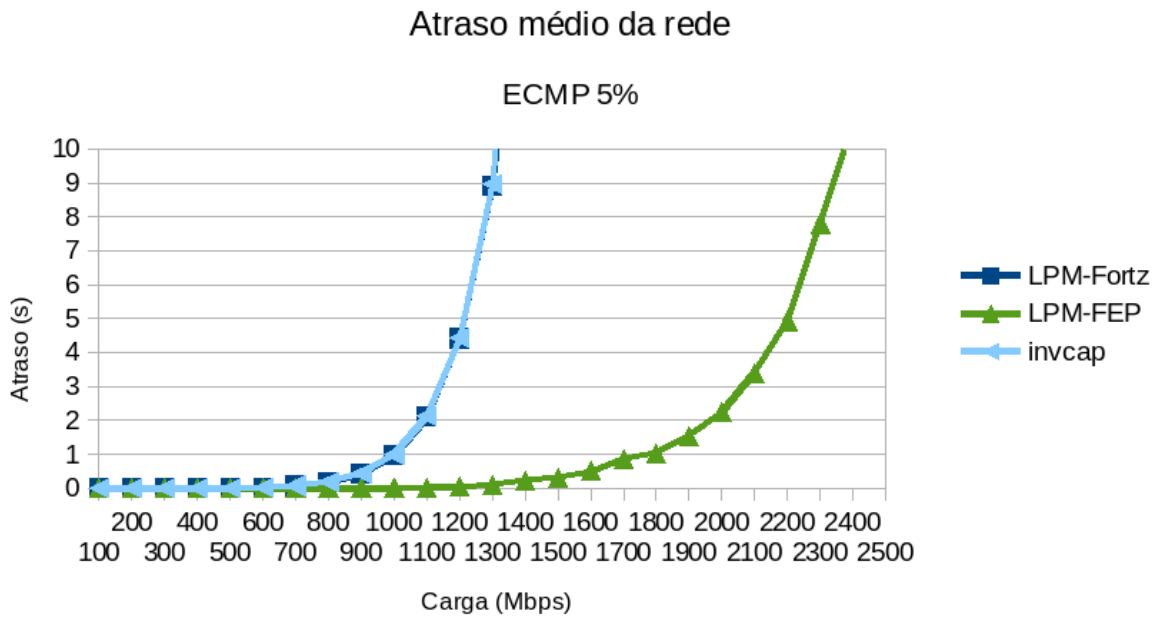


Figura 4.17: Atraso médio da rede com 5% de tráfego de fundo

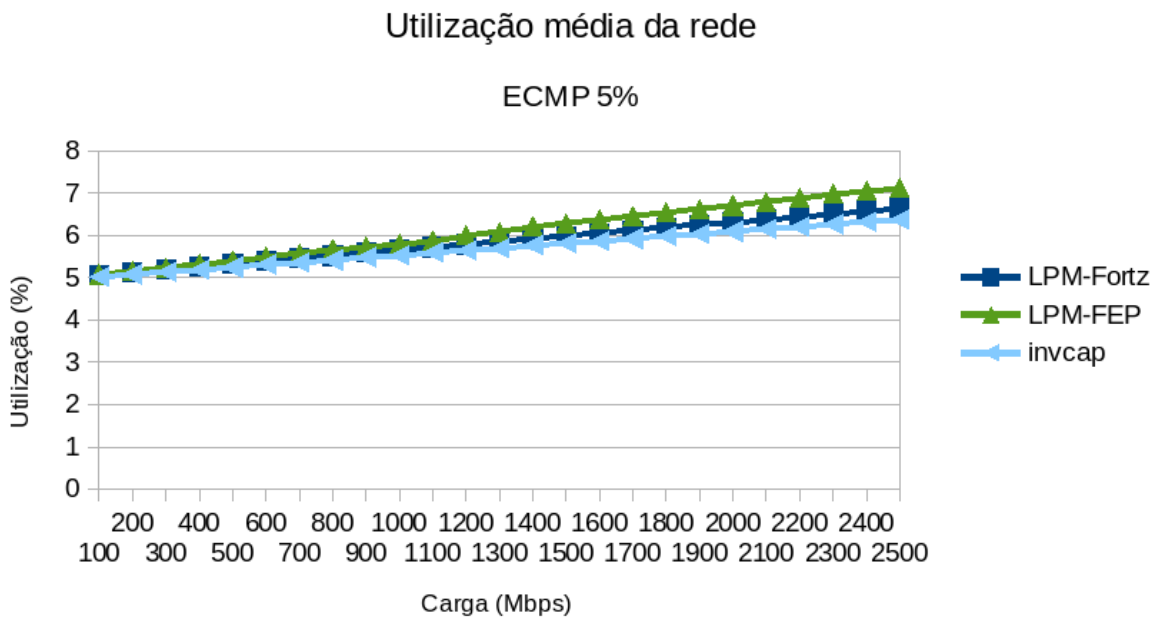


Figura 4.18: Utilização média da rede com 5% de tráfego de fundo

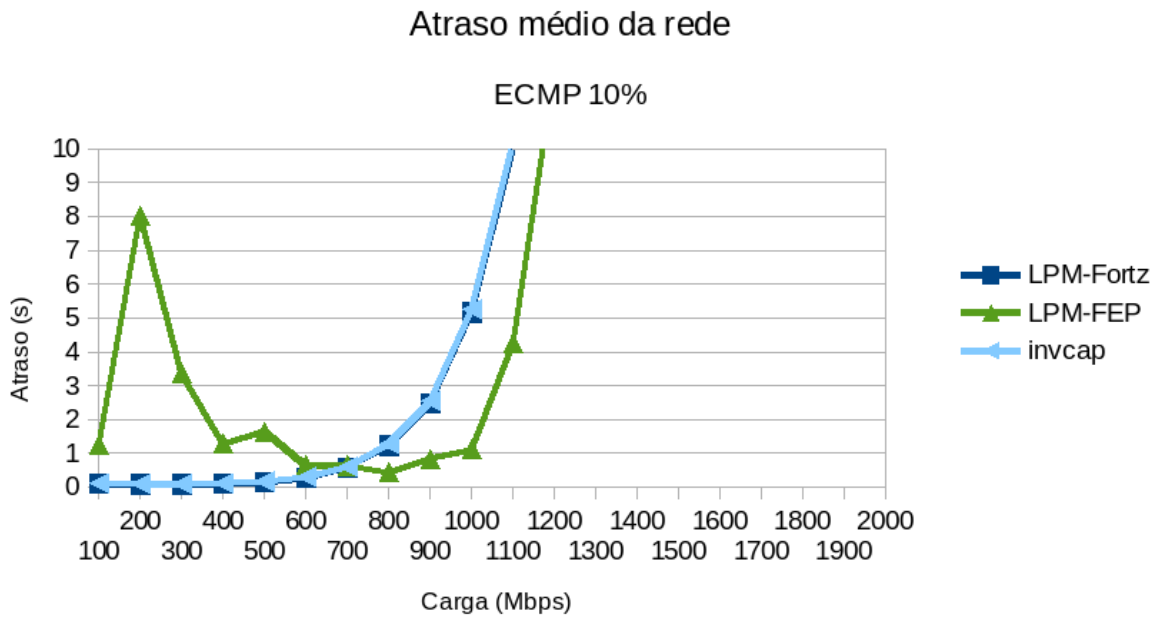


Figura 4.19: Atraso médio da rede com 10% de tráfego de fundo

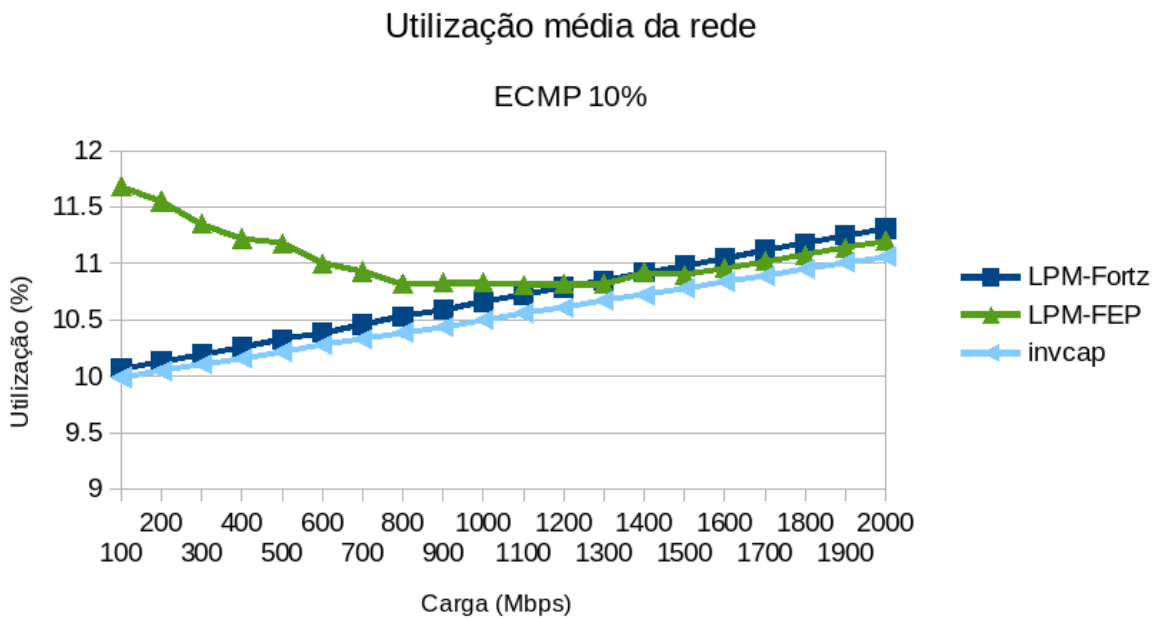


Figura 4.20: Utilização média da rede com 10% de tráfego de fundo

menor que o modelo de Fortz, mas suas curvas apresentam um comportamento semelhante atingindo atrasos de 5 segundos com um pouco mais de 700 Mbps, 53% menos do que o LPM-FEP. Quanto a utilização, mostrada na Figura 4.22, o modelo InvCap apresentou a menor utilização média, enquanto o LPM-FEP e LPM-Fortz apresentaram utilização muito semelhante.

