

# Otimização de Roteamento com Protocolos Tradicionais para Aplicações Free-Viewpoint Television (FTV)

Henrique D. Garcia<sup>1</sup>, Priscila Solis<sup>1</sup>

<sup>1</sup>Departamento de Ciência da Computação – Instituto de Ciências Exatas  
Universidade de Brasília (UnB) – Campus Universitário Darcy Ribeiro  
Caixa postal 4466 – CEP 70.910-900 – Brasília – DF – Brazil

henriquedgarcia@gmail.com, pris@cic.unb.br

**Abstract.** *This paper proposes a traffic engineering methodology that uses the self-similarity parameter of video traffic as a restriction on an Integer Linear Programming problem (ILP) for the definition of link costs to be used by the OSPF (Open Shortest Path First). The purpose of the methodology is to set a maximum delay value for transmission between image capture and middleboxes offering the service FTV (Free Viewpoint Television). The experimental results show that the proposed method can significantly reduce the maximum delay in the network and allow identification of threshold values for load on the links to facilitate the definition of maximum delay boundaries in video transmission environments on the Internet.*

**Resumo.** *Este trabalho propõe uma metodologia de engenharia de tráfego que utiliza o parâmetro de autossimilaridade do tráfego de vídeo como restrição em um problema de programação linear inteira (PLI) para a definição dos custos dos enlaces a serem utilizados pelo OSPF (Open Shortest Path First). O objetivo da metodologia é definir um valor de atraso máximo para transmissão entre a captura de imagens e os middleboxes que oferecem o serviço de FTV (Free Viewpoint Television). Os resultados experimentais mostram que a metodologia proposta pode reduzir significativamente o atraso máximo na rede e permitir a identificação de valores de limiar para a carga nos enlaces que facilitem a definição de limites máximos de atraso em ambientes de transmissão de vídeo na Internet.*

## 1. Introdução

Com as demandas atuais de banda das diversas aplicações multimídia, estima-se que até 2018 mais de 80% do volume de tráfego na Internet será de vídeo [VNI 2014]. Considerando que existe um consenso sobre as características estatísticas do tráfego multimídia serem definidas como autossimilar ou *multifractal* [Solis Barreto 2007][Sahinoglu and Tekinay 1999] [Reljin et al. 2006], torna-se um grande desafio garantir métricas de desempenho para as aplicações em uma arquitetura baseada no melhor esforço, como é o caso do TCP/IP.

Além da função básicas de transmitir dados, as redes atuais executam diversas tarefas para melhorar segurança, desempenho ou fornecer novas funcionalidades para o tráfego. Para este fim são utilizados *middleboxes*, tais como tradutores de endereços, *firewalls*, processadores de vídeo, entre outros. Quando os *middleboxes* são virtualizados,

acabam por gerar uma Rede de virtualização de funções (NFV *Network Functions Virtualization*) podendo serem inicializados ou desativados conforme necessário, aumentando a eficiência e dinamismo da rede.

Assim, os *middleboxes* têm acentuado o interesse em algoritmos que permitam:

1. Otimizar a colocação dos *middleboxes* em um ambiente de servidores de rede;
2. Acompanhar o tráfego através de sequências de *middleboxes* com base em uma política de alto nível e
3. Otimizar o roteamento do tráfego entre servidores e *middleboxes* através do cálculo de caminhos eficientes para cumprir requerimentos necessários de desempenho [Charikar et al. 2015].

Considerando que estas aplicações são hoje executadas em ambientes que utilizam principalmente o IPv4 (*Internet Protocol Version 4*), é factível considerar a otimização do roteamento com protocolos tradicionais, tais como o OSPF, que é suportado por praticamente todos os roteadores da Internet, não exigindo modificações adicionais nos equipamentos, são estáveis e de fácil gerenciamento.

Diversos trabalhos já estudaram a otimização de roteamento com protocolos tradicionais [Papadimitriou et al. 2015] [Leduc et al. 2006] [Fortz et al. 2002] fazendo uso de métricas dinâmicas nos enlaces, porém nenhum destes trabalhos considera a autossimilaridade do tráfego como componente da métrica para definir o custo do enlace.

Altin em [Altin et al. 2013] apresenta um levantamento das pesquisas mais relevantes relacionadas com a otimização de protocolos de menor caminho. Neste escopo, Fortz em [Fortz et al. 2002] desenvolve um modelo de Programação Linear Inteira (PLI) utilizado como referência neste trabalho. Posteriormente, em [Leduc et al. 2006] implementa as heurísticas propostas por Fortz no simulador TOTEM (TOolbox for Traffic Engineering Methods) e Balon em [Balon S, Skivée F. e Leduc G., 2006] comparam de diversas funções objetivos para engenharia de tráfego, na qual o modelo de Fortz apresenta um dos melhores resultados. Por outro lado, alguns autores como em [Papadimitriou et al. 2015] buscam novas técnicas para resolver problemas de otimização relacionadas a descoberta dos custos ótimos, uma vez que a maioria destes problemas são do tipo NP-Completo.

O presente trabalho pretende abordar a otimização de roteamento do tráfego em uma rede IP tradicional com uma arquitetura baseada no uso de *middleboxes* para aplicações FTV (*Free-View-Point Television*), ainda em desenvolvimento e padronização pela MPEG (*Moving Picture Experts Group*) [Tanimoto et al. 2011] [Scandarolli et al. 2013]. A FTV é uma aplicação que exige muito poder de processamento e alta largura de banda por transmitir e processar centenas de fluxos de vídeo ao vivo em alta resolução.

O principal objetivo é avaliar a eficiência de uma metodologia de engenharia de tráfego que considere as características estatísticas de autossimilaridade do tráfego de vídeo agregado como restrição em um problema de programação Linear Inteira (PLI) para a descoberta dos custos dos enlaces utilizados por protocolos do menor caminho, afim de reduzir o atraso máximo dentro de sistema autônomo (AS - *Autonomous System*). Assim, é proposta a otimização do roteamento da rede afim de oferecer Qualidade de Serviço (QoS - *Quality of Service*) para o tráfego de vídeo entre a captura do vídeo e os *middleboxes* que posteriormente poderão ser disponibilizados para a comunidade.

Este artigo é organizado da seguinte forma: na seção 2 é apresentada a fundamentação teórica e revisão bibliográfica usada como base para o desenvolvimento da proposta. A seção 3 discute o modelo de otimização proposto e em seguida, a seção 4 apresenta os resultados experimentais. A seção 5 apresenta as conclusões e trabalhos futuros.

## 2. Fundamentação Teórica e Trabalhos Relacionados

### 2.1. As Aplicações de Vídeo Digital

As aplicações de vídeo consomem muito mais largura de banda do que aplicações de voz ou *web*, por exemplo. Um vídeo com resolução de 525 x 858 pixels, 30 quadros por segundo e 16 bits de cores pode exigir uma taxa de transmissão constante de aproximadamente 216 Mbps. Logo é necessário algum mecanismo de compressão para reduzir a taxa de transmissão, principalmente nos os novos formatos de Televisão de Ultra Alta Definição (UHDTV - *Ultra High Definition TV*) que pode atingir a resolução de 7680 x 4320 pixels.

Para transmitir vídeo pela Internet pode-se adotar uma arquitetura cliente-servidor, no qual o transmissor transmite o fluxo requisitado diretamente para o cliente ou uma arquitetura P2P(*Peer-to-Peer*), onde de forma distribuída, cada nó da rede é considerado um cliente e servidor ao mesmo tempo. Para garantir escalabilidade em redes cliente-servidor, geralmente são implementadas Redes de Distribuição de Conteúdo (CDN – *Content Delivery Network*), um sistema de servidores distribuídos pela Internet para garantir acesso à mídia de forma mais eficiente para usuários geograficamente próximos.

A *Free viewpoint TeleVision* (FTV) é uma aplicação de vídeo ao vivo que permite ao espectador total liberdade na escolha do ângulo de visão dentro da cena do vídeo um ou mais ângulos de visão de forma a trazer o ambiente para mais próximo da realidade [Scandarolli et al. 2013]

A arquitetura dessa aplicação é mostrada na Figura 1. Nesta aplicação um arranjo de câmeras captura a cena que inicialmente passa por correção de cores e ângulos das diversas câmeras. Em seguida é realizada a estimação de profundidade utilizando câmeras adjacentes para gerar um mapa de profundidade que será codificado junto com os fluxos de cada câmera antes de ser transmitido ou armazenado. No receptor, o vídeo é decodificado e os fluxos de câmeras adjacentes são interpolados junto com o mapa de profundidade, que criando um novo fluxo de vídeo em um ângulos de onde fisicamente não existem câmeras, de acordo com a solicitação do espectador. Este novo fluxo pode ser exibido por diversos tipos de dispositivos, seja 3D, estereoscópico ou 2D.

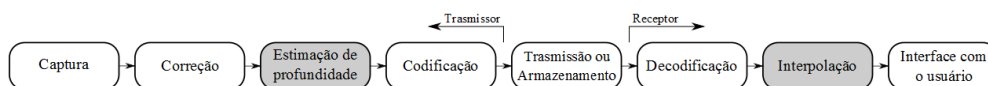


Figura 1. Arquitetura da aplicação FTV

### 2.2. Engenharia de Tráfego e Otimização de Roteamento

Um dos protocolos de roteamento mais antigos e sólidos na Internet é o protocolo Aberto de Primeiro Menor Caminho (OSPF - *Open Shortest Path First*) [Moy 1998], que usa como base o estado do enlace para o cálculo de rotas e funciona da seguinte maneira: após

Variável	Descrição
$i, j, s, t$	Nós em uma rede
$N$	Conjunto de todos os nós em uma rede
$(i, j)$	Um enlace que sai do nó $i$ em direção ao nó $j$
$A$	Conjunto de todos os enlaces da rede
$\phi_{(i,j)}$	Custo do enlace $(i,j)$
$(s, t)$	Par origem-destino
$\delta_{(s,t)}$	Tráfego de origem em $s$ e destino em $t$
$f_{(i,j)}^{(s,t)}$	Fluxo de origem $s$ e destino $t$ passando pelo enlace $(i,j)$
$l_{(i,j)}$	Carga total em um enlace
$c_{(i,j)}$	Capacidade do enlace $(i,j)$

**Tabela 1. Notação das variáveis utilizadas para o MCNFP.**

descobrir seus vizinhos, os roteadores inundam a rede com informações de seus bancos de dados de rotas e custos de enlace. Assim, cada roteador constrói a topologia completa da rede e executa o algoritmo de menor caminho de Dijkstra [Bertsekas and Gallager 1992] para determinação das melhores rotas para todos os possíveis caminhos.

Na prática a descoberta dos custos ótimos é um problema de Programação Linear Inteira (PLI) do tipo NP - Completo, sendo bastante difícil encontrar uma solução ótima rapidamente. Porém, pode-se encontrar uma solução sub-ótima em tempo polinomial através de alguma heurística como *Branch and Cut*, Busca Local ou Busca Tabu [Fortz and Papadimitriou 2014].