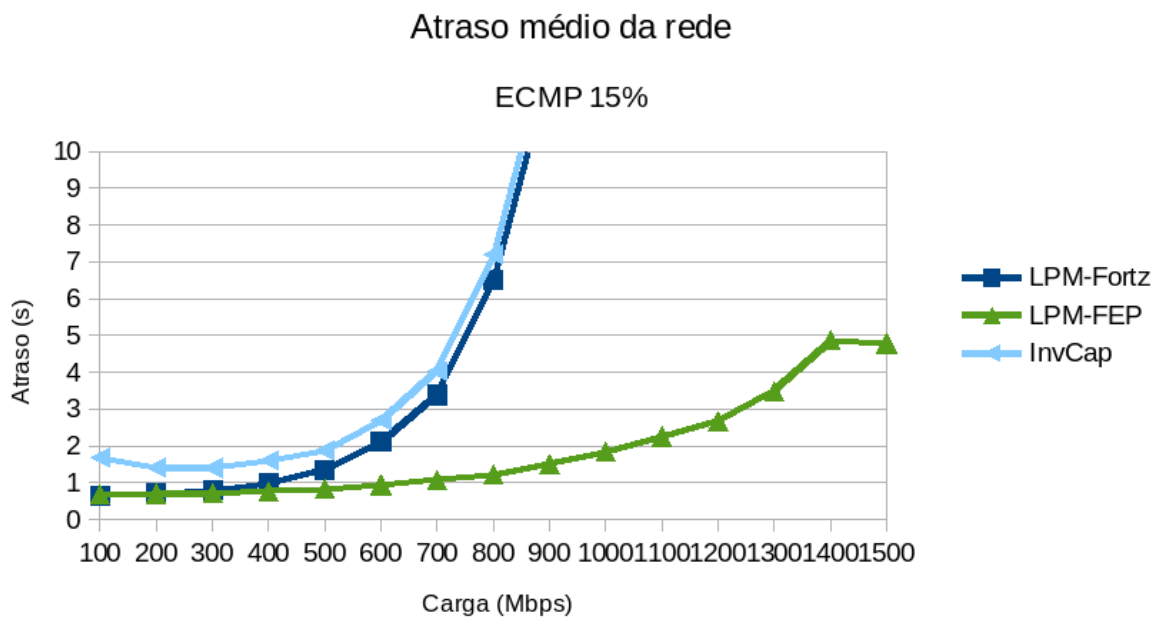


Figura 4.21: Atraso médio da rede com 15% de tráfego de fundo

Nas Figuras 4.23 e 4.24 é possível verificar que a partir de 400 Mbps o LPM-FEP apresentou atrasos quase constantes, mesmo com uma maior utilização da rede. Os valores do atraso são consideravelmente menores que nos outros dois modelos em que o InvCap apresentou os maiores atrasos. Há um pico de atraso em torno de 300 Mbps no LPM-FEP que é rapidamente normalizado. Esta variação é percebida no gráfico da utilização que apresenta uma alta utilização para baixas cargas e vai reduzindo o tráfego quando a carga aumenta. Porém percebe-se que neste cenário o atraso é muito alto, atingindo mais do que 10 segundos, o que pode gerar atrasos fim a fim extremamente altos.

4.4.3 Análise geral dos resultados

De acordo com os resultados da simulação dos cenários 1 e 2, pode-se observar que existe uma coerência nos resultados do atraso e carga média nos dois cenários. No caso do cenário 1, por se utilizar uma matriz de tráfego de fundo com 70% dos enlaces com autossimilaridade. Os valores dos atrasos nesta rede tendem a ser menores.