O problema geral de roteamento, também conhecido como Problema de Roteamento de Multifluxos de Rede (MCNFP – *Multi-Commodity Network Flow Problem*) [Bertsekas and Gallager 1992] é descrito pelo PLI nas Equações 1 a 5, em que as demandas de transmissão competem entre si por recursos na rede. A notação utilizada nas equações é descrita na Tabela 1. Apesar do roteamento dentro de uma rede ser mais complexo, o MCNFP é uma boa aproximação para se encontrar os pesos de forma rápida e eficiente [Bertsekas and Gallager 1992].

$$\min \quad \Phi = \sum_{(i,j) \in A} \phi_{(i,j)} \quad (1)$$

$$\sum_{j \in N} f_{(i,j)}^{(s,t)} - \sum_{j \in N} f_{(j,i)}^{(s,t)} = \begin{cases} \delta(s, t) & \text{se } i = s \\ -\delta(s, t) & \text{se } i = t \\ 0 & \text{caso contrario} \end{cases} \quad s, t \in N, (i, j) \in A \quad (2)$$

$$l_{(i,j)} = \sum_{(s,t) \in N} f_{(i,j)}^{(s,t)} \quad (3)$$

$$l_{(i,j)} \leq c_{(i,j)} \quad (i, j) \in A \quad (4)$$

$$\phi \geq \alpha_z l_{(i,j)} - \beta_z c_{(i,j)} \quad (i, j) \in A, z \in Z \quad (5)$$

A Equação 1 apresenta a função objetivo que procura minimizar os custos dos enlaces da rede em função do custo  $\phi$ . A Equação 2 é a restrição de continuidade que

garante que cada entidade que produz e consome tráfego  $(s, t)$  esteja descrita em uma matriz de tráfego  $\delta$ . A Equação 3 apresenta a definição de carga como sendo a soma de todos os fluxos em um enlace  $(i, j)$ , a Equação 4 garante que a carga em um enlace nunca ultrapasse sua capacidade, a Equação 5 é a função de custo que representa o aumento ou diminuição do custo dos enlaces formada por um conjunto de inequações lineares. Os pesos  $\alpha$  e  $\beta$  estão associados a velocidade de crescimento do custo. No modelo proposto por este trabalho estes coeficientes são calculados realizando a regressão linear nos intervalos de interesse do modelo de atraso utilizado.

### 2.3. Caracterização e Modelagem do Tráfego de Vídeo em Aplicações FTV

Até meados da década de 90 o comportamento do tráfego era modelado mediante o processo de Poisson e o modelo M/M/1. Diversas pesquisas confirmaram que este modelo não era adequado para o tráfego LAN, WAN ou multimídia pois se tratavam de distribuições de cauda longa e não exponencial [Leland et al. 1994] [Feldmann et al. 1998] [Beran et al. 1995] e sua modelagem deve ser feita através de um processo que considere o índice de autossimilaridade e a sua variabilidade. Este índice de autossimilaridade também é chamado de parâmetro de Hurst ( $H$ ), onde  $0,5 < H < 1,0$  para séries autossimilares.

Norros em [Norros 1994], descreve um processo estocástico autossimilar chamado Processo Envelope Fractal (FEP - *Fractal Envelope Process*) caracterizado pelo valor de  $H$ , pelo desvio padrão e pela utilização da rede, ao considerar encapsular a fração  $1 - \epsilon$  do tráfego gerado pelo processo do Movimento Browniano Fracionário (fBm), em que  $\epsilon$  é a perda de pacotes por estouro de *buffer* dos roteadores e equivale a probabilidade do fBm ultrapassar o FEP, que será usado como o limite superior da fila. Desta forma, o número máximo de elementos em uma fila do FEP é dada pela Equação 6, em que  $k = -2 \ln \epsilon$  e  $q_{max}$  representa o tamanho máximo da fila.

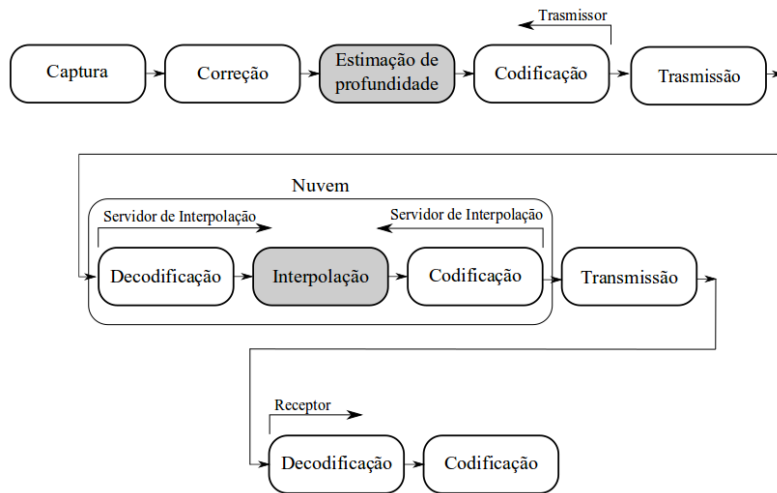
$$q_{max} = (c - \bar{a})^{\frac{H}{H-1}} (\kappa\sigma)^{\frac{1}{1-H}} (H)^{\frac{H}{1-H}} (1 - H) \quad (6)$$

Com base nessas ideias é possível estimar a largura de banda necessária para a transmissão do tráfego autossimilar e o atraso máximo da fila pelas Equações 7 e 8 respectivamente, conforme proposto em [Fonseca et al. 2000].

$$\hat{c} = \bar{a} + K^{\frac{H-1}{H}} (\kappa\sigma)^{\frac{1}{H}} H (1 - H)^{\frac{1-H}{H}} \quad (7)$$

$$t_{max} = \left( \frac{\kappa\sigma H}{c - \bar{a}} \right)^{\frac{1}{1-H}} \quad (8)$$

Quando o tráfego é agregado com vários fluxos autossimilares em um enlace, é necessário calcular o novo valor do  $H$  e da variância chamado de parâmetro  $H$  agregado ou  $H_g$  e de variância agregada  $\sigma^2$  [Solis Barreto 2007], conforme mostrado nas Equações 9 e 10, respectivamente.



**Figura 2. Arquitetura FTV utilizada neste trabalho proposta em [Scandarolli et al. 2013]**

$$H_g = \frac{\sum_s H_s \sigma_s^2}{\sum_s \sigma_s^2} \quad (9)$$

$$\sigma_g^2 = \sum_s \sigma_s^2 \quad (10)$$

O tráfego de vídeo FTV consiste na transmissão de centenas de fluxos de vídeo junto com seus respectivos mapas de profundidade, tudo simultaneamente para o cliente. Como atualmente não existe um padrão de codificação para o FTV, neste trabalho foi considerado que a aplicação codifica os fluxos individualmente usando o H.265.

Para a modelagem do tráfego de vídeo utilizou-se os traços do vídeo *Tears of Steel* em resolução 4K (4096 x 1744) codificado com o padrão H.265 [Seeling and Reisslein 2014]. A taxa média de transmissão do vídeo é de 5,67 Mbps, o parâmetro H calculado é de 0,87, com um coeficiente de variação de Pearson igual 1,5. Nesta caracterização foi considerado um fluxo agregado de 10 fluxos de vídeo, totalizando uma taxa média de transferência igual a 56,7 Mbps. Foi considerado também o  $\epsilon = 0,001$  e o parâmetro H do tráfego de fundo variado de acordo com o cenário.

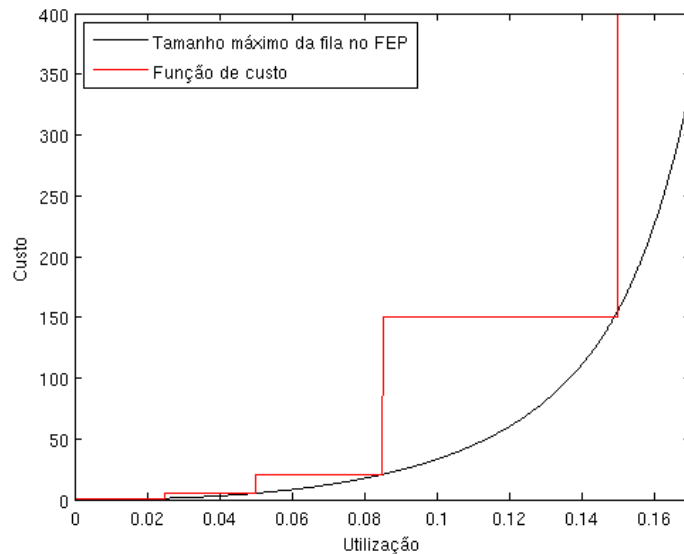
A arquitetura padrão do FTV não é totalmente adequada para transmissão em *broadcast* e algumas propostas tem sido feitas nesse sentido [Scandarolli et al. 2013]. A Figura 2 mostra a arquitetura proposta, onde o transmissor de vídeo poderá enviar todos os fluxos para um *middlebox* responsável pela interpolação e fisicamente próximo do cliente final através de uma arquitetura cliente-servidor, como se fosse uma Rede de Distribuição de Conteúdo (CDN - *Content Distribution Network*). Este servidor por sua vez, se comunicará com os clientes finais que solicitarão um ângulo de visão específico que será interpolado, codificado e entregue por algum sistema distribuição de fluxo de vídeo como P2P ou DASH, por exemplo. Essa arquitetura é utilizada neste trabalho para avaliar a otimização de roteamento entre a captura das imagens e o *middlebox*.

### 3. O modelo de otimização dos custos no OSPF

O protocolo OSPF realiza a descoberta de rotas considerando unicamente o custo estático do enlace. Considerando o crescimento da fila como um processo de Poisson, Fortz em [Fortz et al. 2002] desenvolveram um modelo linear de custos que acompanha o crescimento da fila, mas que não considera a autossimilaridade e nem a variância do tráfego.

$$\varphi' = \begin{cases} 1 & 0 \leq l/c < 2,5/100 \\ 5 & 2,5/100 \leq l/c < 5/100 \\ 20 & 5/100 \leq l/c < 8,5/100 \\ 150 & 8,5/100 \leq l/c < 15/100 \\ 1300 & 15/100 \leq l/c < 20/100 \\ 17000 & 20/100 \leq l/c < \infty \end{cases} \quad (11)$$

Com base no conceito desse modelo, neste trabalho foi desenvolvida uma função de custo descrita pela Equação 11 em que o crescimento do custo acompanha o crescimento de uma fila FEP, conforme mostrado na Figura 3. Assim, propõe-se um problema de programação linear, conforme mostrado nas Equações 12 a 22. Este modelo é chamado de LPM-FEP (*Linear Program Model - Fractal Envelope Process*)



**Figura 3. Crescimento do custos do enlace em função da utilização para o FEP**

Para a resolução do modelo linear foi considerado como demanda a estimativa de largura de banda do FEP. Como o objetivo é atribuir custos menores aos enlaces com menor autossimilaridade agregada, e consequentemente menores atrasos máximos, o parâmetro de Hurst agregado de todos os fluxos de vídeo aparece como multiplicador do coeficiente angular, regulando o crescimento da função de custo, gerando custos maiores a enlaces com alto  $H_g$  e custos menores a enlaces com baixo  $H_g$ . O PLI foi elaborado em Uma Linguagem de Programação Linear (AMPL - *A Mathematical Programming Language*) e resolvido com o Kit de Programação Linear da GNU (GLPK - *GNU Linear Programming Kit*)

$$\text{Minimizar } \Phi = \sum_{(i,j) \in A} \phi_{(i,j)} \quad (12)$$

$$\text{Sujeito a :} \quad (13)$$

$$\sum_{j \in N} f_{(i,j)}^{(s,t)} - \sum_{j \in N} f_{(j,i)}^{(s,t)} = \begin{cases} \delta(s,t) & \text{se } i = s \\ -\delta(s,t) & \text{se } i = t \\ 0 & \text{caso contrario} \end{cases} \quad s, t \in N, (i,j) \in A \quad (14)$$