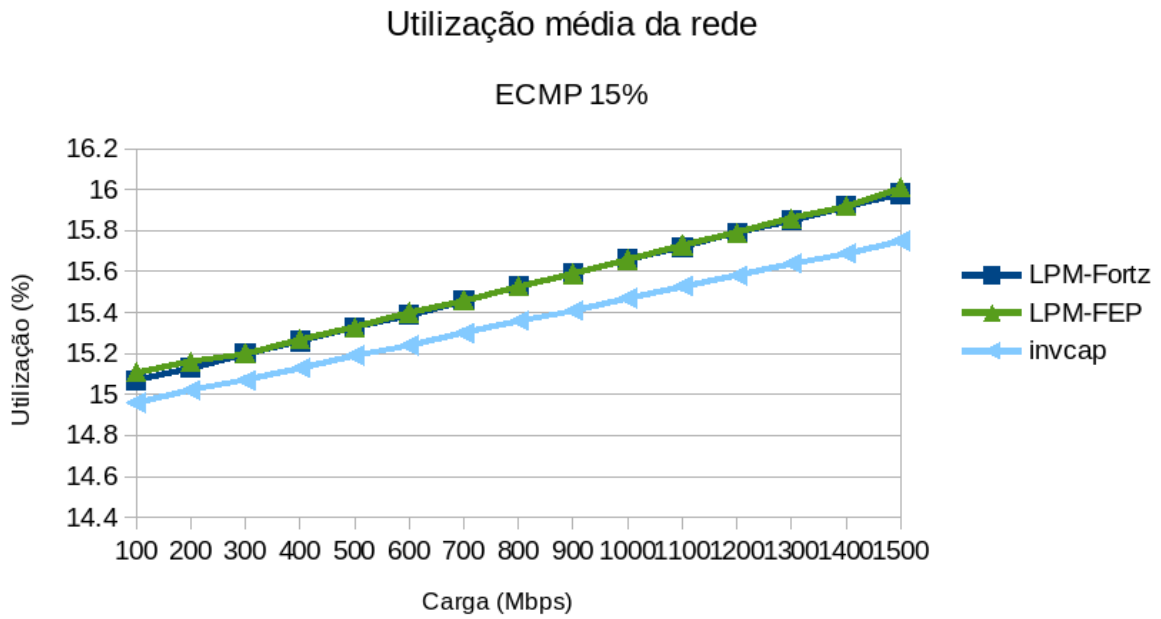


Figura 4.22: Utilização média da rede com 15% de tráfego de fundo

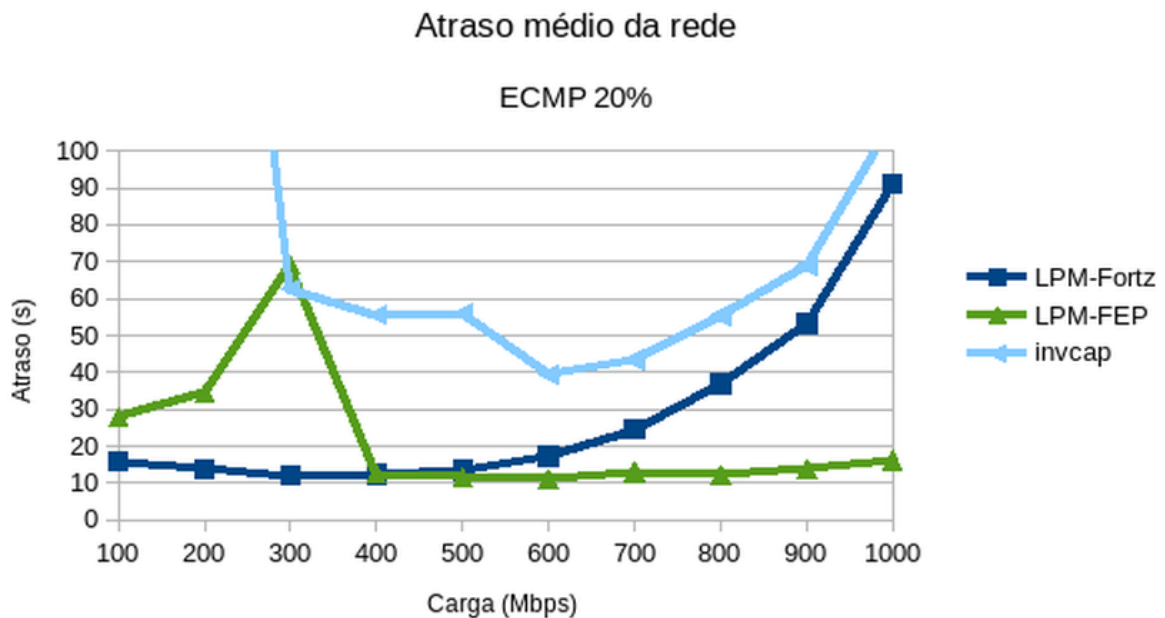


Figura 4.23: Atraso médio da rede com 20% de tráfego de fundo

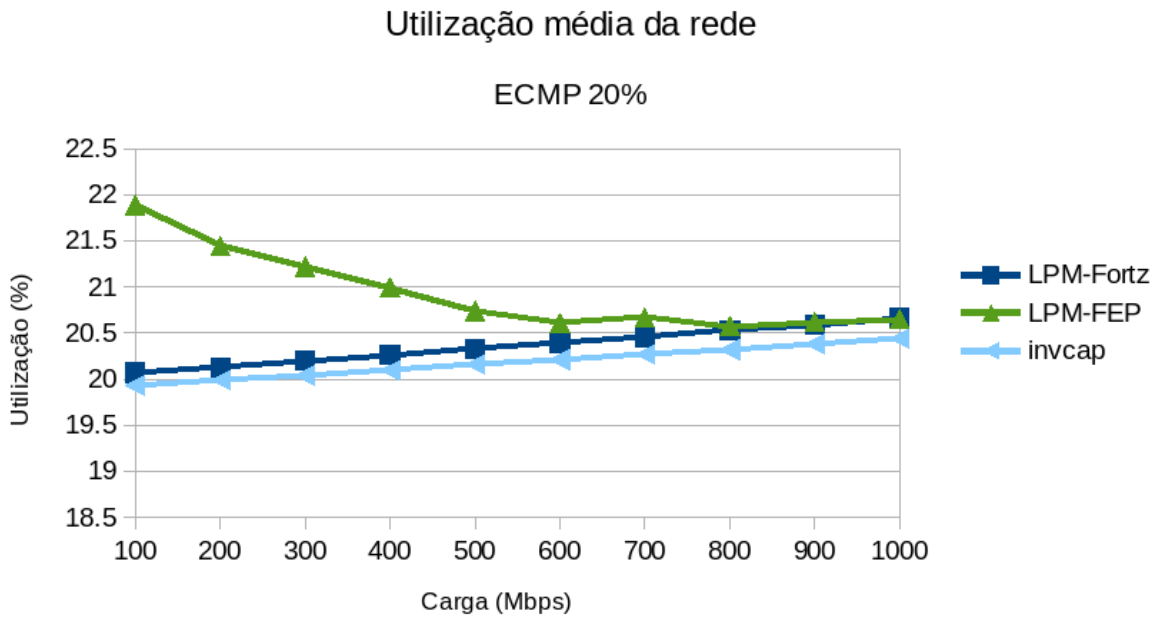


Figura 4.24: Utilização média da rede com 20% de tráfego de fundo

De acordo com [52], para aplicações de *stream* de vídeo que não é interativo, os roteadores na rede devem ser configurados com atrasos máximos igual a 5 segundos por nó, que representa o caso do FTV. Além disto, de acordo com Tanenbaum em [53], é bastante tolerável que aplicações de vídeo ao vivo tenham atrasos máximos de até 15 segundos.

Com os resultados das simulações apresentados nas seções anteriores, para manter o atraso em até 5 segundos por nó, é necessário manter a carga média da rede abaixo de 25%. Este resultado se baseia na consideração de que o tráfego de vídeo é autossimilar e que ainda será agregado a um tráfego de fundo que também é autossimilar.

O valor de 25% de carga média na rede parece ser muito baixo e é um resultado interessante. Hoje em dia, a utilização média de uma rede como a utilizada na simulação, formada principalmente por enlaces ópticos, oscila ao redor de 16,27%, de acordo com a página da RNP [46]. Este limite mostra como é forte a influência da autossimilaridade no congestionamento e na necessidade de otimizar o roteamento.

Entretanto, o FEP, utilizado na avaliação da proposta deste trabalho, estabelece valores máximos do atraso. Este resultado se avaliado em um ambiente real tende a apresentar valores inferiores do atraso.

Os resultados experimentais mostram que a metodologia de otimização de roteamento proposta neste trabalho poderia reduzir o atraso em relação aos modelos tradicionais de roteamento. Em todos os casos analisados, o atraso calculado com a metodologia proposta neste trabalho apresentou uma tendência de diminuição em relação aos outros

modelos, entretanto, na utilização média da rede, pode-se observar que existe uma certa equivalência entre os modelos.

Capítulo 5

Conclusões e Trabalhos Futuros