$$l_{(i,j)} = \sum_{(s,t) \in N} f_{(i,j)}^{(s,t)} \quad (15)$$

$$l_{(i,j)} \leq c_{(i,j)} \quad (i,j) \in A \quad (16)$$

$$\phi_{(i,j)} \geq l_{(i,j)} \quad (17)$$

$$\phi_{(i,j)} \geq 5l_{(i,j)}H_{(i,j)} - \frac{1}{10}c_{(i,j)} \quad (18)$$

$$\phi_{(i,j)} \geq 20l_{(i,j)}H_{(i,j)} - \frac{17}{20}c_{(i,j)} \quad (19)$$

$$\phi_{(i,j)} \geq 150l_{(i,j)}H_{(i,j)} - \frac{119}{10}c_{(i,j)} \quad (20)$$

$$\phi_{(i,j)} \geq 1300(i,j)H_{(i,j)} - \frac{922}{5}c_{(i,j)} \quad (21)$$

$$\phi_{(i,j)} \geq 17000l_{(i,j)}H_{(i,j)} - 3325c_{(i,j)} \quad (22)$$

A Figura 4 compara o crescimento dos custos em função da utilização entre o LPM-FEP para diversos  $H_g$  e o LPM-Fortz em escala logarítmica. Pode-se observar o crescimento do custo conforme o parâmetro de autossimilaridade aumenta. Para realizar a otimização geral da rede, baseado nos modelos propostos anteriormente, foi aplicado o *framework* mostrado na Figura 5, descrito a seguir:

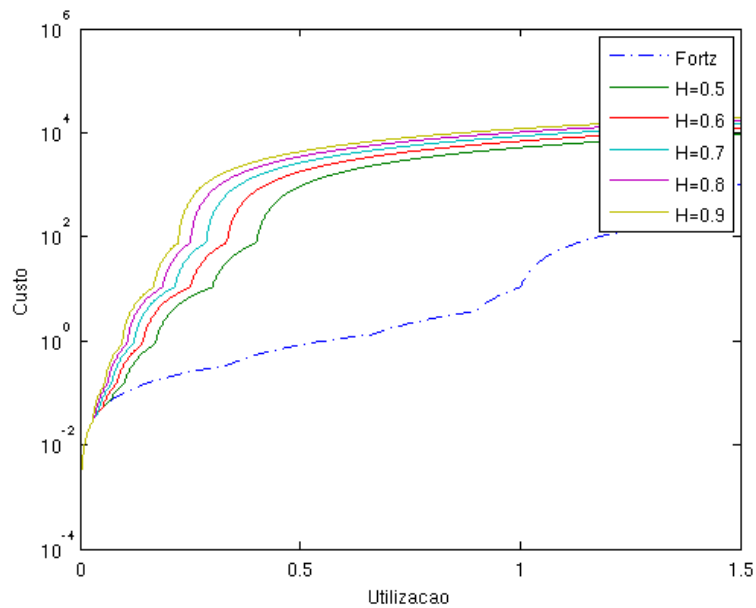
- Passo 1: Caracterizar a fila dos roteadores com base o processo autossimilar da Equação 6;
- Passo 2: Obter informações da topologia e do tráfego de fundo;
- Passo 3: Calcular  $H_g$ ,  $\sigma_g^2$  e a banda efetiva para todos os enlaces da rede conforme Equações 9, 10 e 7 respectivamente;
- Passo 4: Gerar a matriz de tráfego com os dados gerados no passo 3;
- Passo 5: Resolver o modelo de PLI e extrair o valor dos custos dos enlaces;
- Passo 6: Inserir os novos custos na rede, obtidos no passo 5;
- Passo 7: Monitorar as métricas

Para validar a proposta, serão feitas comparações com o modelo descrito em [Fortz et al. 2002], aqui chamado de LPM-Fortz e o modelo de custo padrão da Cisco que usa o inverso da capacidade do enlace [Cisco 2015] de forma estática, ou seja  $\phi_{(i,j)} = 1/c_{i,j}$ , aqui chamado de Invcap.

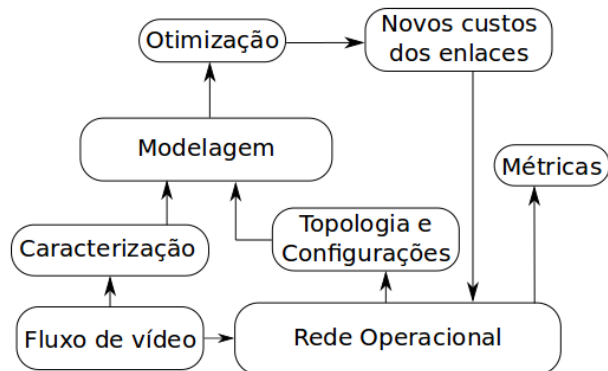
#### 4. Resultados Experimentais

Para simular o roteamento do tráfego foi utilizado o simulador TOTEM (*TOolbox for Traffic Engineering Methods*), que implementa diversos algoritmos para engenharia de tráfego com protocolos de menor caminho [Leduc et al. 2006].





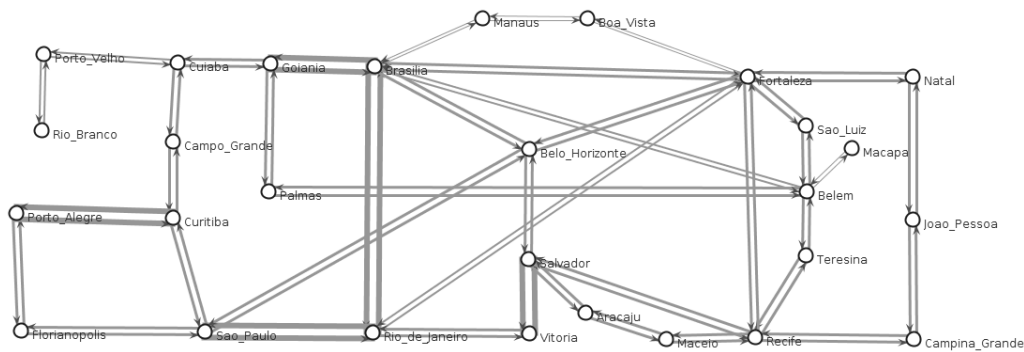
**Figura 4. Comparação do crescimento do custo do enlace para o LPM-FEP com vários valores de  $H$  e o LPM-Fortz (azul pontilhado).**



**Figura 5. Fluxograma utilizado para a simulação**

No TOTEM foi implementada a topologia da Rede Nacional de Pesquisa (RNP), disponível em [RNP 2015], conforme a Figura 6, porém com alteração na capacidade de alguns enlaces afim de reduzir a homogeneidade da rede. A rede conecta 27 cidades com enlaces pelo Brasil com capacidades de 100 Mbps a 20 Gbps. Foi considerado que a rede utiliza o protocolo OSPFv2 conforme definido no RFC 2328 [Moy 1998] sob IPv4, ou seja o protocolo utiliza o algoritmo de Dijkstra e distribui o tráfego igualmente entre todos os enlaces que possuem o mesmo custo, técnica conhecida como Multicaminho de Mesmo Custo (ECMP - Equal-Cost Multi-Path).