Este trabalho apresentou uma metodologia de engenharia de tráfego baseada na otimização do roteamento de protocolos tradicionais da pilha TCP/IP. Para a otimização foi utilizado o protocolo OSPF que é reconhecido como confiável, escalável e simples. O OSPF escolhe as melhores rotas baseado no menor custo dos enlaces no caminho até o destino e caso encontre vários menores caminhos, poderá dividir o tráfego utilizando um mecanismo chamado ECMP. Foram analisados também diversos trabalhos que utilizam a engenharia de tráfego em protocolos tradicionais de menor caminho afim de obter uma melhor distribuição do tráfego pela rede, porém foi constatado que nenhuma destas pesquisas considera a autossimilaridade como métrica na escolha das rotas.

Neste trabalho, com o objetivo de avaliar o desempenho da proposta, foi escolhida a aplicação FTV que se baseia na tecnologia de vídeo digital. Este vídeo é caracterizado como autossimilar, uma característica das séries temporais com invariância a escala e dependência de longa duração. Este tipo de tráfego foi extensamente estudado nas últimas décadas por não poder ser representado pelo popular processo de Poisson, uma vez que possui um comportamento pouco empírico e difícil de ser previsto. Assim, sua caracterização foi feita considerando um processo estocástico chamado Fractal Envelope Process, conforme diversos trabalhos relatados. Este processo permite encontrar um limite superior de atraso e utilização para uma porção significativa do trafego. A partir dessas características foi possível modelar o tráfego de vídeo FTV como autossimilar de forma que forneça os parâmetros necessários para a realização da tarefa de otimização.

Desta forma foi desenvolvido um modelo de engenharia de tráfego para otimização de roteamento que considera o comportamento autossimilar do tráfego agregado e a carga do enlace como métrica em protocolos de roteamento tradicionais de menor caminho. Este modelo permitiu estabelecer um limite superior para o atraso do tráfego de vídeo FTV o qual é de grande utilidade no processo de planejamento da rede.

Com este modelo foi possível demonstrar por simulação que o atraso de uma aplicação

autossimilar pode ser reduzido ao considerar a autossimilaridade na escolha das rotas. Foram avaliados três modelos para escolha de pesos dos enlaces: O LPM-FEP, LPM-Fortz e InvCap, este último é a escolha padrão dos roteadores Cisco.

Considerando a topologia da Rede Nacional de Pesquisa (RNP) para cada modelo, foram simuladas quatro redes transportando uma grande variedade de cargas que representavam o tráfego de vídeo. Para cada rede foi reservada uma fração da capacidade para o tráfego de fundo, também autossimilar. E para cada uma destas situações foram simulados dois cenários com distribuições de autossimilaridade diferentes. Para acrescentar aleatoriedade ao modelo foram gerados 10 modelos de tráfego de fundo com diferentes parâmetros de similaridade para cada cenário totalizando 240 iterações de simulação. Foram calculadas as médias de utilização de cada enlace da rede e agregados por tráfego de fundo e por cenário para calcular o atraso médio da rede sob estas condições.

Os resultados de ambos cenários foram bastante coerentes, porém o primeiro cenário apresentou o atraso um pouco menor por possuir apenas 70% do tráfego autossimilar. A escolha das rotas considerando a autossimilaridade foi decisiva para a descoberta de rotas com menor atraso para o tráfego de vídeo autossimilar. Em todos os cenários o modelo LPM-FEP apresentou atrasos até 80% menores que os outros modelos, mesmo apresentando a utilização média 10% superior em alguns casos.

Considerando que para transmissões de vídeo ao vivo pela Internet é aceitável obter atrasos máximos de 5 segundos por nó, chegou-se à conclusão que para a fila do roteador não ultrapassar este valor, o tráfego médio da rede não pode ultrapassar 25% da capacidade média da rede. Este valor é necessário para que a rede possa comportar os picos de longa duração, característicos do tráfego autossimilar, desde que o protocolo de roteamento utilizado seja um protocolo tradicional de roteamento por menor caminho, como é o caso do OSPF.

Para trabalhos futuros, considera-se aplicar a metodologia desenvolvida em um ambiente dinâmico utilizando uma ferramenta de simulação como NS-3 (*Network Simulator*) que possa utilizar de rastros de vídeos reais para simular o comportamento dinâmico da rede e propor a alteração dinâmica dos pesos em protocolos tradicionais de menor caminho.

Referências

- [1] P. Abry and D. Veitch. Wavelet analysis of long-range-dependent traffic. *Information Theory, IEEE Transactions on*, 44(1):2–15, January 1998. 38
- [2] Ayşegül Altın, Bernard Fortz, Mikkel Thorup, and Hakan Ümit. Intra-domain traffic engineering with shortest path routing protocols. *Annals of Operations Research*, 204(1):65–95, 2013. 2, 3, 6, 14, 17
- [3] F. Baker, J. Krawczyk, and A. Sastry. Rsvp management information base using smiv2. RFC 2206 (Proposed Standard), sep 1997. RFC 2206 (Proposed Standard). 25, 30
- [4] S. Balasubramaniam, J. Mineraud, P. McDonagh, P. Perry, L. Murphy, W. Donnelly, and D. Botvich. An evaluation of parameterized gradient based routing with qoe monitoring for multiple iptv providers. *IEEE Transactions on Broadcasting*, 57(2):183–194, June 2011. 10
- [5] Sasitharan Balasubramaniam, Dmitri Botvich, Julien Mineraud, and William Donnelly. Parameterised gradient based routing (pgbr) for future internet. In *Proceedings of the 2009 International Conference on Advanced Information Networking and Applications*, AINA '09, pages 58–65, Washington, DC, USA, 2009. IEEE Computer Society. 13
- [6] Simon Balon, Fabian Skivée, and Guy Leduc. How well do traffic engineering objective functions meet the requirements. 3976:75–86, 2006. 2, 3, 18
- [7] J. Beran, R. Sherman, M.S. Taqqu, and W. Willinger. Long-range dependence in variable-bit-rate video traffic. *Communications, IEEE Transactions on*, 43(2/3/4):1566–1579, Feb 1995. 36
- [8] Dimitri P. Bertsekas and Robert G. Gallager. *Data networks*. Prentice-Hall International, London, 1992. 1ère édition publiée en 1987. International student edition en 1992. 4, 16
- [9] Andreas Bley. Routing and capacity optimization for ip networks. In Jörg Kalcsics and Stefan Nickel, editors, *Operations Research Proceedings 2007*, volume 2007 of *Operations Research Proceedings*, pages 9–16. Springer Berlin Heidelberg, 2008. 17
- [10] Shigang Chen and K. Nahrsted. An overview of quality of service routing for next-generation high-speed networks: problems and solutions. *Network, IEEE*, 12(6):64–79, Nov 1998. 2