Na simulação foi considerado como métrica a Média do atraso máximo da rede, que é a razão da soma dos atrasos máximos de todos os enlaces pelo número de enlaces. Esta métrica é relevante neste trabalho pois reflete o limite superior que um pacote espera no *buffer* do roteador antes de ser encaminhado para o próximo salto ou descartado até chegar ao *middlebox* que fará a distribuição aos usuários finais.

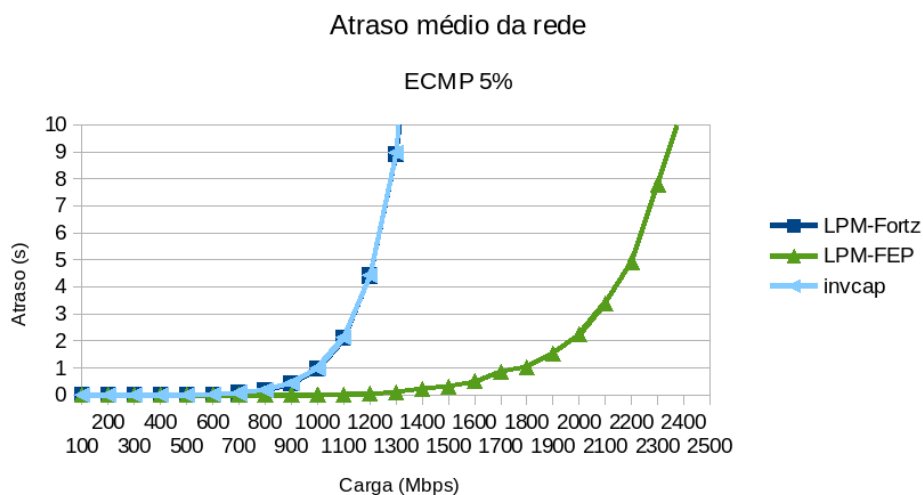


**Figura 6. Topologia da Rede Ipê utilizada para simulação**

De acordo com Szigeti em [Szigeti and Hattingh 2004] o atraso médio em cada nó não deve ultrapassar 5 segundos para transmissão de vídeo e em [Tanenbaum and Wetherall 2011] é especificado que até 15 segundos de atraso médio para vídeo ao vivo é aceitável. Logo, a análise dos resultados foi feita considerando esse atraso.

Para a simulação do tráfego de vídeo principal foi simulada uma demanda de tráfego autossimilar do nó Porto Alegre ao nó Belém variando de 100 Mbps a 2500 Mbps, de forma que o tráfego total sobre o enlace não ultrapasse 3000 Mbps, o coeficiente de variação igual a 1,5 e parâmetro H igual a 0,87. Todos os enlaces apresentam tráfego de fundo que varia de 5% até 20% de sua capacidade e com H aleatório entre 0,5 e 0,85 distribuído uniformemente. Foram realizadas 10 simulações para cada cenário e computados os valores médios.

A Figura 7 apresentam o atraso médio 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 LPM-Fortz mesmo quando aplicada mais carga.



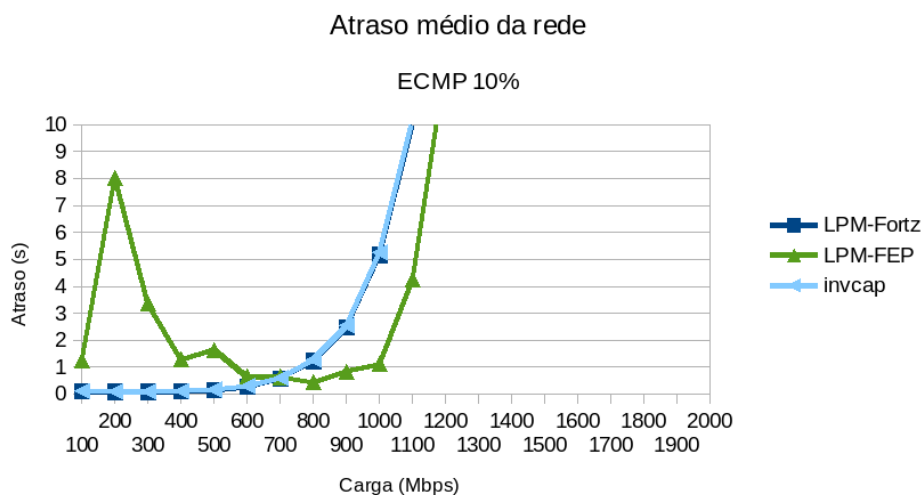
**Figura 7. Atraso médio da rede com 5% de tráfego de fundo**

Foram alcançados 5 segundos com 2300 Mbps de carga média na rede, 75% a

mais de carga do que LPM-Fortz e o Invcap atingiram esse mesmo atraso. Juntamente com o tráfego de fundo, esta carga corresponde a 28% da capacidade média da rede.

Quando o tráfego de fundo aumenta para 10% há um pico de atraso máximo do LPM-FEP a 200 Mbps de carga, como pode-se verificar no gráfico da Figura 8, porém logo o atraso se estabiliza e o atraso se mantém inferior aos outros modelos. Esta flutuações pode ocorrer devido ao espaçamento das utilizações na função de custo e podem ser resolvidas em um modelo de custo aperfeiçoado.