- [11] CollStreaming. About us, agosto 2014. <http://www.coolstreaming.us/page/company.html>. 25
- [12] T. H. et al. Cormen. *Introduction to Algorithms*. MCGraw-Hill, 2 ed. edition, 2001. 8
- [13] S. E. Deering and D. R. Cheriton. Host groups: A multicast extension to the internet protocol. RFC 966, Dec 1985. Obsoleted by RFC0988. Status:UNKNOWN. 25
- [14] A. Feldmann, A. C. Gilbert, and W. Willinger. Data networks as cascades: Investigating the multifractal nature of internet wan traffic. *SIGCOMM Comput. Commun. Rev.*, 28(4):42–55, oct 1998. 1, 36
- [15] W. Fenner. Internet group management protocol, version 2. RFC 2236, Nov 1997. Updated by RFC 3376, <http://www.ietf.org/rfc/rfc2236.txt>. 25
- [16] Nelson L. S. Fonseca, Gilberto S. Mayor, and Cesar A. V. Neto. On the equivalent bandwidth of self-similar sources. *ACM Trans. Model. Comput. Simul.*, 10(2):104–124, April 2000. 1, 41, 43, 55
- [17] Behrouz A. Forouzan and Firouz Mosharraf. *Redes de Computadores: Uma Abordagem Top-Dow*. AMGH Editora, Porto Alegre, RS, Brazil, 1st edition, 2013. 5, 7, 10, 22, 23, 25, 27, 29
- [18] B. Fortz, J. Rexford, and M. Thorup. Traffic engineering with traditional ip routing protocols. *Communications Magazine, IEEE*, 40(10):118–124, Oct 2002. xi, 2, 13, 15, 16, 17, 18, 48, 56, 59
- [19] Bernard Fortz and Hakan Ümit. Efficient techniques and tools for intra-domain traffic engineering. *International Transactions in Operational Research*, 18, 2010. 59
- [20] Mark W. Garret and Walter Willinger. Analysis, modeling and generation of self-similar vbr video traffic. In *SIGCOMM 94*, London England, Aug 1994. ACM, ACM. 1, 40
- [21] GLPK. GNU Linear Programming Kit. Disponível em: <http://www.gnu.org/software/glpk>. 56, 57
- [22] Guilherme Godinho. Lg cria parceria para difusão da tecnologia 3d tv. In Internet, Dez 2009. <http://www.tecnologia.com.pt/2009/12/lg-cria-parceria-para-difusao-da-tecnologia-3d-tv/>. xi, 33
- [23] Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. RFC 1889 (Proposed Standard), January 1996. Obsoleted by RFC 3550. 25
- [24] Jiayue He, Jennifer Rexford, and Mung Chiang. Design for optimizability: Traffic management of a future internet. In Graham Cormode and Marina Thottan, editors, *Algorithms for Next Generation Networks*, Computer Communications and Networks, pages 3–18. Springer London, 2010. 2, 13

- [25] C.L. Hedrick. Routing information protocol. RFC 1058 (Historic), jun 1988. Updated by RFCs 1388, 1723. 11
- [26] IEEE. IEEE standards for local and metropolitan area networks: Supplement to carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications media access control (MAC) parameters, physical layer, medium attachment units, and repeater for 100 Mb/s operation, type 100base-t (clauses 21-30). IEEE Std 802.3u-1995 (Supplement to ISO/IEC 8802-3: 1993; ANSI/IEEE Std 802.3, 1993 Edition), January 1995. <http://standards.ieee.org/findstds/standard/802.3u-1995.html>. 21
- [27] ITU. Rec. itu-r BT.601-7. Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios, March 2011. <http://www.itu.int/rec/R-REC-BT.601/en>. 21
- [28] ITU. High efficiency video coding. Recommendation ITU-T H.265, 04 2013. 23
- [29] D. Katz, K. Kompella, and D. Yeung. Traffic engineering (te) extensions to ospf version 2. RFC 3630 (Proposed Standard), sep 2003. Updated by RFCs 4203, 5786. 13
- [30] Houssain Kettani and John A. Gubner. A novel approach to the estimation of the long-range dependence parameter. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS - II: EXPRESS BRIEFS*, 53(6):463–467, Jun 2006. 38
- [31] James F. Kurose and Keith Ross. *Redes de Computadores e a Internet*. Pearson, Boston, MA, USA, 2nd edition, 2003. 8, 22, 27
- [32] G. Leduc, H. Abrahamsson, S. Balon, S. Bessler, M. D’Arienzo, O. Delcourt, J. Domingo-Pascual, S. Cerav-Erbas, I. Gojmerac, X. Masip, A. Pescapè, B. Quoitin, S. P. Romano, E. Salvadori, F. Skivée, H. T. Tran, S. Uhlig, and H. ímit. An open source traffic engineering toolbox. *Comput. Commun.*, 29(5):593–610, March 2006. Disponível em <http://totem.run.montefiore.ulg.ac.be/>. 2, 3
- [33] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Netw.*, 2(1):1–15, feb 1994. 36
- [34] Baochun Li, Zhi Wang, Jiangchuan Liu, and Wenwu Zhu. Two decades of internet video streaming: A retrospective view. *ACM Trans. Multimedia Comput. Commun. Appl.*, 9:33, October 2013. 24, 27
- [35] Ze-Nian Li, Mark S. Drew, and Jiangchuan Liu. *Fundamentals of Multimedia*. Springer Publishing Company, Incorporated, 2nd edition, 2014. xi, xiii, 20, 21, 22, 23, 27
- [36] P. Malindi and Mte Kahn. Characterisation and modelling of h.264 video traffic source. *Journal for New Generation Sciences: Volume 7 Number 2*, 7(2):17, 2009. 1, 40

- [37] J. Moy. Ospf specification. RFC 1131 (Proposed Standard), oct 1989. Obsoleted by RFC 1247. 11
- [38] J. Moy. Ospf version 2. RFC 2328 (INTERNET STANDARD), apr 1998. Updated by RFCs 5709, 6549, 6845, 6860. 12, 15, 17, 53
- [39] Ilkka Norros. A storage model with self-similar input. *Queueing Systems*, 16(3-4):387–396, 1994. 1, 39, 41
- [40] Micha Pióro and Deepankar Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Elsevier, 2004. 14, 15
- [41] F. et al Porikli. Multimedia quality assessment. *IEEE Signal Processing Magazine*, 2011:164–177, November 2011. 27
- [42] PPLive. About us. Internet, agosto 2014. <http://www.pptv.com/aboutus/>. 25
- [43] Razer. Gaming systems. In Internet, dez 2014. <http://www.razerzone.com/gaming-systems>. xi, 33
- [44] Irini Reljin, Andreja Samčović, and Branimir Reljin. H.264/avc video compressed traces: Multifractal and fractal analysis. *EURASIP J. Appl. Signal Process.*, 2006:123–123, jan 2006. xi, xiii, 36, 40, 41, 42
- [45] Jennifer Rexford and Constantine Dovrolis. Future internet architecture: Clean-slate versus evolutionary research. *Commun. ACM*, 53(9):36–40, September 2010. 2
- [46] RNP. Topologia. na Internet, 07 2015. Disponível em <http://www.rnp.br/servicos/conectividade/rede-ipe>. 53, 70
- [47] Z. Sahinoglu and S. Tekinay. On multimedia networks: self-similar traffic and network performance. *Communications Magazine, IEEE*, 37(1):48–52, Jan 1999. 1
- [48] Fórum SBTv. Informações técnicas. In Internet, August 2014. www.dtv.org.br. 20
- [49] T. Scandarolli, R.L. de Queiroz, and D.A. Florencio. Attention-weighted rate allocation in free-viewpoint television. *Signal Processing Letters, IEEE*, 20(4):359–362, April 2013. xii, 59
- [50] P. Seeling and M. Reisslein. Video transport evaluation with H.264 video traces. *IEEE Communications Surveys and Tutorials, in print*, 14(4):1142–1165, 2012. Traces available at trace.eas.asu.edu. 44
- [51] Priscila Solis Barreto. *Uma Metodologia de Engenharia de Tráfego Baseada na Abordagem Auto-Similar para a Caracterização de Parâmetros e a Otimização de Redes Multimídia*. Tese, Universidade de Brasília, 2007. 16, 37, 40
- [52] Tim Szigeti and Christina Hattingh. *End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs (Networking Technology)*. Cisco Press, 2004. 54, 70

- [53] Andrew S. Tanenbaum. *Redes de Computadores*. Campus, 4th edition, 2003. 4, 22, 28, 30, 70
- [54] A.S. Tanenbaum and D. Wetherall. *Computer Networks*. Pearson, 5th edition, 2011. xiii, 2, 5, 6, 7, 8, 11, 22, 24, 25, 28, 45, 54
- [55] M. Tanimoto, M.P. Tehrani, T. Fujii, and T. Yendo. Free-viewpoint TV - a review of the ultimate 3DTV and its related technologies. *Signal Processing Magazine, IEEE*, 28(1):67–76, January 2011. xi, 31, 32, 33
- [56] M.P. et al. Tehrani. Use cases and requirements on free-viewpoint television (ftv). In *MPEG 106 - Geneva*. ISO, MPEG, October 2013. 32, 33
- [57] Founding Universe. Realnetworks, inc. history. In Internet, August 2014. <http://www.fundinguniverse.com/company-histories/realnetworks-inc-history/>. 19
- [58] Visual Networking Index VNI. Forecast and methodology, 2013–2018. Technical report, Cisco, 2014. <http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html>. 1
- [59] Oculus VR. Next-gen virtual reality. In Internet, Nov 2014. <http://www.oculus.com/rift/>. xi, 33
- [60] Tomohiro Yendo. Cylindrical 3D display visible from any angle. Seelinder, Ago 2014. <http://www.yendo.org/seelinder/>. xi, 32, 33