Para O LPM-FEP os valores de atraso de 5 segundos ocorrem entre 1100 Mbps e 1200 Mbps de carga, que junto como tráfego de fundo é equivalente a 22% da capacidade média da rede, enquanto o LPM-Fortz e Invcap apresentam um atraso de 5 segundos entre 900 Mbps e 1000 Mbps de carga.

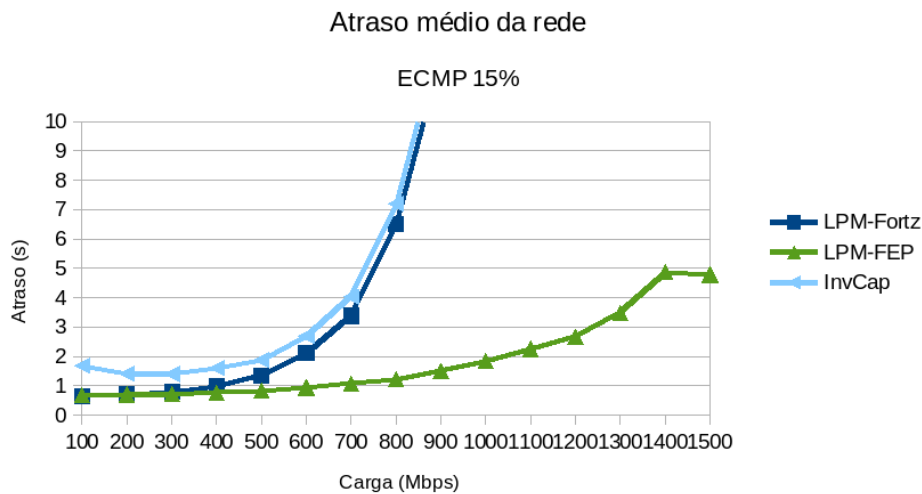


**Figura 8. Atraso médio da rede com 10% de tráfego de fundo**

Com 15% de tráfego de fundo, como mostra a Figura 9, o LPM-FEP apresenta uma distribuição mais uniforme do tráfego pela rede, onde pode-se observar que 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. Neste cenário o modelo Invcap apresenta os maiores atrasos e atinge atrasos de 5 segundos com a carga de 700 Mbps, 53% menos do que o LPM-FEP e praticamente o mesmo que o modelo LPM-Fortz.

A Figura 10 mostra os resultados para o tráfego de fundo igual a 20% a capacidade dos enlaces. O atraso médio utilizando o LPM-FEP foi superior aos LPM-Fortz até a carga de 400 Mbps e a partir deste ponto os atrasos foram inferiores a todos os outros modelos. Porém, pode-se observar que neste cenário o atraso é muito alto, atingindo mais de 10 segundos em toda a simulação, o que pode gerar atrasos fim a fim extremamente altos.

Assim, baseado no comportamento do tráfego gerado pelas simulações conclui-se que para manter o atraso médio de 99,9% do tráfego abaixo de 5 segundos por enlace, é necessário manter a carga média da rede abaixo de 25%. Este resultado baseia-se na consideração de que o tráfego de vídeo é autossimilar e que foi agregado a um tráfego de fundo que também é autossimilar.



**Figura 9. Atraso médio da rede com 15% de tráfego de fundo**

O valor de 25% de carga média na rede pode parecer muito baixo, o que é um resultado interessante, pois essa baixa utilização é necessária para proteger a rede das rajadas de longa duração gerada pelo tráfego autossimilar. Entretanto, o Processo Envelope Fractal, estabelece valores estatísticos máximos do atraso que o tráfego alcançar. Na prática, se este modelo for avaliado em um ambiente real o atraso pode apresentar valores menores. Até a data da publicação deste artigo a utilização média da rede IPê da RNP, oscila ao redor de 16,27% [RNP 2015].

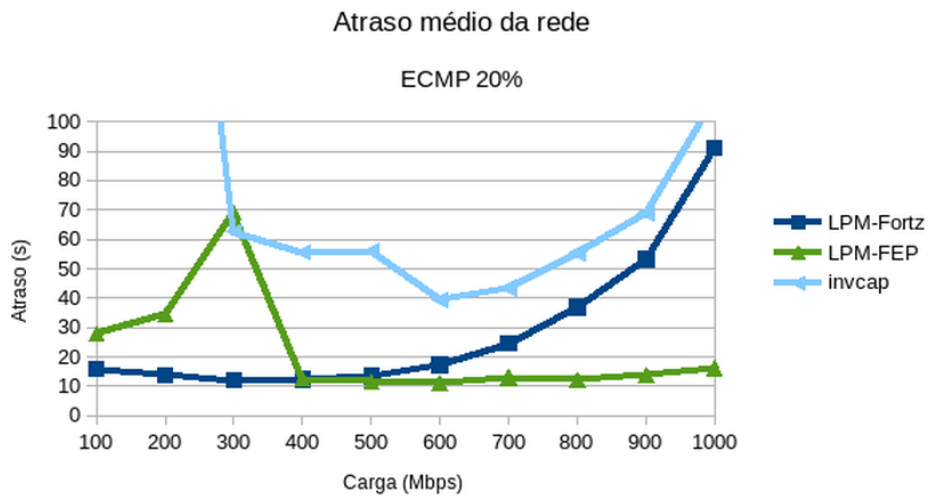
Os resultados experimentais mostram que a metodologia de otimização de roteamento proposta neste trabalho, com base em uma métrica resultante da caracterização do tráfego de vídeo autossimilar, tem o potencial de reduzir o atraso em relação ao uso de métricas tradicionais de roteamento, desde que a utilização da rede seja baixa o suficiente. Em todos os casos analisados, o atraso obtido com a proposta de otimização de roteamento deste trabalho, apresentou uma forte tendência de ser inferior do que aos outros modelos existentes.

## 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. Foi analisado e caracterizado o tráfego de vídeo digital e foi avaliado em uma arquitetura adaptada de FTV dentro de um sistema autônomo com o uso de *middleboxes*.

Com base nos parâmetros de autossimilaridade do tráfego de vídeo, foi desenvolvido um modelo de programação linear inteira para otimizar a escolha dos custos dos enlaces considerando o tráfego agregado. Foi demonstrado também que a escolha das rotas com base nesse modelo consegue reduzir o atraso de enfileiramento e manter a utilização da rede dentro de valores coerentes com outros modelos de otimização de roteamento.

Com os resultados, conclui-se de que a utilização média da rede não deve ultrapassar 25% para cumprir os requerimentos das transmissões de vídeo digital ao vivo na Internet. Este valor é um resultado interessante que demonstra a forte influência dos pi-



**Figura 10. Atraso médio da rede com 20% de tráfego de fundo**

cos de atividade de longa duração do tráfego característicos da autossimilaridade e as limitações existentes para garantir métricas de QoS para este tipo de aplicação.

Os trabalhos futuros desta pesquisa se concentrarão em aprimorar o modelo e estende-lo para outro tipo de aplicações e arquiteturas de serviço na Internet. Pretende-se aplica-lo a outros cenários e ampliar o estudos de algoritmos para resolução deste problemas de otimização.

## Referências

- Altın, A., Fortz, B., Thorup, M., and Ümit, H. (2013). Intra-domain traffic engineering with shortest path routing protocols. *Annals of Operations Research*, 204(1):65–95.
- Beran, J., Sherman, R., Taquu, M., and Willinger, W. (1995). Long-range dependence in variable-bit-rate video traffic. *Communications, IEEE Transactions on*, 43(2/3/4):1566–1579.
- Bertsekas, D. P. and Gallager, R. G. (1992). *Data networks*. Prentice-Hall International, London. 1ère édition publiée en 1987. International student edition en 1992.
- Charikar, M., Naamad, Y., Rexford, J., et al. (2015). Multi-commodity flow with in-network processing. Technical Report TR-995-15, Department of Computer Science, Princeton University, <ftp://ftp.cs.princeton.edu/techreports/2015/995.pdf>.
- Cisco (2015). Ios quality of service solutions configuration guide. Technical report. Release 12.2, [http://www.cisco.com/c/en/us/td/docs/ios/12\\_2/qos/configuration/guide/fqos\\_c.html](http://www.cisco.com/c/en/us/td/docs/ios/12_2/qos/configuration/guide/fqos_c.html), acessado em Set, 2015.
- Feldmann, A., Gilbert, A. C., and Willinger, W. (1998). Data networks as cascades: Investigating the multifractal nature of internet wan traffic. *SIGCOMM Comput. Commun. Rev.*, 28(4):42–55.
- Fonseca, N. L. S., Mayor, G. S., and Neto, C. A. V. (2000). On the equivalent bandwidth of self-similar sources. *ACM Trans. Model. Comput. Simul.*, 10(2):104–124.

- Fortz, B. and Papadimitriou, D. (2014). Branch-and-cut strategies for a multi-period network design and routing problem. In *Control, Decision and Information Technologies (CoDIT), 2014 International Conference on*, pages 128–133.
- Fortz, B., Rexford, J., and Thorup, M. (2002). Traffic engineering with traditional ip routing protocols. *Communications Magazine, IEEE*, 40(10):118–124.
- Leduc, G., Abrahamsson, H., Balon, S., et al. (2006). An open source traffic engineering toolbox. *Comput. Commun.*, 29(5):593–610. Disponível em <http://totem.run.montefiore.ulg.ac.be/>.
- Leland, W. E., Taqqu, M. S., Willinger, W., and Wilson, D. V. (1994). On the self-similar nature of ethernet traffic (extended version). *IEEE/ACM Trans. Netw.*, 2(1):1–15.
- Moy, J. (1998). Ospf version 2. RFC 2328 (INTERNET STANDARD). Updated by RFCs 5709, 6549, 6845, 6860.
- Norros, I. (1994). A storage model with self-similar input. *Queueing Systems*, 16(3-4):387–396.
- Papadimitriou, D., Fortz, B., and Gorgone, E. (2015). Lagrangian relaxation for the time-dependent combined network design and routing problem. In *Communications (ICC), 2015 IEEE International Conference on*, pages 6030–6036.
- Reljin, I., Samčović, A., and Reljin, B. (2006). H.264/avc video compressed traces: Multifractal and fractal analysis. *EURASIP J. Appl. Signal Process.*, 2006:123–123.
- RNP (2015). Topologia. na Internet. Disponível em <http://www.rnp.br/servicos/conectividade/rede-ipe>.
- Sahinoglu, Z. and Tekinay, S. (1999). On multimedia networks: self-similar traffic and network performance. *Communications Magazine, IEEE*, 37(1):48–52.
- Scandarolli, T., de Queiroz, R., and Florencio, D. (2013). Attention-weighted rate allocation in free-viewpoint television. *Signal Processing Letters, IEEE*, 20(4):359–362.
- Seeling, P. and Reisslein, M. (2014). Video traffic characteristics of modern encoding standards: H.264/avc with svc and mvc extensions and h.265/hevc. *The Scientific World Journal*, 2014:16.
- Solis Barreto, P. (2007). *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.
- Szigeti, T. and Hattingh, C. (2004). *End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs (Networking Technology)*. Cisco Press.
- Tanenbaum, A. and Wetherall, D. (2011). *Computer Networks*. Pearson, 5th edition.
- Tanimoto, M., Tehrani, M., Fujii, T., and Yendo, T. (2011). Free-viewpoint TV - a review of the ultimate 3DTV and its related technologies. *Signal Processing Magazine, IEEE*, 28(1):67–76.
- VNI (2014). Visual networking index - forecast and methodology, 2013–2018. Technical report, Cisco. <http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html>.